Tree-based Models for Poverty Estimation

A thesis presented in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

in

Statistics

at

Manawatu

Penelope A. Bilton

07/11/2016

Abstract

The World Food Programme utilises the technique of poverty mapping for efficient allocation of aid resources, with the objective of achieving the first two United Nations Sustainable Development Goals, elimination of poverty and hunger. A statistical model is used to estimate levels of deprivation across small geographical domains, which are then displayed on a poverty map. Current methodology employs linear mixed modelling of household income, the predictions from which are then converted to various area-level measures of well-being. An alternative technique using tree-based methods is developed in this study. Since poverty mapping is a small area estimation technique, the proposed methodology needs to include auxiliary information to improve estimate precision at low levels, and to take account of complex survey design of the data. Classification and regression tree models have, to date, mostly been applied to data assumed to be collected through simple random sampling, with a focus on providing predictions, rather than estimating uncertainty. The standard type of prediction obtained from tree-based models, a "hard" tree estimate, is the class of interest for classification models, or the average response for regression models. A "soft" estimate equates to the posterior probability of being poor in a classification tree model, and in the regression tree model it is represented by the expectation of a function related to the poverty measure of interest. Poverty mapping requires standard errors of prediction as well as point estimates of poverty, but the complex structure of survey data means that estimation of variability must be carried out by resampling. Inherent instability in tree-based models proved a challenge to developing a suitable variance estimation technique, but bootstrap resampling in conjunction with soft tree estimation proved a viable methodology. Simulations showed that the bootstrap based soft tree technique was a valid method for data with simple random sampling structure. This was also the case for clustered data, where the method was extended to utilise the cluster bootstrap and to incorporate cluster effects into predictions. The methodology was further adapted to account for stratification in the data, and applied to generate predictions for a district in Nepal. Tree-based estimates of standard error of prediction for the small areas investigated were compared with published results using the current methodology for poverty estimation. The technique of bootstrap sampling with soft tree estimation has application beyond poverty mapping, and for other types of complex survey data.

Acknowledgements

To Geoff Jones and Siva Ganesh I wish to express my gratitude for their guidance, advice and encouragement along this PhD journey, with splashes of humour to ease the task. I would also like to thank Stephen Haslett for his contribution to the thesis.

My thanks also to Massey University, including the Institute of Fundamental Sciences, for the provision of scholarships to fund the research.

I would like to extend my appreciation to Timothy Bilton, Jonathan Godfrey and Hannes Calitz for their support with software issues. To Kathryn Stowell, my deepest thanks for your moral and practical support in my time of crisis.

To my Creator, God and Father of my Lord Jesus Christ, without His grace, strength and wisdom, this work would not have been completed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

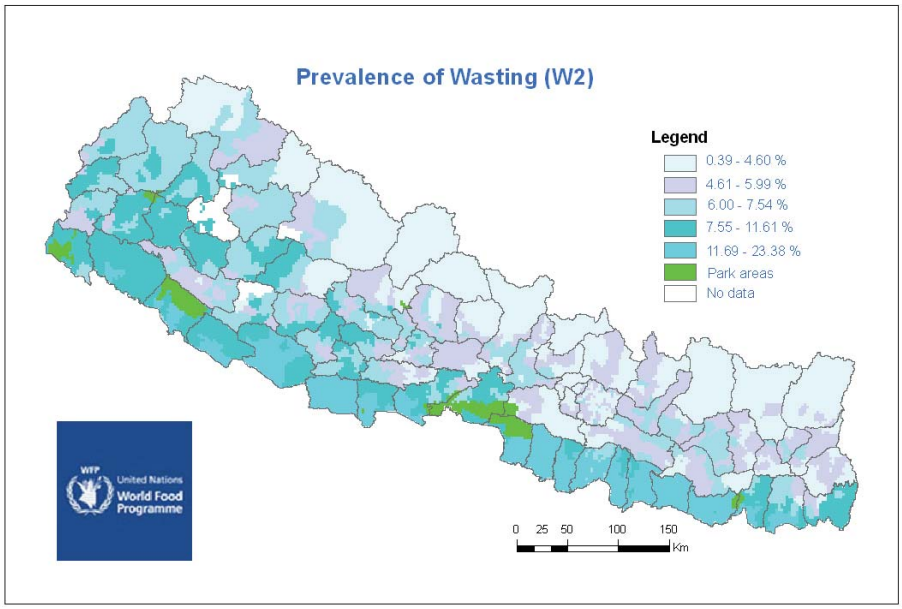
1.1 Poverty mapping

Elimination of poverty and hunger is the first two United Nations Sustainable Development Goals (United Nations 2016). In 2014, over US$5.5 billion in food assistance was provided to around eighty million people in seventy five third world countries (WFP 2015). The UN World Food Programme, WFP, utilises the methodology of poverty mapping to estimate levels of poverty to determine the allocation of food aid in a particular country. Since reducing poverty is a significant issue, involving substantial financial resources, accurate poverty maps are vitally important.

Poverty mapping comprises the measurement and analysis of wellbeing indicators, and the spatial representation of these indicators (Elbers et al. 2007). Statistical techniques are used by the WFP to generate estimates of deprivation measures in a country, which can then be combined with Geographic Information System (GIS) data to produce poverty maps (World Bank 2015). A poverty map can display disaggregated measures of poverty and other deprivation information at low geographical levels and so provide essential information for the distribution of aid. Figure 1.1, an example of a poverty map for Nepal, displays estimates of wasting, a weight-for-height measure which can reflect acute malnutrition in children under six.

In order to implement a programme of poverty reduction, the degree of deprivation at fine geographical levels needs to be ascertained. Estimating poverty over small areas rather than large regions in a country enhances the allocation and distribution of hundreds of millions of dollars of essential aid. However, national sample surveys of household income and expenditure or child nutrition conducted on a small scale have inadequate sample sizes and, consequently, inaccurate poverty estimates. Applying statistical modelling to produce estimates of deprivation and associated poverty maps addresses this core problem. Economists, including those at the World Bank, have used linear mixed models, a type of linear regression modelling, to estimate poverty levels (Minot & Baulch 2005, Quintano et al. 2007).

Figure 1.1: Poverty map of wasting in children under 6 in Nepal (Haslett & Jones 2006)



The current methodology employed by the World Bank to estimate indicators of well-being (Haslett & Jones 2010) is the method proposed by Elbers et al. (2003), and subsequently referred to as the ELL method, which utilises a linear mixed model to provide an estimate of poverty for individual households in a country, and then aggregates results to predict poverty at small area level. The determinants of poverty operate at different spatial scales (eg region vs household), so multiple sources of variation must be incorporated into the model to determine the accuracy of the estimates. Traditional small area estimation techniques directly model the population characteristic being estimated. In contrast, ELL models an underlying continuous variable which is then translated into indicators of well-being using non-linear functions, as discussed in the next section. Analysis of indigence in Nepal, applying the ELL methodology, has examined different categories of deprivation for each of three well-being indicators: poverty, undernourishment and child malnutrition (Haslett & Jones 2006). Poverty measures can be developed from underlying continuous variables representing household income and kilocalorie consumption using a mathematical framework. Anthropometric measures can be developed from underlying variables of height, weight and age.

The ELL methodology for analysing determinants of poverty has been used or tested in many countries, including Thailand (Healy et al. 2003), Cambodia (Fujii 2008), Bhutan (Haslett & Jones 2008*a*), Timor-Leste (Haslett & Jones 2008*b*), Vietnam (Lanjouw et al. 2013), as well as throughout South America (Hentschel et al. 2000, Ebers et al. 2008). Ferré et al. (2012) studied poverty estimates generated using the ELL method to investigate the relationship between poverty and city size for eight developing countries.

But these methods are not perfect and not the only analytic option. The basic statistical problem is one of classifying households as poor or non-poor, or classification

of children under five years of age as being stunted, underweight, or wasted, since this is the basis for aggregation of the area-level estimates of poverty. Alternative statistical methods exist for this purpose but have not, to date, been used for poverty estimation. Exploration of the use of tree based methods for poverty estimation is one of the core components of this research. The ultimate goal is to develop a better technique to predict poverty, that enhances the allocation of aid resources and reduce costs. Small gains in methodology producing improved estimation could be of considerable importance.

1.2 Development of poverty measures

The definition and measurement of poverty is an issue of much debate (Hagenaars & De Vos 1988, Sen 1985). A definition provided by the World Bank asserts that "poverty is pronounced deprivation in wellbeing" (World Bank 2000). The approach to the measurement of poverty recommended by the World Bank, and adopted to monitor the Sustainable Development Goals, is based upon the per capita income of a household. If this quantity is below a specified threshold, z, designated the poverty line, then every member of the household is considered to be in poverty. The population parameters representing poverty measures are estimated using non-linear functions of the underlying variable, per capita household expenditure. Since this variable has a highly right-skewed distribution, the natural log transformation is applied to achieve a symmetric target variable. If $\mathcal{E}_n$ denotes the per capita household expenditure for the n^{th} individual in the population then the value of the target variable of the ELL model for the n^{th} individual is $Y_n = \log(\mathcal{E}_n)$. The poverty measures can be represented mathematically using the Foster, Greer and Thorbecke (FGT) identity (1984):

$$P_a = \frac{1}{N} \sum_{n=1}^{N} \left(\frac{\mathrm{z} - \mathcal{E}_n}{\mathrm{z}} \right)^a \cdot I\left(\mathcal{E}_n < \mathrm{z}\right) \ , \tag{1.1}$$

where N is the total number of individuals in the area of interest. I is an indicator function taking the value 1 for individuals in households with per capita expenditure below the poverty line and 0 otherwise. P_a symbolises the various poverty measures;

1. poverty **incidence** or **prevalence** when $a = 0$; the percentage of the population below the poverty line
2. poverty **gap** when $a = 1$; the average distance below the poverty line
3. poverty **severity** when $a = 2$; the average squared distance below the poverty line, which gives emphasis to those in greatest need

The modelling provides an estimate of poverty, Y_n, for the n^{th} individual in the population, which is then backtransformed to an estimate of per capita expenditure using the relationship $\mathcal{E}_n = \mathrm{e}^{Y_n}$. These values of $\mathcal{E}_n$ are then aggregated across the area of interest.

Poverty is determined at household level, but enumerated at individual level. Varying values for the parameter a in the FGT identity (Equation (1.1)) provide a family of poverty measures which can allow for different levels of sensitivity towards income distribution among the poorest members of society (Baker & Grosh 1994).

The most common measurement of poverty is related to the cost-of-basic-needs approach (Haughton & Khandker 2009). An in-country poverty line is constructed to represent the per capita expenditure required to meet the basic needs of households across the whole country, including food and non-food items (Ravallion & Bidani 1994). Alternatively, a poverty line can be calculated separately for different regions across the country, which is useful when considerable price variation exists in different geographical locations. A third approach is to adjust household per capita expenditure using regional price indices, providing a "real" per capita expenditure and, again, a single poverty line can be implemented across the whole country (Ravallion 1998). The analysis based on tree models in this thesis employs the single poverty line used by Haslett & Jones (2006) of 7695.744 rupees, representing per capita expenditure per year in average 2003 Nepalese rupees. Other measures of deprivation have been developed, but are not considered in the thesis.

1.3 Other measures of deprivation

Poverty is the main indicator of deprivation in a country but other measures have also been used. Caloric intake is an underlying continuous variable which is used to develop indicators of under-nourishment (Wodon 1997). The average adult equivalent caloric intake for a household is measured and compared against a threshold. A per adult equivalent consumption measure can portray differences in need between ages groups (Haughton & Khandker 2009). In a similar process to that used to develop poverty measures, the FGT equations described in Section 1.2 are employed to provide measures of prevalence, gap and severity of caloric intake below the threshold.

Indicators of child malnutrition are obtained from the underlying variables of height, weight and age for children under five years of age (Fujii 2010). From these child anthropometric measurements, indicators of deprivation are constructed at individual child level, rather than at household level. Variables representing height-for-age, weight-for-age and weight-for-height are compared with equivalent international reference standards. The child malnutrition measure of *stunting* describes children with height-for-age value less than two standard deviations below the median in the reference population. Similarly, *underweight* specifies low weight for age as compared with the international reference. Low values of standardised weight-for-height indicates *wasting*, evidence of acute malnutrition.

1.4 Advantages of poverty mapping

Small area estimation using the ELL methodology produces statistically reliable estimates at fine spatial levels. The distributions of these estimates can be projected onto geographical maps using GIS techniques. Estimates of welfare are merged with geographic coordinates at district or sub-district level to produce poverty maps which provide detailed and accurate descriptions of the spatial distribution of poverty at low levels of aggregation (Hentschel et al. 2000).

Elbers et al. (2007) assessed whether poverty could be reduced by targeting of resources based upon poverty maps at fine spatial levels. They found that targeting at low geographical levels resulted in a significant increase in financial efficiency. Across three countries with different stages of welfare distributions and development, the gains were considerable and of similar extent. The effectiveness of poverty mapping is dependent upon careful differentiation of localities with respect to poverty levels, achieved through increased precision in the poverty estimates, largely influenced by the high explanatory power of the derived model. Homogeneity of poverty levels within a locality is also an important factor in the efficacious use of poverty maps. The benefit of poverty maps can also be extended by overlaying with spatially referenced maps relating to public services, infrastructure, etc (World Bank 2015). The objective is a better understanding of factors affecting the distribution of resources.

1.5 Implementation of Poverty Mapping in Nepal

Poverty mapping in Nepal, one of the poorest countries in Asia, is the context of this thesis. The aim of the research is to develop an alternative technique to the linear mixed model devised by Elbers et al. (2003), ELL, which is the standard statistical methodology used to estimate poverty incidence, gap and severity. An accurate assessment of the degree of poverty at fine spatial levels across a country is essential to efficiently target resources to the most deprived and vulnerable. In Nepal, a variety of topography from lowland hills to the world's highest mountains results in diversity of weather and climate. These factors, as well as regional and ethical differences, contribute to the disparate levels of well being or poverty in this nation.

In order to ascertain the spatial patterns of deprivation across Nepal, several poverty measures need to be evaluated over sub-populations including ecological zones, development regions, districts and ilakas. The various poverty measures evaluated are not estimated directly but as non-linear functions of an underlying target variable, Y. Direct measurements of the target variable Y were taken on the sampled units, households (or individual children), within each small area to provide explicit estimates of Y for each sub-population. Data for the estimation process was sourced from national surveys (Central Bureau of Statistics, Nepal 2004a,b, Nepal Demographic and Health Survey 2001) and a census (Central Bureau of Statistics, Nepal 2002), usually undertaken by the Nepal Central

Bureau of Statistics. The surveys, which comprised a two-stage stratified cluster design, had very high response rates (Haslett & Jones 2006). However, sample sizes in the subpopulations tend to be small, resulting in large standard errors and consequently unreliable estimates. This problem is addressed in small area estimation methodology by employing auxiliary information to improve the estimates and increase precision of standard errors. Utilising poverty mapping can provide better estimation of the deprivation at small area level within a country, and facilitate more efficient allocation of aid.

1.6 Scope of the thesis

This thesis focuses on investigating the feasibility of adapting tree based models for small area estimation of poverty. Development of a better model for estimation of poverty measures, with reliable predictions and sensible standard errors of prediction, is a major emphasis of the research. Standard errors are an important feature of the estimation process since they illustrate the practicality of the estimates produced. The amount of variability in estimates determines the level of aggregation required for reasonable accuracy.

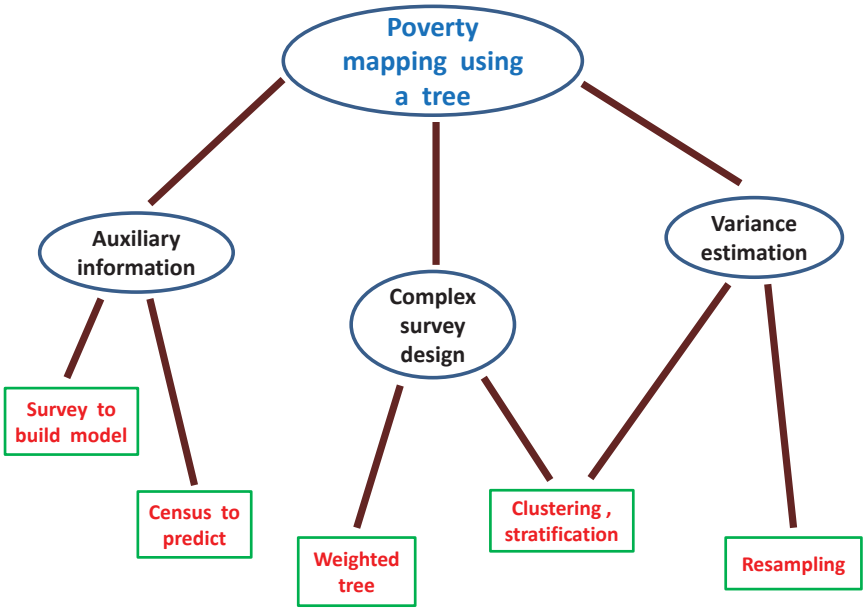
Poverty mapping, as a methodology, comes into the category of small area estimation. A "small area" is any geographical domain which is too small for direct estimates of the quantity of interest to be obtained with sufficient precision (Rao & Molina 2015). The problem of imprecise estimates at low geographic levels needs to be addressed by utilising auxiliary information. When a sample survey has a complex structure, the survey design elements used to collect the data must be accounted for in the modelling process. In addition, a mathematically tractable formula for variability is often not feasible with a complex survey sampling design, so some type of resampling is required for variance estimation. These three aspects of modelling for small area estimation with complex surveys are summarised in Figure 1.2.

ELL, the current standard application of small area estimation for predicting indicators of well being (Haslett & Jones 2010), was specifically designed for small area estimates of poverty. It comprises a weighted linear mixed model with a random error component for cluster effects. Auxiliary information required for precise estimates is provided by using survey data to build the model and census information to predict over the small area. A set of variables common to both survey and census supply the link between modelling and prediction.

The regression type structure of the ELL modelling readily incorporates the survey design elements of stratification, clustering and weighting. However, the dataset of auxiliary information comprises more than one hundred variables, and the task of selecting suitable predictors from individual variables and interactions can be complex. In addition, poverty incidence is estimated as a non-linear function of the underlying variable being modelled. Classification tree modelling (Breiman et al. 1984) provides a simple and direct method of estimating poverty incidence. The technique is independent of parametric assumptions and incorporates with ease non-linear predictors as well as interactions of

explanatory variables.

Figure 1.2: Structure of tree-based modelling to generate poverty estimates



Statistical classification and data mining techniques have not, to date, been used for poverty estimation because these techniques currently assume that the data are independently and identically distributed. However, the sample survey data that must be used in the statistical modelling for poverty mapping contains complex survey design aspects such as stratification, clustering and weighting. Theoretical extensions of classification and regression trees to allow them to be properly applied to complex survey data, and to incorporate multiple sources of variation, forms a key part of the thesis. Current classification techniques tend to focus on classifying individual records, so development of aggregation methods (such as are needed for the small area estimates in poverty mapping) involving tree models are also part of the research. Because sample surveys are used in investigations across a very wide range of science and social science disciplines, such new statistical techniques will have potential application well beyond poverty studies.

The scope of the thesis includes utilising tree based models as an alternative to the linear regression modelling of ELL. The task of the research is to find ways to incorporate small area estimation techniques into the tree model, so as to provide valid predictions of poverty, as well as taking account of the complex survey design. In Chapter 2, the literature review outlines the features of small area estimation , complex survey design and resampling techniques for variance estimation, and describes the ELL method, the current standard application of small area estimation for predicting indicators of well being. The methodology underlying tree based models is also discussed. Chapter 3 introduces classification tree models for poverty incidence built using data from Nepal. Incorporation of survey weights and auxiliary information into the model is discussed. Hard and soft classification tree estimates of poverty incidence are defined. Estimates produced by

classification tree models are compared with those obtained by using the ELL technique.

The task of finding a suitable resampling method for generating standard errors proved problematic due to the instability of tree models. Chapter 4 examines the cause of instability of the classification tree estimates, an inherent feature of tree based models, in the context of the Nepal data. A rigorous method for estimation of standard errors of prediction for poverty incidence when the data has been generated by simple random sampling is presented in Chapter 5. This method is extended to data with a complex survey structure in Chapter 6. The methodology developed for small area estimation of poverty incidence using a classification tree is adapted in Chapter 7 to generate estimates of all three measures of poverty, incidence, gap and severity, utilising regression tree models. Mathematical derivations of soft regression tree estimates for poverty gap and poverty severity are presented and found to produce reliable predictions. Conclusions of the research and further avenues of study are discussed in Chapter 8.

The most important feature of the thesis is the adaptation of tree models to produce reasonable standard errors for aggregates of individual predictions, when using complex survey data. There are three sources of variability when modelling poverty measures in the Nepal context; model uncertainty, variability at cluster level and variability at household level. The clusters and households constitute random effects in the modelling process. Due to the complex survey design in the data structure, a mathematically tractable form of the variance is not feasible. Thus, some type of resampling is required to generate standard errors of prediction. The thesis examines variance estimation using two types of resampling; jackknife and bootstrap methods.

Prediction of poverty in Nepal using tree based models melds the features of complex survey design, incorporation of auxiliary information, and variance estimation, with the theory of classification and regression tree models. The literature relating to these areas of statistics is examined in the next chapter.

Chapter 2

Literature Review

2.1 Introduction

An important focus of the thesis is the development of tree based models for complex survey data, in the context of poverty mapping in Nepal, as outlined in Section 1.6. Since estimation of poverty measures is usually based upon information collected through complex sample surveys, techniques to incorporate elements of complex survey design into the estimation process are needed. In addition, a key component of the research is development of a methodology for generating valid standard errors of prediction. Poverty estimates are generally required for small domains of a country, but small sample sizes lead to unstable results. Small area estimation methods were developed to resolve this problem. The ELL methodology (Elbers et al. 2003) extended the technique of small-area estimation to measures of poverty derived from per capita expenditure at household level (Haslett & Jones 2006).

Poverty mapping is a small area estimation technique. To undertake poverty mapping in Nepal using tree based models, the intertwining of features of small area estimation and complex survey design with the methodologies of classification and regression trees is required. Statistical theory and methods underpinning, and leading up to, the research are discussed in this chapter. Firstly, we describe the key modelling components, including small area estimation techniques, which need to be considered when modelling poverty. An overview of the ELL methodology, the current approach to poverty mapping, is presented. Included in the discussion is the implementation of the three core components of poverty mapping in ELL. Also outlined are the principles undergirding tree based models, the proposed alternative to ELL for poverty estimation.

2.2 Components of poverty mapping

There are three important modelling features which need to be incorporated into any method for small area estimation of poverty: inclusion of auxiliary information, complex survey design and variance estimation. In this section, different methods of providing

auxiliary information to improve estimate precision are presented, probability sampling and complex survey design are outlined, and various techniques of variance estimation are reviewed.

2.2.1 Incorporating auxiliary information

A poverty map displays estimates of deprivation measures across small domains. It is generally the case with sample surveys that information obtained at low geographical level is derived from sample sizes which are too small for good precision of estimates. Extra information is required to provide reliable estimates. The process of incorporating this extra information is referred to as "borrowing strength".

2.2.1.1 Borrowing strength

The term "small area" is applied to a subpopulation for which direct estimates cannot be provided with sufficient precision (Jiang & Lahiri 2006). An alternative is to find indirect estimators which increase the sample size and so decrease the variability (Rao & Choudhry 1995). This approach applies the technique of "borrowing strength" by incorporating auxiliary information available at the small area level (Ghosh & Rao 1994). Large sample data, e.g. census data, is suitable for aggregation at low levels but income and consumption are either not evaluated at all in a census (Alderman et al. 2002) or are measured poorly. One approach is to augment the information contained in a survey from one area with data from the survey from another area. This is especially useful when there is a degree of correlation between the two areas. This additional information can also be collected within a different time frame. The small area estimation methodology, utilising auxiliary information to provide indirect estimates for small domains, is generally classified into two broad groupings, traditional indirect methods based on implicit models, and model-based techniques (Coondoo et al. 2011).

2.2.1.2 Direct estimators

Unless the sample size at small area level is reasonably large, direct estimators, based solely on the sample data from the small area, are not usually utilised, since a small sample size produces imprecise estimates. There are two commonly used direct estimators. The *simple expansion estimator* applies to the sample values weights derived from the population and total sample size, while the *poststratified estimator* weights the observations with respect to the small domain only. The poststratified estimator is more efficient than the simple expansion estimator but its variance is still likely to be too large (Rao & Choudhry 1995).

Direct estimates tend to be unbiased but unstable, large variability being the consequence of small sample size. An alternative to using direct estimates is indirect estimation, which uses supplementary information to "borrow strength" from related small areas or larger areas (Ghosh & Rao 1994). Indirect estimation includes traditional indirect methods and model-based techniques.

2.2.1.3 Traditional indirect estimators

Indirect estimation derives estimates at small area level by utilising extra information. The scarcity of data at small area level provided by a survey can be augmented with additional information from census and other sources. Reliable estimates of the variable of interest may then be obtained using this supplementary data with no increase in survey costs or non-sampling error (Coondoo et al. 2011). Two commonly used traditional indirect estimators are the synthetic estimator and the composite estimator.

2.2.1.4 Synthetic estimator

An estimator for a small domain is described as a synthetic indirect estimator if it is not directly obtained from the survey results but is derived from a reliable direct estimator covering a larger area which includes the particular small domain of interest (Gonzalez 1973). The small area is assumed to contain the same characteristics as the larger area from which the direct estimate was taken (Villa Juan-Albacea 2009). Thus the estimate is taken at a higher level of aggregation and scaled down in proportion to the small area level.

2.2.1.5 Composite estimator

Utilising a synthetic estimator may result in bias in the estimate, since the assumption of similar characteristics between the small domain and larger region may not hold (Ghosh & Rao 1994). A composite estimator balances the instability of a direct estimator (large variance due to a small sample size) with the potential bias of a synthetic estimator by taking a weighted average of the two estimators.

2.2.1.6 Model based estimators

Another approach to indirect estimation at small area level is to use model-based estimators. These estimators can be categorised as using frequentist or Bayesian methodologies. The model-based small area estimation methods can be further classified into two rough groupings. Models for which specific covariates are available at area level only are known as *area level random effects models*. A *nested error unit level regression model* incorporates auxiliary information at unit level (Coondoo et al. 2011). Frequentist model-based techniques include the *Regression-synthetic* estimator, the *Best Linear Unbiased Prediction* or BLUP estimator and the *Empirical Best Linear Unbiased Prediction* or EBLUP estimator. Bayesian techniques for small area estimation discussed are the *Empirical Bayes Estimator* (Pfeffermann 2002) and the *Hierarchical Bayes Estimator* (Rao 2011).

2.2.1.7 Regression-synthetic estimator

As an extension of the synthetic estimator, the regression synthetic estimator at small area level incorporates auxiliary information from predictor variables measured over the small domain (Rao & Molina 2015). The estimates of small area means are based upon the relationship between the variable of interest, Y_i and a vector of predictor variables, $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)^T$, measured over the small area (Levy 1979). When an area level random effect approach is used, the model equation for the small area means is of the form,

$$\hat{\bar{y}}_i \;=\; \boldsymbol{x}_i^T\,\hat{\boldsymbol{\beta}} \;+\; \nu_i \;+\; \epsilon_i\;, \tag{2.1}$$

(Ghosh & Rao 1994), where $\hat{\bar{y}}_i$ denotes the estimate for the i^{th} small area, and $\boldsymbol{x_i} = (x_{i1}, x_{i2}, \ldots, x_{ip})$ represents values of the predictor variables for the small area, and ν_i the i^{th} small area effect. The regression coefficients, $\hat{\boldsymbol{\beta}}$, which specify the effect of the predictor variables, $\boldsymbol{X}$, on Y, are estimated using data from the larger area (Levy 1979), such as a census or survey across the whole population. The two sources of error consist of sampling error, ϵ_i, and the modelling error resulting from the random effect at small area level, ν_i. The model described in Equation (2.1) is a special case of the linear mixed model (Quintano et al. 2007), since it includes random area-specific effects (Ghosh & Rao 1994). In a unit level model, the auxiliary data, $\boldsymbol{x}_{ij}$, are measured for each population unit, j, in the i^{th} small area (Rao 2003), to provide an estimate, $\hat{\bar{y}}_{ij}$, for the j^{th} population unit in the i^{th} small area, such that

$$\hat{\bar{y}}_{ij} \;=\; \boldsymbol{x}_{ij}^T\,\hat{\boldsymbol{\beta}} \;+\; \nu_i \;+\; \epsilon_{ij}\;. \tag{2.2}$$

The relationship of the variable of interest, $\hat{\bar{y}}_{ij}$, to the auxiliary data, $\boldsymbol{x}_{ij}$, is deemed to be via a nested error regression model (Singh et al. 1998). Unit level models have been developed which extend beyond a univariate response with normal distribution under a single stage sampling scheme (Rao 2003). In particular, many small area estimation analyses involve modelling variables of interest which have a non-normal distribution. The approach is to use a generalised linear mixed modelling, which is discussed in Section 2.2.1.12.

2.2.1.8 Best linear unbiased prediction estimator, BLUP

Henderson (1975) first developed the technique of best linear unbiased prediction estimation to allow for non-random sampling when mixed models were applied to animal breeding data. If the sampling errors, ϵ_{ij}, in Equation 2.1 are independent with mean zero and known variance σ_ϵ^2 and the model or area level effects, ν_i, are i.i.d. with mean zero and variance σ_ν^2, then the *Best Linear Unbiased Predictor*, BLUP, estimator to minimise the mean square error (MSE) is a composite estimator, constructed as a weighted average of the direct estimator and the regression synthetic estimator of the small area mean (Villa Juan-Albacea 2009).

2.2.1.9 Empirical best linear unbiased predictor estimator, EBLUP

The BLUP estimator works well when the variance components are known. This is not usually the case in practice. The solution is to replace σ_ν^2 and σ_ϵ^2 with sample estimates obtained through Maximum Likelihood Estimation (MLE), Restricted MLE, or Analysis of Variance (ANOVA) estimators (Prasad & Rao 1990). This gives rise to the *empirical best linear unbiased predictor estimator* or EBLUP.

2.2.1.10 Empirical Bayes estimator

The posterior distribution of a small area mean can be derived given the data and the model parameters, under the assumption that the model parameters are known. The empirical Bayes estimator of a small area mean is provided by the estimated posterior mean and its variability by the estimated posterior variance. All the model parameters can be estimated from the marginal distribution of the data (Coondoo et al. 2011), or, alternatively, only the variances are estimated from the data and the parameters for model coefficients are assigned uniform prior distributions over $(-\infty, +\infty)$, reflecting an absence of information about the model coefficients (Rao & Choudhry 1995).

2.2.1.11 Hierarchical Bayes estimator

The hierarchical Bayes approach is similar to the empirical Bayes method except that all three model parameters are assigned a prior distribution (Rao 2003). Inferences are then obtained from the posterior distribution of the small area mean estimator, as is achieved with the empirical Bayes technique.

Frequentist and Bayesian model-based estimation methods can be expanded using generalised linear mixed models, which extend the standard linear model to accommodate non-normal distributions of the response variable and also random effects components.

2.2.1.12 Generalised linear mixed models

The standard linear model having the form;

$$y \;=\; \beta + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon \;, \tag{2.3}$$

is appropriate only for data from a normal distribution. The generalised linear model, GLM (McCullagh & Nelder 1989), applies likelihood based methods to regression modelling for a non-normal response variable, Y: e.g. Poisson, Binomial, or Gamma. A GLM model comprises three characteristic features: the random component, systematic component, and a link between the random and systematic components (McCullagh & Nelder 1989). The random component, Y, is a response variable having a distribution from the exponential family, with the general form,

$$f\left(y|\theta,\phi\right) \;=\; \exp\left(\frac{y\theta \;-\; b(\theta)}{a(\theta)} \;+\; c(y,\phi)\right)\;, \tag{2.4}$$

for some specified functions $a(\cdot), b(\cdot)$ and $c(\cdot)$. The *canonical parameter*, θ, represents location, while ϕ, the *dispersion parameter*, represents scale. The expected value of the response Y, is $\mathrm{E}(y) = \mu$. The normal distribution, $N(\mu, \sigma^2)$, is also a member of the exponential family, with $\theta = \mu$ and $\phi = \sigma^2$. For the Poisson family, $\theta = \log(\mu)$ and $\phi = 1$. A Binomial distribution corresponds to $\theta = \log(\frac{\mu}{1-\mu})$ and $\phi = 1$.

The systematic component of a GLM constitutes a linear predictor, η, formed from covariates $x_1, x_2, \cdots, x_p$, such that,

$$\eta \;=\; \sum_{j=1}^{p} x_j\,\beta_j \;=\; \boldsymbol{x}^T\boldsymbol{\beta}\,.$$

The connection between the random and systematic components is provided by a function, $g(\cdot)$ which links the mean of the response with the linear predictor (McCulloch et al. 2008),

$$g(\mu) \;=\; \eta \;=\; \boldsymbol{x}^T\boldsymbol{\beta}\,.$$

For a distribution from the exponential family the link function corresponds to the canonical parameter, θ, so that $g(\mu) = \theta(\mu)$. When the response variable, Y, has a normal distribution, its expected value is linked to the linear predictor via the identity function , whereas for the Poisson distribution log function acts as the linking mechanism, and the logit and reciprocal link functions, respectively, are used with the Binomial and Gamma distributions.

Estimation of the fixed effects, the β's, for a generalised linear model is by maximum likelihood, using the method of iterative least squares (McCulloch et al. 2008). A measure of the suitability of a generalised linear model is via the deviance, which is defined as minus two times the log-likelihood ratio. For the Normal distribution, this quantity equates to the residual sum of squares. The generalised linear model can be extended to include random effects, so that,

$$g(\mu) \;=\; \boldsymbol{x}^T\boldsymbol{\beta} \;+\; \boldsymbol{z}^T\boldsymbol{b}\,, \tag{2.5}$$

where $\boldsymbol{x}^T$ is the design matrix for the fixed effects, $\boldsymbol{\beta}$ the vector of parameters for the fixed effects, the model matrix for the random effects is denoted by $\boldsymbol{z}^T$ and the random effects vector by $\boldsymbol{b}$. Generalised linear mixed models are useful for modelling overdispersion often found with Binomial and Poisson distributions, and also when the data are correlated (Breslow & Clayton 1993). Estimation of random effects having a Normal distribution is straightforward. Maximum likelihood based on the marginal distribution of the observations is applied, in which the random effects are "integrated out" (Schall 1991). However, for random effects with unspecified or non-normal distributions, the process of "integrating out" can be unfeasible numerically. In simple cases, for example a single

random effect in the model, numerical integration methods can be applied, such as the Gauss-Hermite quadrature technique (McCulloch et al. 2008). For more complex situations, a penalised quasi-likelihood (PQL) approach can be implemented. A quasi-likelihood is a quantity similar to a likelihood which requires few distribution assumptions, but does specify a relationship between the mean and variance of the response variable. Since the quasi-likelihood method doesn't provide a sufficient basis for estimating the covariance structure, a common approach is to add a penalty function to the quasi-likelihood quantity (Green & Silverman 1994). Another approach is marginal quasi-likelihood estimation (Sutradhar & Rao 2001), in which the joint moments of the clustered observations are calculated up to the fourth order.

Several authors have described the application of generalised linear models to small area estimation. Marker (1999) showed that many of the traditional methods of small area estimation can be classified as generalised linear models. Research by Noble et al. (2002) illustrated how structure preserving estimation can be expressed as a log-linear model. Ghosh et al. (1998) implemented hierarchical Bayes methodology for small area estimation, which included spatial generalized linear models. Folsom, R E and Shah, B V and Vaish, A K (1999) studied drug use using generalized linear mixed models with random effects specific to various age groups. Logistic generalized linear mixed models with random slopes were used by Malec et al. (1997) to study health related binary variables over small areas. In later research into overweight prevalence across population subgroups (Malec et al. 1999), similar hierarchical Bayes methods used pseudo-likelihood to incorporate survey weights. Schall (1991) modelled the salamander mating data described in McCullagh & Nelder (1989) using a logistic generalized linear mixed model with random effects representing male and female individuals. Zeger et al. (1988) used generalized estimating equations to model longitudinal data.

The consumption model in the ELL methodology involves a linear mixed model, comprising a normally distributed response variable, log per capita expenditure, and random cluster effects (Elbers et al. 2003). This is equivalent to the generalised mixed model of Equation (2.5) with the identity link function, $g(\mu) = \mu$.

2.2.2 Complex survey design

Utilising auxiliary information in the prediction process to ensure reasonable precision of estimates is an important aspect of estimation at small area level. Another aspect which must be considered is the mechanism of data collection. If the data includes any elements of complex survey design the effect of these must be factored into the estimation procedure. The essential feature of a sample survey is that it is representative of the population from which it was drawn, so that the estimator of the population parameter is unbiased and precise. This is achievable only through the employment of random selection in conjunction with a probability sampling scheme in the process of data collection (Neyman 1934). In probability sampling, every disparate sample has a known and fixed probability of being selected, and each unit of the population has a known and fixed probability of inclusion

in the sample. The four main types of probability sampling are simple random sampling, systematic sampling, stratified sampling and cluster sampling.

2.2.2.1 Simple random sampling

Simple random sampling is the simplest form of probability sampling and can be achieved with or without replacement of population units. The process involves random selection from the population of the elements of a sample, with the only condition being that each unit of the population has equal probability of selection (Cornfield 1944). Simple random sampling is an example of an *epsem* probability sampling scheme; an *equal probability selection method*. Each member of the population has the same probability of selection into the sample at every stage of the sampling process (Barnett 2002).

If $y_1, y_2, ..., y_n$ represent the n measurements taken on the units of a sample of size n, then the sample mean, $\bar{y}$, is an unbiased estimate of the population mean, μ (Cornfield 1944), where,

$$\hat{\mu} \;=\; \bar{y} \;=\; \frac{1}{n}\sum_{i=1}^{n} y_i \,. \tag{2.6}$$

An unbiased estimate of the variance of the sample mean, $V(\hat{y})$, is of the form (Lehtonen & Pahkinen 2004),

$$\hat{V}_{srs}(\bar{y}) = \frac{s^2}{n} \cdot \; f \,, \tag{2.7}$$

where,

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n} (y_i - \bar{y})^2 \;,$$

is an unbiased estimate of the population variance σ^2. The term f is known as the *finite population correction*. For simple random sampling without replacement, $f = \left(1 - \frac{n}{N}\right)$. When the sampling is with replacement, then $f = \left(1 - \frac{1}{N}\right)$. If the population size N is large with respect to the sample size, the usual situation with large scale sample surveys, then f becomes of little practical importance since its value is very close to 1. In a large population the precision of the estimator is determined by the size of the sample selected, not by the proportion of the population sampled (Lehtonen & Pahkinen 2004).

Simple random sampling without replacement is used as the reference by which other sampling schemes are compared (Lehtonen & Pahkinen 2004). The efficiency of a particular sampling scheme can be represented by the *design effect*, the ratio of variance under the sampling scheme to the variance under simple random sampling, specifically,

$$DEFF_{sampl}(\bar{y}) \;=\; \frac{\hat{V}_{sampl}(\bar{y})}{\hat{V}_{srs}(\bar{y})} \,.$$

A value of $DEFF$ less than one indicates a more efficient sampling scheme than that produced with simple random sampling.

Utilising simple random sampling to select a sample survey is usually time consuming and expensive, particularly if the population is very large. To simplify the data collection process, the strategies of systematic sampling, stratification and clustering can be employed. The application of survey design effects may alter the formulation of the sample mean as an unbiased estimator of the population parameter. The complexity of the survey design must also be reflected in the computation of standard errors (Kish & Frankel 1974).

2.2.2.2 Systematic sampling

A systematic sample is one in which the elements of the sample are selected in a regular manner from a listing of the population which is divided into separate classes for the selection process. If S_i denotes the systematic sample then an unbiased estimator of the population mean is the sample mean,

$$\bar{y} = \frac{1}{n} \sum_{i \in S_i} y_i \,, \tag{2.8}$$

and the variance of the sample mean has the form (Madow & Madow 1944),

$$\hat{V}_{sys}(\bar{y}) = \left(1 - \frac{n}{N}\right) \frac{s^2}{n} \left[\, 1 + (n-1)\, \rho_k \,\right] \tag{2.9}$$

where ρ_k is the intra-class correlation coefficient and s^2 is the standard deviation of the n sample elements, as previously defined. The *design effect* (Lehtonen & Pahkinen 2004), is the ratio of variance under systematic sampling to the variance under simple random sampling,

$$\begin{aligned} DEFF_{sys}(\bar{y}) &= \frac{\hat{V}_{sys}(\bar{y})}{\hat{V}_{srs}(\bar{y})} \\ &= 1 + (n-1)\, \rho_k \,. \end{aligned}$$

Hansen & Hurwitz (1942) provides examples in which the intra-class correlation, ρ_k, is negative, so that the systematic sampling process is more efficient than simple random sampling, i.e. $DEFF < 1$. When the population listing is in random order then $\hat{V}_{sys}$ is shown to be asymptotically equivalent to $\hat{V}_{srs}$ (Hartley 1966).

Before the probability schemes of stratification and clustering are outlined, some discussion on survey design weights is needed, as these are required to ensure the survey is representative of the population when stratification and clustering are part of the data structure.

2.2.2.3 Survey design weights

Sampling design weights are incorporated into a model for various reasons: when sample elements have different inclusion probabilities (Pfeffermann 1993), to compensate for

frame inequalities or non-responses in the survey, or as statistical adjustments such as post-stratification weighting (Kish 1990). The function of design weights is to monitor and reflect selection probability for each stratum and cluster at each stage of sampling. Disproportionate sampling fractions are often incorporated into a design in order to reduce variability and sampling costs, but need to be balanced by inverse weights so as to avoid bias in the resulting statistics (Kish 1990).

For some *epsem* samples, such as simple random sampling, systematic sampling and stratified sampling with proportional allocation, weighted and unweighted estimators coincide (Nathan & Holt 1980). However, sample weights are required in stratified and cluster sampling designs with disproportionate sampling fractions. Weighting can also be used to compensate for frame inequalities, non-response and modelling adjustments such as post-stratification weighting.

2.2.2.4 Stratified sampling

Stratified sampling involves dividing the population into disparate, but relatively homogeneous, strata, usually defined by a specific feature such as gender, age group, geographical region etc. A simple random sample is taken without replacement from each stratum (Cochran 1977). The sizes of the strata subsamples are often chosen using proportional allocation, with the subsample size being proportional to the size of the stratum.

A consequence of utilising proportional allocation is that each element of the sample represents the same number of population units and the probability of selection is now $\frac{n}{N}$, identical to a simple random sample (Lohr 1999). A stratified simple random sample chosen in this manner, selected using the principle of *probability proportional to size*, where size denotes the number of units in the stratum, is another example of an *epsem* sampling scheme. When proportional allocation is applied to the stratification design the sample is self-weighting. Every unit in the sample has the same weight and therefore represents the same number of units in the population.

In stratified sampling without proportional allocation, weightings are applied to each sample member to avoid bias in the resulting estimates. Given a population of size N, partitioned into H strata of size N_h with a sample size n_h taken from the h^{th} stratum, the estimate of the population mean under stratified sampling has the form,

$$\bar{y}_{str} \quad = \quad \sum_{h=1}^{H} \frac{N_h}{N} \, \bar{y}_h \, , \tag{2.10}$$

which can be regarded as a weighted sum of the estimated stratum means (Cochran 1977). Similarly, an unbiased estimate of the variance of the stratified sample mean is obtained through a weighted sum of the estimated variances within the strata, with weights $W_h^2 = \left(\frac{N_h}{N}\right)^2$, such that (Cochran 1977),

$$\hat{V}_{str}(\bar{y}) = \sum_{h=1}^{H} \left(1 - \frac{n_h}{N_h}\right) \left(\frac{N_h}{N}\right)^2 \frac{s_h^2}{n_h} \text{,} \tag{2.11}$$

where $\left(1 - \frac{n_h}{N_h}\right)$ represents the finite population correction factor for the h^{th} stratum and s_h^2 is the variance of the subsample taken from the h^{th} stratum and has the form,

$$s_h^2 = \sum_{j \in S_h} \frac{(y_{hj} - \bar{y}_h)^2}{n_h - 1} \,.$$

As with systematic sampling the design effect for stratification is the ratio of the variance under stratification to the variance from simple random sampling, given by

$$DEFF_{str}(\bar{y}) = \frac{\hat{V}_{str}(\bar{y})}{\hat{V}_{srs}(\bar{y})} \,.$$

If the strata are homogeneous then s_h^2 will be small. Consequently, the design effect for a stratified sample is usually less than one, reflecting the extra efficiency of stratification in estimating the sample mean. Another advantage of stratification is that it can ensure representation of small subgroups in the population (Lehtonen & Pahkinen 2004).

2.2.2.5 Cluster sampling

A cluster is, like a stratum, a group of members of the population. But rather than being based upon a demographic, geographical or socio-economic attributes clusters are generally chosen for administrative convenience. A typical cluster might consist of a block of households in a city or a grouping of villages in a rural area. Once the clusters have been identified a simple random sample of clusters is taken (Lehtonen & Pahkinen 2004). In a single stage clustering design all units in the selected clusters are sampled. In more complex cluster sampling processes the cluster acts as the primary sampling unit (psu) with additional sampling undertaken using, for example, systematic sampling of each chosen cluster to select the final sample. Clusters tend to consist of fairly similar elements. For example, households close together in a village are more likely to be alike than households in a distant village. The intra-cluster correlation, an indication of the homogeneity of clusters, must be accounted for in the modelling (Lohr 1999).

Weighting is required for two-stage cluster sampling and single-stage cluster sampling with unequal numbers of elements in each cluster. The estimate for the population mean under cluster sampling can be written as (Lohr 1999),

$$\bar{y}_{cl} \quad = \quad \frac{M}{Nm} \sum_{c \in S} N_c \, \bar{y}_c \,, \tag{2.12}$$

where N is the population total, M denotes the number of clusters, or psu's, in the

population, m denotes the number of psu's in the sample, and N_c the size of the c^{th} psu. The sample mean for the c^{th} cluster, $\bar{y}_c$ is given by,

$$\bar{y}_c = \frac{1}{n_c} \sum_{k \in S_c} y_{ck} \, .$$

So, the estimate of the population mean under cluster sampling can be considered as a weighted sum of the estimated cluster means. An unbiased estimate of the variability of the sample mean is of the form (Lehtonen & Pahkinen 2004),

$$\hat{V}_{cl}(\bar{y}) = \frac{1}{N^2} \left[M^2 \, \frac{M-m}{M} \, \frac{s_b^2}{m} + \frac{M}{m} \sum_{c \in S} N_c^2 \, \frac{N_c - n_c}{N_c} \, \frac{s_{w_c}^2}{n_c} \right] \, , \tag{2.13}$$

where,

$$s_b^2 = \frac{1}{m-1} \sum_{c \in S} \left(N_c \, \bar{y}_c - \frac{N \bar{y}}{M} \right)^2 \quad \text{and} \quad s_{w_c}^2 = \frac{1}{n_c - 1} \sum_{k \in S_c} (y_{ck} - \bar{y}_c)^2 \, .$$

The terms s_b^2 and $s_{w_c}^2$ above denote the between cluster and within cluster sampling variance respectively. Thus the estimate of variance for the sample mean in a cluster design is a weighted sum of the within and between cluster variances. The design effect can then be expressed in terms of the intra-cluster correlation coefficient, ICC, expressed as

$$DEFF_{cl}(\bar{y}) \quad = \quad \frac{\hat{V}_{cl}(\bar{y})}{\hat{V}_{srs}(\bar{y})} \quad = \quad \frac{M \bar{N}_c - 1}{\bar{N}_c \, (M-1)} \, \left[1 + \left(\bar{N}_c - 1 \right) ICC \right] \, , \tag{2.14}$$

where the intra-cluster correlation coefficient ICC, can be expressed in terms of the ANOVA within sum of squares SSW, and total sum of squares, $SSTO$,

$$ICC \quad = \quad 1 - \frac{\bar{N}_c}{\bar{N}_c - 1} \, \frac{SSW}{SSTO} \, .$$

This section on complex survey design has discussed probability sampling and the different aspects that comprise a sample design, stratification, clustering and weighting. The complex survey design elements in the Nepal survey data, which are used to build tree models in the thesis, are discussed in the next section.

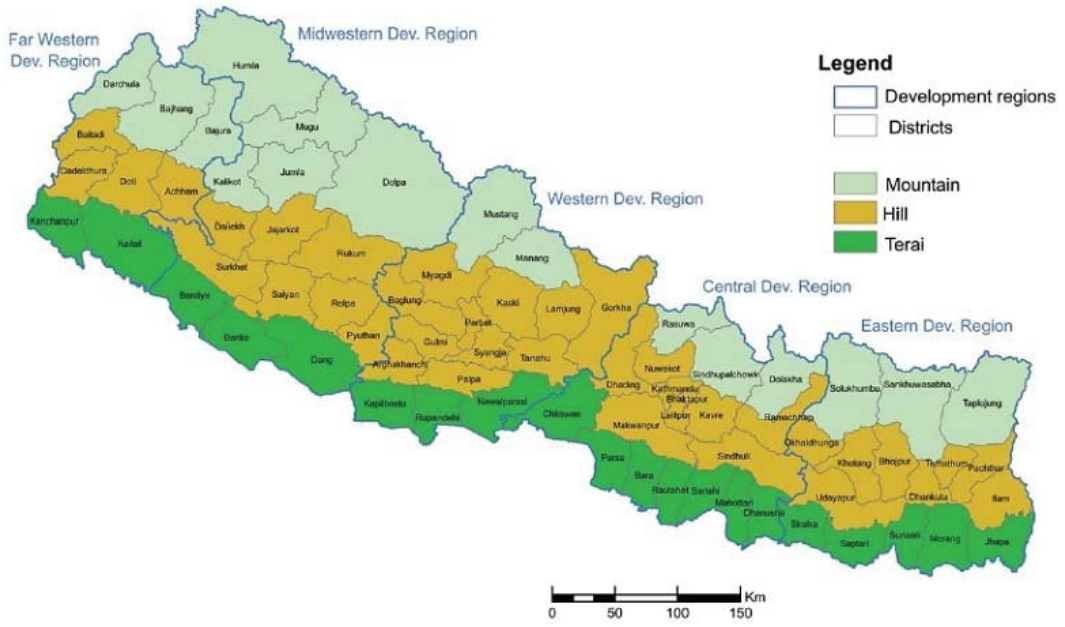
2.2.2.6 Complex survey design for Nepal

The survey design for the analysis of poverty in Nepal using the ELL methodology included stratification, clustering, systematic sampling and weighting. The application of these aspects of complex survey design was determined by the geographic and administrative features of Nepal. The nation of Nepal has an interesting geographical structure.

It comprises three disparate ecological zones, mountains, hills and terai (the lowlands), running transversally across the country. Intersecting these ecological zones are five development regions; Eastern, Central, Western, Mid-Western and Far-Western. Spread across the five development regions are seventy five districts. The division of Nepal into these zones, regions and districts is illustrated in Figure 2.1.

Each district comprises village development committees (VDC's) in rural areas and municipalities in urban areas (Haslett & Jones 2006). The smallest administrative unit is a ward; a VDC has nine wards, but municipalities can contain more than nine wards. To facilitate the construction of census and survey sampling frames, larger wards were divided into sub-wards, but these sub-wards usually did not have well defined boundaries.

Figure 2.1: Geographical and administrative divisions in Nepal



Another administrative feature, domains known as *ilakas* and defined by electoral boundaries, provided the target small areas for estimation purposes. The *ilakas* comprise groups of VDCs in rural areas and municipalities in urban areas. For the purpose of poverty mapping in Nepal with ELL, the *ilakas* were redefined - the rural part of each original *ilaka* was retained, and the municipalities within an *ilaka* combined to form a new, separate *ilaka*. Thus the domains for estimation became either groups of rural VDCs or urban municipalities.

The data from two separate surveys were utilised in estimating indicators of well being in Nepal. Using information gleaned from a living standards survey (Central Bureau of Statistics, Nepal 2004a,b), predictions of per capita household income were constructed to provide estimates of poverty measures, and kilocalorie intake was converted to indicators of undernourishment. Measures of malnutrition, wasting, stunting and underweight for children under five were extracted from data provided by a demographic and health survey (Nepal Demographic and Health Survey 2001).

The sample design for the living standards survey comprised a two-stage stratified random sampling approach (Central Bureau of Statistics, Nepal 2004a,b). Six strata were constructed from an interaction of ecological zones and rural/urban characteristics. They comprised Mountains, Kathmandu Urban, Other Urban Hills, Rural Hills, Urban Terai and Rural Terai. In the first stage of sampling 334 primary sampling units (psu's) were chosen using stratified random sampling with probability proportional to size. Each psu, or cluster, corresponded approximately to wards in rural areas and sub-wards in municipalities. The measure of size was the number of households. At the second stage of the process twelve households were selected by systematic sampling from each psu. This provided a self-weighting sample (before adjustments for non-response). Due to political unrest enumeration could not be conducted in 8 of the psu's, reducing the number of households surveyed to 3912. The target variable for modelling poverty measures was log annual per capita expenditure. The variable of interest for measures of undernourishment was log kilocalorie intake, adjusted to be per adult equivalent.

For the demographic and health survey the sample design included thirteen strata (Nepal Demographic and Health Survey 2001). These were constructed from the intersection of ecological belts with development regions, but with the mountainous areas of the Western, Mid-western and Far-western zones amalgamated into a single stratum. The analysis of malnutrition indicators involved only households with children under five. The modelling was carried out using data from 5883 children in 4001 households from 241 psu's. The households represented the secondary sampling units, ssu's.

The research into developing tree based models for poverty mapping in Nepal utilises the same datasets involved in the ELL estimation of poverty measures in Nepal. The survey dataset comprises 326 ilakas, the clusters, each having 12 households, a total of 3912 household units. Modelling is carried out at household level, but prediction is made at individual level with the employment of survey weights (see Section 1.2), and then aggregated for a small area estimate. The stratification used in the modelling comprises the six regions of Mountains, Katmandu Urban, Other Urban Hills, Rural Hills, Urban Terai and Rural Terai.

A key component of the thesis is estimation of standard errors of prediction. Mathematical equations for variance of the sample mean or proportion, such as those displayed in Equation (2.11) and Equation (2.13), are applicable only when the sample design comprises a single design element, such as stratification or clustering. With a complex survey design comprising more than one element of survey design and several sampling stages, in many cases the variability of the sample mean does not have a tractable mathematical form. Another method for variance estimation must be used in order to provide estimates with reasonable precision. Several different techniques for estimating the variability of estimators of population quantities of interest are discussed in the next Section.

2.2.3 Variance estimation for complex survey design

It is usually not satisfactory to obtain only point estimates of population parameters, means, totals or proportions, from sample survey data. The precision of such estimates is also required. When the sample design comprises a single design element, population means and their variances can be estimated using mathematical equations which incorporate the appropriate sampling weights. Complex survey structures arise when more than one of the survey design elements of stratification, clustering and weighting are included in the survey design used to collect the data. For a complex survey design comprising more than one stage of sampling, all the information required to build a point estimate is provided by amalgamating the weightings for each individual stage of sampling (Lohr 1999). A sample weight for a particular element of the sample is the inverse of the inclusion probability of that sample member at the given stage of sampling.

With a complex survey design comprising more than one design element and sampling stages, the variability of the sample mean does not have a tractable mathematical form. A variance estimation method must be used in order to provide poverty estimates which have reasonable precision. Various approaches to the estimation of variability under a complex survey design have been developed, most involving some type of subsampling to provide multiple estimates of the population parameter of interest. The average of these multiple estimates provides a point estimate of the population parameter of interest, $\hat{\theta}$. The variance of $\hat{\theta}$ can be estimated from the variability of the multiple estimates. Several different variance estimation methodologies, developed to evaluate the variability of estimators of population quantities of interest, are outlined in the following sections. These methods include, replication, balanced repeated replication, jackknife, bootstrap, linearisation using Taylor Series and inverse sampling.

2.2.3.1 Replication method

Replication is considered the simplest subsampling method by Thompson (1997). Wolter (2007) referred to it as the method of *random groups* and the use of the term *the method of replicate samples* is attributable to Deming (1956), amongst other authors. The concept of replicate samples, originally mooted by Mahalanobis (1946), involves drawing, independently and with replacement, random samples which replicate the survey design. In actual practice, these replicate subsamples are not drawn independently, but a single sample is selected using the survey design and this complete sample is divided into subsamples, each of which mirrors the survey design (Lohr 1999). The subsamples are then analysed as though they were independent replicates of the original survey design. Each replicate sample can provide an independent estimate of the population parameter of interest, $\hat{\theta}$. The variance of $\hat{\theta}$ can be estimated from the variability of the sample replicates. Given t replicate estimates, $\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_t$, with mean $\bar{\theta}$, the commonly used formula to estimate the variance of the parameter $\hat{\theta}$ is (Kalton 1983),

$$\hat{V}_{Rep}\left(\hat{\theta}\right) = \frac{\Sigma\left(\hat{\theta}_i - \bar{\theta}\right)^2}{t(t-1)}\,, \tag{2.15}$$

where $\bar{\theta}$ is the mean of $\hat{\theta}_i$. These pseudo-random groups are not quite independent replicates because an observation unit can be selected for only one group. However, the groups can be treated as approximately independent replicates if the sample size is small relative to the population size (Wolter 2007).

Other resampling techniques examined are balanced repeated replication, jackknife and bootstrap. These methods of variance estimation are more correctly called *pseudoreplication* replication methods (Lee & Forthofer 2006), since they involve reuse of data from a single sample rather than drawing several independent samples from the population (Lehtonen & Pahkinen 2004).

2.2.3.2 Balanced repeated replication

The technique of balanced repeated replication was first proposed by McCarthy (1969). The methodology requires a survey design consisting of the maximum number of strata with only two primary sampling units, psu's, per stratum. This structure affords an optimal level of stratification with the minimum number of psu's to provide a valid variance estimator (Kovar, J.G., Rao, J.N.K and Wu, C.F.J 1988). Two pseudo-replicates, or half-samples, are then formed by including one psu from each of the stratum in one pseudo-replicate, and the remaining psu in the other pseudo-replicate. A set of balanced half samples can be constructed consisting of different psu's from the strata at each new selection. To ensure unbiased estimates, an orthogonal matrix design proposed by Plackett & Burman (1946) is used to balance the construction of the pseudo-replicates.

2.2.3.3 Jackknife resampling

In jackknife resampling, a pseudo-replicate is formed by excluding one or more observations from the original survey sample. The concept of jackknife sampling, creating subsamples by excluding some members of the sample, was first used by Quenouille (1949) as a non-parametric estimate of bias. Initially, Quenouille split the data into only two half samples. In later work (Quenouille 1956), he extended the number of groups, establishing the method referred to as the "grouped jackknife" (Efron 1982). In the simplest form of this technique, *delete-1 jackknife*, a single observation is deleted from the original sample to create the pseudo-replicate.

Suppose we have n observations $y_1, y_2, \ldots, y_n$ of n independent and identically distributed random variables $Y_1, Y_2, \ldots, Y_n$. Suppose further that the data can be divided into g groups, each of size m, so that $n = gm$. The composition of the groups may be determined by the structure of the data, or may be an arbitrary assignment (Miller 1968). Let $\hat{\theta}$ be an estimator of the parameter θ based on the sample of size n. Let $\hat{\theta}_{(j)}$ represent the corresponding estimator of θ based on the sample of size $(g-1)m$, in which the i^{th}

group of size m has been omitted. Also denote the average of the estimates $\hat{\theta}_{(j)}$ by $\hat{\theta}_{(\cdot)}$. Quenouille's bias-corrected "jackknife estimate" of $\hat{\theta}$, referred to as "pseudo-values", has the form,

$$\tilde{\theta}_j \;=\; g\hat{\theta} \;-\; (g-1)\,\hat{\theta}_{(j)}\,,$$

and, $\tilde{\theta}_{\cdot}$ denotes the average of the bias corrected estimates $\tilde{\theta}_j$ where

$$\tilde{\theta}_{\cdot} \;=\; g\hat{\theta} \;-\; (g-1)\,\hat{\theta}_{(\cdot)}\,.$$

Tukey (1958) proposed in an abstract that the bias corrected jackknife estimates, $\tilde{\theta}_j$, could be considered as approximately independently and identically distributed, and thus provide a non-parametric estimate of variance for building confidence intervals. He is credited with introducing the term "pseudo-values" and attributing the name "jackknife" to the technique (Miller 1974). The jackknife estimate of variance can be expressed in terms of the pseudo-values (Efron 1980), but then be reformulated in terms of $\hat{\theta}_{(j)}$, as follows

$$\begin{aligned} Var_{JK}(\hat{\theta}) \;&=\; \frac{1}{g(g-1)} \sum_{j=1}^{g} \Big(\tilde{\theta}_j \;-\; \tilde{\theta}_{\cdot} \Big)^2 \\ &=\; \frac{g-1}{g} \sum_{j=1}^{g} \Big(\hat{\theta}_{(j)} \;-\; \hat{\theta}_{(\cdot)} \Big)^2 , \end{aligned} \tag{2.16}$$

the latter being the more commonly used form of the jackknife estimate of variance.

The jackknife technique resembles the method of replication, or random groups, but without specifically replicating the sample design, and is useful when the sample design is unknown (Lee & Forthofer 2006). Moreover, the jackknife variance estimator has been shown to be asymptotically normally distributed as $n \to \infty$ (Miller 1968). The jackknife technique has been usefully applied to variance estimation of ratio statistics, transformed statistics and for maximum likelihood estimation (Miller 1968). However, the jackknife method is unsatisfactory for some other types of estimators, for example a median and quantiles (Shao & Tu 1995). The next section outlines the bootstrap, another resampling technique for variance estimation. The bootstrap methodology can be applied more widely than the jackknife, which can be considered as a linear approximation of the bootstrap (Efron & Tibshirani 1993). For linear statistics there is no loss of information by using the jackknife, but for non-linear statistics there is a loss of information. If a large enough number of bootstrap samples are generated, then the units in the original sample are asymptotically selected an equal number of times, so the limit on the bootstrap is the jackknife.

2.2.3.4 Bootstrap resampling

The Bootstrap method for variance estimation was first introduced by Efron (1979), as an alternative to, and a more general example of, the jackknife procedure. The concept of bootstrapping conveys the idea of "pulling oneself up by one's bootstraps" (Dictionary.com 2011). The original sample, of size n, is treated as though it was a population and bootstrap pseudoreplicates, also of size n, are obtained by sampling with replacement from the initial survey sample. Consider a survey sample consisting of n independently and identically distributed (i.i.d.) observations $y_1, y_2, \ldots, y_n$. Suppose that B bootstrap replicate subsamples are generated, of the form $(y_1^*, y_2^*, \ldots, y_n^*)^T$. Each pseudoreplicate provides an estimate, $\hat{\theta}_b^*$, of the estimator $\hat{\theta}$ relating to the quantity of interest. A point estimate for the population parameter is provided by the average of the pseudoreplicate estimates,

$$\hat{\theta}^* = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}_b^* \, .$$

A bootstrap estimate for the variance of $\hat{\theta}$ is given by,

$$\hat{V}_{Boot} \;=\; \frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{\theta}_b^* \;-\; \hat{\theta}^* \right)^2 \, . \tag{2.17}$$

Although Efron (1980) specifies $B-1$ in the denominator of the multiplicative factor in Equation (2.17), some authors use B instead (Rao 2007, Sitter 1992). When bootstrap resampling in a finite population, with sample of size n and sample standard deviation s^2, to obtain a unbiased bootstrap estimator of variance, i.e. so that

$$\hat{V}_{Boot}\left(\bar{y}\right) \;=\; \frac{s^2}{n} \, ,$$

the size of the bootstrap sample must be $n-1$ (Wolter 2007).

One advantage of the bootstrap over the delete-1 jackknife procedure is that the number of disparate pseudoreplicates that can be generated is much greater than the sample size n. The development of very fast computer technology facilitates the application of the bootstrap. DiCiccio & Efron (1996) describe bootstrapping as an automatic algorithm for estimating variability. Other forms of the bootstrap have been developed, and Efron's original bootstrap method is now referred to as the "naive bootstrap" (Efron & Stein 1981). These other forms include the Bayesian bootstrap (Rubin 1981), the smoothed bootstrap method (Efron 1982), the double bootstrap Efron (1983) and the *m-out-of-n* bootstrap method developed by Bickel et al. (1997). A parametric form of the bootstrap (Efron 1982) can be used when the data is assumed to to be i.i.d. from a particular probability distribution.

The techniques for variance estimation discussed in the preceding sections, replication or random groups, balanced repeated replication, jackknife and the bootstrap, are examples of non-parametric methods of generating variance estimates, since sampling is

conducted with replacement from an empirical distribution. These four resampling methods apply a similar approach to the task of estimating the variance of a non-linear estimator. Several different replicates of the original sample are generated to provide multiple estimates of the population parameter of interest. The variance of the parameter being studied is obtained from the variability of these replicate estimates. An alternative to using resampling for variance estimation, linearisation of the non-linear estimator, is briefly discussed in the next section.

2.2.3.5 Taylor Series Method

Taylor Series linearisation offers an alternative to using resampling for variance estimation. Also known as the Delta method (Efron 1982), the Taylor Series Linearisation procedure provides a linear approximation to a non-linear estimator of a population quantity of interest. Standard variance estimation techniques can then be applied to the linear approximation of the estimator to extract its variance. The method is based upon the Taylor Series expansion (Stein 1987) used extensively in Mathematics.

The Taylor Series linearisation method has been used in many different situations to estimate the variance of complicated estimators, including in the methodology of poverty mapping. Keyfitz (1957) applied the method to estimation of variance under stratification, with each stratum comprising only two units. Verma & Betti (2011) explored the application of Taylor linearisation to the variance of poverty measures, and compared the technique with the Jackknife Repeated Replication method. An empirical study was conducted by Betti & Ballini (2008) who constrasted the effectiveness of Taylor linearisaton and Jackknife Repeated Replication in estimating the variance of measures of poverty and inequality in Albania.

The usual non-parametric Delta method estimate of standard error using a Taylor series expansion of $\hat{\theta}$ has been shown by Efron & Stein (1981) to be identical to Jaeckel (1972)'s infinitesimal jackknife. The Delta method is closely related to both the jackknife and the bootstrap, but in many situations it badly underestimates the standard error (Lepage & Billard 1992). One disadvantage of the Delta method is that evaluation of partial derivatives may prove problematic for complex statistics (Krewski & Rao 1981).

The discussion involving the various resampling schemes for variance estimation has so far focused on data collected through simple random sampling. We now consider the extension of jackknife and bootstrap methods to data with a complex sample survey structure.

2.2.3.6 Jackknife and bootstrap for complex data

Variance estimation methods applied to population parameters of data with a complex survey structure must account for the elements of stratification and clustering present in the data. In this section we discuss jackknife and bootstrap variance estimators for complex survey data. The variance estimation method of random groups can be difficult

to apply with complex data, since the design structure of each random group should mimic that of the original sample (Lohr 1999). Balanced repeated replication is only applicable when each stratum contains exactly two clusters.

When stratification is present, the jackknife is applied to each stratum. Extension of the jackknife to a cluster sample requires using a delete-cluster jackknife. Excluding a complete cluster at each iteration of the jackknife, rather than a single observation, is necessary to retain the cluster structure, preserving the dependence between units within the same cluster (Lohr 1999). Thus, in a stratified multistage cluster sample, a separate jackknife estimate is obtained from each stratum at the first stage of sampling, by deleting one cluster at a time. The variance estimator of $\hat{\theta}$ under stratification has the form (Krewski & Rao 1981),

$$Var_{JK}(\hat{\theta}) \;=\; \sum_{h=1}^{H} \frac{(n_h \;-\; 1)}{n_h} \sum_{i=1}^{n_h} \left(\hat{\theta}_{-hi} \;-\; \hat{\theta}\right)^2 , \tag{2.18}$$

for H the number of strata, n_h observations in the h^{th} stratum and $\hat{\theta}_{-hi}$ an estimator corresponding to $\hat{\theta}$ but based on the sample constructed by omitting the i^{th} observation from stratum h. When clustering is also present in the data, the jackknife estimates, $\hat{\theta}_{-hi}$, are generated by excluding the i^{th} cluster rather than the i^{th} observation (Lohr 1999). In this scenario, n_h denotes the number of clusters in the stratum, rather than the number of individual units.

When applying the bootstrap to stratified data, the bootstrap replicate should comprise a stratified subsample from the original parent sample. So the bootstrap estimate of variance for a stratified sample has the form,

$$\hat{V}_{Boot} \;=\; \frac{1}{H} \sum_{h=1}^{H} \frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{\theta}^*_{hb} \;-\; \hat{\bar{\theta}}^*_h\right)^2 ,$$

where $\hat{\theta}^*_{hb}$ denotes the bootstrap estimate from the b^{th} bootstrap replicate in the h^{th} stratum, and $\hat{\bar{\theta}}^*_h$ the mean of bootstrap estimates, $\hat{\theta}^*_{hb}$, in the h^{th} stratum

A bootstrap sample which replicates the original probability sampling design should be generated within each stratum (Shao 2003). One method to achieve this is to use a bootstrap size of $n^*_h = n_h - 1$ in the h^{th} stratum, where n_h denotes the number of observations in the h^{th} stratum (McCarthy & Snowden 1985). The weight, w_{hi} for y_{hi}, the i^{th} observation in the h^{th} stratum, is then adjusted by a factor of $\frac{n_h}{n_h-1}$ to compensate for the bootstrap sample being smaller than the stratum size, and the bootstrap estimate computed using these adjusted weights. Another approach, proposed by Rao & Wu (1988) rescales the original bootstrap sample. When the size of the bootstrap replicate is $n^*_h = n_h - 1$, this procedure is equivalent to the method of McCarthy and Snowden.

In a multistage sampling scheme, where psu's within each stratum are selected using probability proportional to size, the cluster bootstrap is employed in each stratum, so that the psu's are bootstrapped rather than individual observations. All second stage

and third stage sampling units from each selected psu should be included in the bootstrap replicate. To provide an unbiased estimator, the size of the bootstrap replicate should be $n_h^* = n_h - 1$, where n_h is the number of psu's in the stratum (Wolter 2007).

When either stratification or clustering are present in the data, design weights must also be included in the modelling, to ensure that the sample is representative of the population. When applying the jackknife and bootstrap methods, the design weights must be amended to preserve correct representation. The total weight for a replicate subsample taken within a stratum is adjusted to equal the actual total weight for that stratum (Rao & Wu 1988).

2.2.3.7 Inverse sampling

Section 2.2.3.1 discussed the resampling method known as *replication* or *random groups*. A replicate is selected to be a subsample of the original sample that reflects the survey design structure. If the replicate subsamples are drawn independently using the same design for sample selection then each replicate can contribute an independent estimate of the parameter of interest, θ. The variability of these replicate estimates then provides an estimate of the variance of θ based upon all subsamples. The usual approach for replicates samples when clusters form part of the data structure is to divide the clusters among the replicates, each cluster taking all its observation units into the assigned replicate, so that the clustering structure is retained in each replicate. An alternative approach when the data is clustered is to utilise inverse sampling. Most fields of applied statistics are concerned with fitting complicated models using sophisticated methods but assume that the sampling structure of the data is very simple. Practitioners in survey sampling have tended to focus on the estimation of simple quantities, population totals, means and proportion, utilizing samples selected by complicated but efficient processes. When complex survey data is used to fit complicated models, the application of standard analytical methods is problematic (Skinner et al. 1989). A solution is to work in reverse; select the sample to fit the method rather than adapting the method to fit the data (Rao & Scott 2000).

The concept of inverse sampling, as proposed by Hinkins et al. (1997), involves selecting a subsample whose structure is akin to simple random sampling. From the original complex sample drawn from the population, select a subsample which inverts the original complex sample selection process to provide a simple random sample. Repeating this process several times and averaging the results reduces any loss of efficiency.

Suppose that g inverse subsamples are generated to estimate the parameter θ. Let $\hat{\theta}_j^*$ and $\hat{V}_j^*$ represent respectively the estimate of θ and its variance derived from the j^{th} inverse subsample. The inverse sampling estimate of θ is given as (Rao & Scott 2000),

$$\hat{\theta}_{IS} \;=\; \frac{1}{g} \sum_{j=1}^{g} \hat{\theta}_j^* \,, \tag{2.19}$$

and the estimator of variance from the inverse samples is,

$$\hat{V}_{IS} = \frac{1}{g}\sum_{J=1}^{g}\hat{V}_j^* - \frac{\sum_{j=1}^{g}\left(\hat{\theta}_j^* - \hat{\theta}_{IS}\right)^2}{g}. \tag{2.20}$$

The form of the variance estimator reflects the fact that the estimates generated from individual inverse subsamples are conditional on the original sample survey.

Rao & Scott (2000) discusses an application of inverse sampling to clustered data proposed by Hoffman & Weinberg (1998), in which an inverse subsample is constructed by selection of a single observation at random from each cluster in the original complex survey sample. The process can be repeated many times, and each inverse subsample comprises a set of independent observations. Rao et al. (2003) outline how inverse sampling algorithms can be applied for different complex survey sampling designs.

2.3 ELL methodology for poverty mapping

One purpose of the research is the adaptation of ELL methodology for poverty estimation using tree models rather than the linear mixed model employed by the ELL method. Various techniques have been applied in different countries to provide small-area estimates of poverty measures. Examples includes augmentation of survey data with census information to produce poverty maps in Ecuador (Hentschel et al. 2000); applying principal component analysis to poverty measures in Cambodia (Fujii 2008); *Empirical Best Linear Unbiased Predictor* estimates of poverty in Italy (Quintano et al. 2007); graphically weighted regression techniques to investigate spatial relationships of poverty in Bangladesh (Kam et al. 2005). Traditional small area estimation techniques directly model the population characteristic being estimated. The ELL technique is one example among many small area estimation methods. It differs from other approaches to small area estimation with respect to the quantity of interest being modelled, and the inclusion of auxiliary information. ELL was developed to model an underlying continuous variable which could then be translated into indicators of well being using non-linear functions (Haslett & Jones 2006). Precision of estimates across small domains are improved by generating predictions for all households in the census, using census variables which match the survey variables used to build the model.

The ELL methodology for poverty estimation employs a linear mixed model (Section 2.2.1.12) based on the unit level model of Equation (2.2) with random effects. It differs from the usual approach to small area estimation in several respects. Firstly, the ELL technique doesn't model the quantity of interest directly, but utilises underlying continuous variables of interest which are then transformed into measures of deprivation. In contrast to other model based estimation methods, such as EBLUP (Section 2.2.1.9), the random effects in the ELL model are not at small area level, but represent variability for clusters within areas, and households within clusters (Haslett et al. 2010). The appropriate area level for estimation purposes can be decided post hoc, by determining the lowest level

of disaggregation that provides reasonable standard errors of prediction (Alderman et al. 2002). ELL also differs from other small area estimation techniques in that it generates predictions at unit level, then aggregates these predictions to the desired area level.

Since poverty mapping is used to guide the distribution of aid resources by local organisations, the level of the geographic unit at which allocation is made is an important issue (Baker & Grosh 1994). For many years poverty was estimated only for large domains. For example, World Bank Living Standard Measurement Surveys provided estimates barely extending beyond a simple division of urban and rural within broad regions in a particular country (Alderman et al. 2002). Elbers et al. (2007) found greater alleviation of poverty through targeting small administrative units, i.e. districts and villages. Disaggregation at even lower levels is possible when information on income and consumption is available at household level.

In the Nepal analysis, small area estimates of indicators of well-being such as poverty, caloric intake and malnutrition in children were generated at regional, district and sub-district level (Haslett & Jones 2006). The units of disaggregation at sub-district level were *ilakas*[1].

The ELL linear mixed model, built from the survey data, has the form (Elbers et al. 2003);

$$\begin{aligned} Y_{ij} &= E\left[Y_{ij} \mid \mathbf{x}_{ij}\right] \\ &= \mathbf{x}_{ij}^T \boldsymbol{\beta} + \gamma_j + \epsilon_{ij} \,, \end{aligned} \tag{2.21}$$

where Y_{ij} denotes the measurement on the i^{th} household in the j^{th} cluster, and $\mathbf{x}_{ij}$ is the vector of values of predictor variables from the survey dataset measured for the i^{th} household in the j^{th} cluster. The vector $\boldsymbol{\beta}$ represents the regression coefficients to be estimated, which describe the effect of the survey predictor variables, $\mathbf{x}$, on the variable of interest, Y. The term γ_j denotes the common error term for the j^{th} cluster, and the household-level error within the cluster is represented by ϵ_{ij}. Each of the the error terms, γ_j and ϵ_{ij}, is assumed to have some distribution with mean of zero (Elbers et al. 2003). In addition, γ_{cj} and ϵ_{ij} are assumed to be independent of each other and uncorrelated with the observed variables, $\mathbf{x}_{ij}$. The $\boldsymbol{\beta}$'s represent the variation in Y attributable to its relationship with $\mathbf{x}$, the set of predictor variables, while γ and ϵ encapsulate the unexplained variability.

Estimates of the variance components at cluster and household levels, σ_c^2 and σ_ϵ^2, are obtained from the cluster level effects, $\hat{\gamma}_j$, and household level effects, $\hat{\epsilon}_{ij}$ respectively (Elbers et al. 2003). The overall standard error of small area estimates includes error at cluster and household level, so it is desirable to minimize both of these sources of variation. Since the number of households exceeds the number of clusters (Haslett et al. 2010),

[1]The *ilakas* comprise VDCs in rural areas and municipalities in urban areas. For modelling purposes the *ilakas* were redefined - the rural part of each orignal *ilaka* was retained, and the municipalities within an *ilaka* combined to form a new, separate *ilaka*. Thus the domains for estimation become either rural VDCs or urban municipalities.

for example an average of 15.5 households and 1.25 clusters per ilaka in the Nepal survey dataset, it is especially important that variability at cluster level be smaller than that at household level. This may be achieved by efficient selection of auxiliary variables. In particular, the inclusion in the model of location variables, including census means, can help to reduce the size of the cluster and household contribution to variability. These auxiliary variables are a key component of the modelling, since they contribute extra information so that small area estimates having adequate degree of precision can be obtained (Rao & Molina 2015). The process by which auxiliary information in incorporated into the ELL methodology is discussed in the next section

2.3.1 Auxiliary information

The approach used in the ELL method for incorporating auxiliary information, in order to improve the precision of estimates (Section 2.2.1), is to supplement survey data with census and geographical information. The survey data contributes detailed information, and extensive coverage is provided by census and geographic information (Alderman et al. 2002). The predictors for the linear mixed model comprise a set of auxiliary variables, $\mathbf{x}$, common to a survey and a population census (Elbers et al. 2003). Consistency of definition and measurement is important when drawing these common variables from the different data sources. Additional predictors are provided by including geographical information encapsulated in location variables (Elbers et al. 2003). In the Nepal context, these consisted of geographical indicators at VDC or ilaka level, such as mean elevation above sea level, population density, etc. Location variables were also created by taking census means, as regional averages, at VDC or ward level, of population variables not already included in the set of common variables, i.e. for which a match within the survey dataset could not be found. The location variables are required in order to capture unobserved geographical effects not incorporated in the household variables (Elbers et al. 2003). Supplementary variables incorporated into the set of common predictors for the ELL modelling of poverty in Nepal were constructed by transformations on existing auxiliary variables, or through interactions between pairs of auxiliary variables. This approach, of supplementing survey information with census variables, is a feature of the ELL methodology that differentiates it from other small estimation techniques for poverty measures.

The survey data from which the ELL regression model of Equation (2.21) is constructed generally has a complex design survey structure. The features of complex design, clustering, stratification and weighting, need to be incorporated into the ELL modelling, and some type of variance estimation applied for generating standard errors of prediction. These two issues are discussed in the following sections.

2.3.2 Complex survey design

Fitting of the ELL model must also account for the complex survey design elements in the data structure. A weighting (Section 2.2.2.3) is applied to the value of the response

variable, Y_{ij} in Equation (2.21), for each household in the survey to indicate the number of individuals in the population which are represented by that particular household (Elbers et al. 2003). Household weights quantify the number of households in the population which are represented by a particular household in the survey. Individual person weights, rather than household weights, are more appropriate for poverty indicators. Weightings, defined as the inverse of selection probabilities, are required since the survey design features of stratification (Section 2.2.2.4) and clustering (Section 2.2.2.5) result in non-constant probabilities of inclusion.

The units of measurement in the Nepal analysis are households, which may not be independent since households tend to cluster into villages, or other small administrative or geographic aggregations, which are relatively homogeneous. Households close together tend to be more alike that households which are distant. This feature is incorporated into the ELL regression model by the inclusion of a random error component, γ_j in Equation (2.21), representing intra-cluster correlation. The variability at cluster and household level, γ_j and ϵ_{ij} respectively, have a direct effect on the standard error of average predictions of the target variable. Failure to include the term γ_j, representing spatial correlation in the unexplained variation in the model, would result in standard errors being underestimated. Although the small area estimates involve a non-linear transformation of the target variable, e.g. using the FGT equations (Section 1.1), it is sensible to assume that the cluster effects exert a similar influence on the standard errors of prediction across the small areas.

Treating the clusters, PSU's, as fixed effects implies that they are separate, independent units. By considering the clusters as representing random effects, we assume that the clusters arise from some distribution with mean of zero (McCulloch et al. 2008), and so are more similar to each other than as fixed effects. The consequence of constraining the cluster effects into a random term is to induce the cluster estimates to be more similar and to shrink in value towards their overall mean. Typically, only a small fraction of population clusters are sampled, about 1% in the Nepal survey, so treating the clusters as fixed effects does not provide a basis for prediction of the non-sampled clusters.

2.3.3 Variance estimation

Under the assumption that the linear mixed model in Equation (2.21) is correct, the expected value of the cluster and household components is zero. Thus, the model built from the survey data needs to be applied only once to obtain a point estimate of the required poverty measure, since the small area estimates average across all households. However, estimation of the standard errors of prediction needs to allow for the variability at each different level, cluster and household. Because of the complex structure in the survey data, some type of resampling for variance estimation is required to achieve this (Section 2.2.3).

The ELL methodology employs the Monte Carlo simulation technique of bootstrapping (Section 2.2.3.4) to estimate standard errors of prediction (Elbers et al. 2003).

The bootstrap procedure is applied to all three sources of variability in Equation (2.21), variability due to uncertainty in the regression model coefficients, as well as arising from clustering and household effects. The bootstrap estimates, Y_{ij}^b, have the form

$$Y_{ij}^b = \mathbf{x}_{ij}^T \boldsymbol{\beta}^b + \gamma_j^b + \epsilon_{ij}^b \,, \qquad b = 1, ..., B. \tag{2.22}$$

The usual application of bootstrapping is to resample from the original data to obtain new bootstrap estimates; either parametrically by sampling with replacement from the distributions of the predictor variables (Efron 1980), or non-parametrically by resampling the observations, the empirical distribution. The approach used in ELL is to retain the values of the predictor variables, the $\mathbf{x}$'s, and create new data using the model, by bootstrapping the estimated components of the model. Bootstrap values are obtained from the sampling distributions of the regression parameter estimates and empirical distributions of cluster and household effects. Each $\boldsymbol{\beta}^b$ is independently drawn from a multivariate normal distribution with mean $\hat{\boldsymbol{\beta}}$ and covariance matrix V_β. This approach is valid since $\hat{\boldsymbol{\beta}}$ is an unbiased estimate of the vector of regression coefficients $\boldsymbol{\beta}$. The Nepal analysis uses a non-parametric approach to bootstrapping the random effects of the model at cluster and household level. Cluster-level effects, the γ_j^b, are drawn randomly with replacement from $\hat{\gamma}_j$, the set of cluster-level residuals. These comprise residuals only from the clusters sampled, not for all clusters in the population. Each ϵ_{ij}^b is achieved by a random draw from the empirical distribution of $\hat{\epsilon}_{ij}$, either from the complete set of model household residuals or restricted to those residuals relating to the chosen cluster. Multiple bootstrap estimates can be generated to provide a standard error of prediction of poverty across a specific small level.

We conclude the section on the ELL methodology with a summary of the steps involved in generating small area estimates using this technique. The process to obtain a mean (point estimate) and standard error of prediction for the small area estimation of a poverty measure is illustrated with reference to the underlying variable of interest, Y, being log per capita expenditure. The general structure of the algorithm outlined in Section 2.3.4 is applied in the research on poverty estimation using tree-based models. The significant difference in the procedures for the linear mixed model and the tree-based model is at Step 4, generation of standard errors.

2.3.4 Summary of the ELL methodology

Since the predictors used to build the model in Equation (2.21) have been restricted to variables which are also observed in the census, the model constructed from survey variables can then be applied to the corresponding census variables to estimate the distribution of Y for each household in the population, conditional on the observed variables for that specific household. Household census predictions are aggregated across small domains to provide small area estimates, and standard errors of prediction are obtained from multiple small area estimates across a specific domain. This process is outlined as follows:

1. Construct a set of predictor variables common to both survey and census (Section 2.3.1)
2. Build a linear mixed model having the form of Equation (2.21). Apply weightings to the response variable, Y_{ij} (Section 2.3.2)
3. Apply the parameter estimates from the model build in Step 2 to the census data
4. Generate bootstrap predictions, Y_{ij}^b (Section 2.3.3), of log per capita expenditure for each household in the population, using Equation (2.22)
5. Exponentiate each of these household bootstrap predictions to obtain estimates in terms of per capita expenditure. Apply a non-linear transformation, as per Equation (1.1), to provide a prediction of the poverty measure at household level
6. Aggregate these bootstrap predictions at household level over a specific spatial domain to provide an estimate of the poverty measure at small area level, for example across each ilaka
7. Repeat Steps 4 to 6 to obtain 100 predictions of the specified poverty measure at small area level
8. The mean and standard deviation of these 100 predictions of poverty at the small domain level provide the small area estimate of the poverty measure with its associated standard error (Section 2.3.3)

The procedure involved in standard small area estimation methods begins with constructing a survey design, then conducting a survey and collecting data. A linear mixed model model is built from the survey data and auxiliary information is incorporated into the process to provide reasonable precision in the small area estimates. The ELL methodology differs from other small area estimation techniques in that the auxiliary information is provided by census data.

2.4 Tree based methods

2.4.1 Introduction

Research into the level of deprivation in Nepal (Haslett & Jones 2006) included estimation of three measures of poverty; poverty incidence, poverty severity and poverty gap. This thesis extends the ELL methodology by utilising tree based models, instead of the linear mixed model, for small area estimation of poverty. The scope of the research is adaptation of the classification and regression trees to incorporate the elements of survey design and to generate standard errors of estimates. The methodology of decision trees is utilised in the discipline of machine learning, and also in decision analysis for decision making (Goodmin & Wright 2014). In decision analysis a decision tree is a structure which displays sequences

of possible decisions, with probabilities for associated outcomes. The major goal is to determine the best sequence of decisions. Decision trees in the machine learning context were initially developed as a classification tool (Quinlan 1990). In this approach, decision trees are used to graphically display the results of applying an algorithm to input data, with the purpose of providing predictions or decisions. It is the machine learning approach to decision trees which is used in this research.

Classification and regression tree algorithms extend the machine learning technique of decision trees to a class of statistical models. The concept of tree-based statistical models was first proposed to handle variable interactions in the analysis of survey data (Morgan & Sonquist 1963). Their article focused on predictor variables and did not address the issue of adjusting for survey design. In the area of machine learning, complementary work was carried out and various algorithms for decision trees devised, one of the earliest being *ID3* by Quinlan (1986). Breiman et al. (1984) brought the topic of decision trees into the statistical arena and proposed new algorithms for tree construction, in particular the CART (Classification and Regression Trees) algorithm. A continuous response variable gives rise to a model called a "regression tree", whereas a "classification" tree is the result of modelling a categorical variable. The objective in the machine learning approach is to construct a decision tree which correctly classifies all units from the input dataset (Quinlan 1986). In contrast, statistical application of decision trees is based upon the probability of correct classification.

2.4.2 Building the tree

Construction of a tree model is by means of a top-down iterative process which recursively partitions the data space into different regions, smaller subsets, then fits a simple model in each region. A classification tree assigns to each region a class which is one level of the categorical response. The simple model fitted in each partition of a regression tree is the mean value of the continuous response variable for that region. The most commonly used tree-based model is a binary classification tree (Venables & Ripley 2002) which partitions the data into only two regions at each split. A key advantage of a binary tree is its interpretability (Hastie et al. 2001). A single tree can fully describe the partitioning of the dataspace. It is the binary tree model which is used in this study.

Each partitioning of the data space is determined by a "splitting rule", based upon the values of a single predictor variable. For a binary tree, observations satisfying a splitting condition at particular junction of the tree are sent down the left branch emanating from that junction. The remaining observations pass down the right branch. If the splitting attribute is a categorical variable A, having levels $L_1, L_2, \ldots, L_n$, the splitting test is of the form $L_i \in S_A$, where S_A is the splitting subset of categorical attribute A, comprising specific levels L_i of the variable A. A continuous splitting variable, X, divides the observations into left or right branches according to rules such as $X \leq x_i$ and $X > x_i$ respectively for some value x_i of the continuous variable X.

A point in the tree is referred to as a "node". The initial point of the tree is

called the "root node", and the tree endpoints are known as terminal nodes or leaves. In a classification tree, each node is assigned a label which represents one of the categories of the response variable. Classification is determined by the majority class in the node. The value assigned to each node in a regression tree is the average of the response values for the observational units which are directed into that node by the partitioning process. The terminal nodes, leaves, represent the final "decision" allocated to each observational unit; a class of the binary categorical response in a classification tree; the mean response for all units in the terminal node for a regression tree.

Each node of a tree can be considered to have its own distribution, dependent upon the type of tree built and the observational units which pass into a particular node. The node distribution is employed in determining the best splitting criterion for that node.

2.4.2.1 Distributional structure of tree nodes

Given a response variable Y and predictors $\boldsymbol{X}$, the tree algorithm can be interpreted as a step function, a piecewise constant function, approximation to a generic function $y = f(\boldsymbol{x})$ (Azzalini & Scarpa 2012). This approach is easily visualised when applied to a regression tree, but also is applicable to a classification tree. Each partitioning of the data space extends the tree to another branch level and produces a more accurate approximation of $y = f(\boldsymbol{x})$. If the step function is considered to relate to a parameter in the conditional distribution of $y|\boldsymbol{x}$, then the basis for choosing partitions is the likelihood function (Clark & Pregibon 1992). For each new split at an internal node, the likelihood ratio statistic, the *deviance*, is used to determine which partitioning of the node is most likely, i.e. will provide the best separation of observational units in the node, based on the given data. The deviance function is defined as being minus twice the log-likelihood of a particular distribution. The likelihood function approach to determining the best split at a node is discussed separately for classification and regression trees.

2.4.2.2 Determining the best split in a classification tree

A probability distribution of the form p_{ck} can be assigned to each node in a classification tree, where p_{ck} represents the probability of a unit being assigned to the c^{th} class in the k^{th} node. The probability of being categorised into a particular class is computed within a Bayesian framework, with default prior probabilities for each class being the class proportions of the dataset. The posterior probability of the c^{th} class in a specific node is the proportion of sample units in the node which belong to class c . For a given node, the Bayes decision rule assigns the class with largest posterior probability to each unit in that node (Venables & Ripley 2002). Let N_k represent the k^{th} node, and $p_{ck} = P(c|N_k)$ denote the probability of observing class c given the k^{th} node, N_k. Then, by the Bayes rule, the posterior probability of an observation from node N_k being in the c^{th} class, $P(c|N_k)$, is given by,

$$P(c|N_k) \;=\; \frac{\pi_c \, P(N_k|c)}{P(N_k)} \,, \tag{2.23}$$

where π_c is the prior probability for the c^{th} class, $P(N_k|c)$ is the probability that an observation in the k^{th} node is in class c , and $P(N_k)$ is the probability of an observation being in the k^{th} node. Equation (2.23) is approximately equivalent to (Therneau & Atkinson 2013)

$$\frac{\pi_c \dfrac{n_{ck}}{n_k}}{\sum_c \pi_c \dfrac{n_{ck}}{n_k}} , \tag{2.24}$$

where n_k is the number of observations in the k^{th} node, N_k, and n_{ck} the number of observations in the k^{th} node having class c. When the prior probabilities, the π_c, are the observed class frequencies, then the posterior probability of the c^{th} class reduces to,

$$p_{ck} \;=\; P(c|N_k) \;=\; \frac{n_{ck}}{n_k} \, . \tag{2.25}$$

Since each observational unit in the training set is finally allocated to a terminal node, or leaf, then each leaf comprises a random sample n_{ck} from the multinomial distribution specified by the probabilities p_{ck} (Venables & Ripley 2002). By conditioning on the observed predictors in the training set, the number of cases, n_k, in the k^{th} node are known. The conditional likelihood then is proportional to

$$\prod_{cases\, i} p_{[i]y_i} \quad = \quad \prod_{leaves\, k} \prod_{classes\, c} p_{ck}^{n_{ck}} \quad ,$$

where $[i]$ represents the leaf assigned to case i (Venables & Ripley 2002). The deviance for a classification tree, the sum of deviance over the leaves, is thus defined as,

$$D \;=\; \sum_k D_k, \quad \text{where} \quad D_k \;=\; -2 \sum_k n_{ck} \log(p_{ck}) \quad . \tag{2.26}$$

The deviance equates to zero if the node is pure, i.e. if all observations in the node have the same class, and deviance increases with greater diversity in the node. The deviance function can be used to choose the best splitting criterion for each node. Consider a "parent" node, S, which splits into two "daughter" nodes, one on the right hand side, R, and the other on the left hand side, L. Then, the best splitting variable is that which maximises the reduction in deviance between parent and daughter nodes, $D_S - D_R - D_L$. Ciampi, A., Chang, C.H., Hogg, S. and McKinney, S. (1987) and Clark & Pregibon (1992) propose this methodology for determining the best split for a classification tree, but the more common approach utilises a measure of impurity at the node. In this approach the choice of predictor for a given split is based upon the variable which minimises the impurity, or diversity, of the split. Two common measures of impurity are *entropy* and the *Gini index* (Venables & Ripley 2002).

These two measures of impurity are illustrated in the context of estimation of poverty incidence. Suppose that the k^{th} node of a classification tree modelling poverty incidence comprises a subset of the data S_k consisting of n_k observations. The term $\hat{p}_k$, denoting the estimated proportion of poor in the k^{th} node, is of the form (Hastie et al.

2001),

$$\hat{p}_k \;=\; \frac{1}{n_k} \sum_{\boldsymbol{x}_i \in S_k} I(y_i = 1)\,, \tag{2.27}$$

where y_i, for $i = (1, 2, \ldots, n_k)$, is the response variable for the i^{th} observation, and takes value of 1 if the household is classified as being poor and 0 when the household is categorised as not being poor. The term $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ denotes the values of the p predictors for the i^{th} observation. One measure of node diversity is *entropy*,

$$\text{entropy} \;=\; \sum \hat{p}_k \; log(\hat{p}_k)\,.$$

The most commonly used measure of impurity is the Gini index. For a binary categorical variable the Gini measure is defined as,

$$\text{Gini index} \;=\; 2\hat{p}_k(1 - \hat{p}_k)\,.$$

When a parent node splits into two child nodes, the impurity of the split is determined as a weighted average of the impurities of each child node. The formulation of the Gini index to represent node impurity is illustrated in the context of poverty incidence. Consider a parent node, S of size $n_{_S}$ and $p_{_S}$ the proportion of poor households in the node, a right child node R of size $n_{_R}$, and a left child nodes L of size $n_{_L}$, with proportions of poor being denoted by $p_{_R}$ and $p_{_L}$ respectively.

Using the Gini index, the measure of impurity for the split is,

$$\text{Gini}_{\,\text{split}} \;=\; 2\,p_{_S}\,(1 - p_{_S}) \;-\; 2\left[\frac{n_{_R}}{n_{_S}}\,p_{_R}(1 - p_{_R}) \;+\; \frac{n_{_L}}{n_{_S}}\,p_{_L}(1 - p_{_L})\right]\,. \tag{2.28}$$

The variable which minimises the Gini index for the split is chosen as the splitting variable. The selection of splitting variable in the algorithm used by the function *rpart* (Therneau et al. 2013), used to build the trees, is based upon maximising the statistic $improve = n_{_S} \times IG$, where IG, the information gain, is equivalent to Equation (2.28). Defining a deviance function in a regression tree is a much simpler procedure, since the formulation of deviance depends upon the residual sum of squares.

2.4.2.3 Determining the best split in a regression tree

The expression for deviance for a regression tree is defined as the aggregation of the sum of squares for each leaf in the tree,

$$D \;=\; \sum_i D_k, \quad \text{where} \quad D_k \;=\; \sum_{i=1}^{n_k} (y_{ik} - \mu_k)^2\,, \tag{2.29}$$

for y_{ik} the response value for the i^{th} observation in the k^{th} leaf, and μ_{ik} the mean response for the k^{th} leaf. The mean parameter, μ_k is constant for all observations in the k^{th} leaf. The average value of response values of all observations in the leaf is the maximum-likelihood

estimate for μ_k, and also the minimum-deviance estimate. A leaf is pure when all response values, all the y_{ik}, are identical.

Hence, the probability model adopted for a regression tree (Clark & Pregibon 1992) is to assign a normal distribution at the k^{th} leaf having the form $N\left(\mu_k, \sigma_k^2\right)$, where σ_k^2 denotes the variance of response values for observations in the node. Then, the form of Equation (2.29) is the usual scaled deviance for a Gaussian generalised linear model, i.e. minus twice the log-likelihood multiplied by the scale parameter for the normal distribution, σ^2, which is assumed to be constant for all values of i (Clark & Pregibon 1992). Venables & Ripley (2002) assert that the deviance formulation should only be applied at the leaves of the tree, since internal nodes comprise a mixture of normal distributions. However, the usual method for choosing a splitting variable at an internal node is based upon the predictor which maximises $SS_{parent} - (SS_{right} + SS_{left})$, the change in deviance between the parent node and daughter nodes (Clark & Pregibon 1992). Consider the deviance at the root node, which is given by

$$D \;=\; \sum_{i=1}^{n} (y_i - \bar{y})^2 \;. \tag{2.30}$$

If the root node then splits into a right child node, R, of size $n_{_R}$ and a left child node, L, of size $n_{_L}$, then the residual sum of squares at the root node can be expressed in terms of sums of squares for the two child nodes

$$\sum_{j=1}^{n} (y_j - \bar{y})^2 \;=\; \sum_{j_R=1}^{n_R} \left(y_{j_R} - \bar{y}_R\right)^2 + \sum_{j_L=1}^{n_L} \left(y_{j_L} - \bar{y}_L\right)^2 + n_{_R} \left(\bar{y}_R - \bar{y}\right) + n_{_L} \left(\bar{y}_L - \bar{y}\right) \;. \tag{2.31}$$

The first two terms in Equation (2.31) denote the "within-group" sum of squares and the last two terms represent the "between-group" sum of squares. The splitting variable chosen is that which produces the smallest possible sum of the first two terms and largest possible sum of the final two terms (Maindonald & Braun 2010), which is equivalent to the approach used in a one-way analysis of variance of maximising the between-groups sums of squares. The algorithm used in *rpart* reflects the approach by Clark & Pregibon (1992), and bases the splitting criteria for a regression tree on the difference between the sum of squares for the parent node and combined sum of squares for the two daughter nodes (Therneau & Atkinson 2000). This procedure is equivalent to change in deviance as defined by Equation (2.29). The summary output for a regression tree model using *rpart* includes, for each internal node, a term denoted as *improve* which is the percentage change in sums of squares from parent to daughter nodes, $1 - (SS_{right} + SS_{left}) / SS_{parent}$.

Having built a classification or regression tree, the optimal tree size becomes an issue. The tree size is managed in *rpart* through various control parameters, the complexity parameter, tree depth and minimum split specification.

2.4.2.4 Pruning the tree

Optimising the tree to a suitable size so as to prevent overfitting is achieved through "pruning" the tree. Pruning is applied to the tree to remove leaves and branches in order to improve the tree's performance. Each pruned tree is a subset of the full tree, leading to a sequence of nested models (Breiman et al. 1984). Trees which are too large tend to overfit the data. Undersized trees, on the other hand, may omit important features of the tree structure. One process of pruning involves minimising the cost-complexity measure $R_\alpha(T)$ for a given tree T (Breiman et al. 1984),

$$R_\alpha(T) \;=\; R(T) \;+\; \alpha|T|\,. \tag{2.32}$$

$R(T)$ denotes model error, as defined by the misclassification rate, using the Gini or entropy measures. The complexity of the tree is represented by $|T|$, the tree size, i.e. the number of leaves, and α denotes the cost-complexity parameter. The largest possible tree is very complex but has no error. Smaller trees are less complex but contain error. Thus, this method of pruning the tree involves balancing the complexity or size of the tree with node impurity.

Two alternative means of controlling tree size are by specifying the minimum number of observations in a node before splitting is allowed, or by fixing tree depth, i.e. the number of levels to which the tree can extend, with the root node being at Level 0. Once an optimal tree has been decided upon, the model can be assessed for suitability.

2.4.3 Assessing model fit

In the context of the discipline of Machine Learning, the dataset used to build a decision tree is usually divided into three parts, training, validation and test sets (Berry & Linoff 2004). The training set is used to build the model and the validation data is applied to generalise the model so it is less dependent upon the training data. The test set examines model fit, by assessing the likely effectiveness of the model with new data.

In a classification tree, model fit can be quantified by misclassification rate, the probability of misspecifying a new sample of data using the given model. Misclassification is the proportion of observations in a node which are incorrectly classified, i.e. their true class does not match the class assigned by the model. The most common method for estimating misclassification is by cross-validation (Breiman et al. 1984), the "leave-one-out" procedure. The data is divided into several parts. One of these parts is used as a test set and the rest of the data comprises the training set from which a tree is built. The model built from the training data is then applied to the test set, which contains observations with known class labels of the response variable. A count is taken of the numbers of observations which are correctly and incorrectly predicted, and these counts are tabulated in a *confusion matrix*. From this table the proportion of misclassified can be determined.

Finding the best classification or regression tree model is essentially achieved in the pruning process, by choosing the value of the cost-complexity parameter, α, which minimises the the cost-complexity measure $R_\alpha(T)$ (Equation (2.32)). Optimising α is carried out using the cross-validation method. In the *rpart* algorithm, the division into training and test sets is generally in a ratio of 90%:10% for training:test. This is known as "ten-fold" cross validation, and provides ten estimates of misclassification. Using this method results in some degree of bias since it essentially employs resubstitution of the training data, the same data used to build the model is also used to test its effectiveness. However, cross-validation is parsimonious in its use of the data (Breiman et al. 1984), and is the preferred method especially with small datasets.

Chapter 3

Classification tree models for poverty estimation

3.1 Introduction

The ELL technique (Elbers et al. 2003) was devised specifically for the estimation of poverty, based on per capita expenditure, at small domain level. The purpose of the research is to investigate tree-based models as an alternate approach to the regression approach to modelling poverty used in the ELL methodology. Classification tree models (Breiman et al. 1984) provide a simple and direct method of estimating poverty incidence. The methodology is independent of parametric assumptions and incorporates non-linear predictors as well as interactions of explanatory variables as a matter of course. The standard application of tree-based models, however, assume that the data are identically and independently distributed, so that adjustments to the technique are required if it is to be applied to complex survey data. Modification of the tree-based method includes incorporation of the survey design elements of stratification, clustering and weighting into the tree model, and a suitable method to ensure valid estimates of variability. The problems associated with estimation at small area level are addressed by utilising auxiliary information in the modelling process.

This chapter explores classification trees as an alternate methodology to ELL for small area estimation of poverty incidence in Nepal. Construction of a set of auxiliary variables for the modelling is discussed. Using poverty incidence as response variable, different classification tree models were built and compared with the ELL linear regression mixed model. Initially, an unweighted tree model was built, then a weighted tree, to examine possible changes in the configuration of splitting criteria when survey weights were incorporated into the model. The effects on tree structure by applying different methods of tree pruning were also investigated. A weighted tree model of suitable size was selected and applied to the auxiliary variables drawn from the census dataset to provide small area predictions of poverty incidence for eighteen ilakas in a specific district of Nepal. “Hard” and “soft” point estimates of poverty incidence for the eighteen ilakas

were defined and generated, and compared with the corresponding point estimates from the ELL analysis of poverty incidence in Nepal (Haslett & Jones 2006).

3.2 Building the classification tree model

The structure of the ELL methodology to generate small area estimates of poverty was outlined in Section 2.3.4. A similar algorithm was applied when estimating poverty rates using classification tree modelling. The first step was to construct a set of auxiliary variables from the survey data to provide predictors for the model.

3.2.1 Obtaining a suitable dataset for modelling

The dataset used to build the ELL regression model comprised data from a Nepal Living Standards Survey (Central Bureau of Statistics, Nepal 2004a,b), a population census (Central Bureau of Statistics, Nepal 2002) and geographical information. The census dataset provided census means, variables which represented averages across all households in the census at VDC or ward level. For regression modelling using the ELL technique, factors were converted to dummy variables and additional predictors were constructed from the two-way interaction of continuous auxiliary variables.

However, classification tree modelling requires categorical information in the form of factor, not dummy, variables. In addition, the tree structure handles interactions between variables automatically through the process of consecutive splits. Thus, the original factor variables, components of the set of common predictors extracted from the Nepal Living Standards Survey dataset, were used in the tree model rather than dummy variables constructed from the categorical predictors to represent different levels of each factor. Consequently, the survey dataset for classification tree modelling comprised the set of auxiliary variables used for the ELL model, but omitting all two-way interaction predictors and dummy variables. Classification tree modelling was then applied to the dataset.

3.2.2 Construction of an unweighted classification tree

The classification tree technique was applied to the amended Nepal survey dataset described in Section 3.2.1. The response variable for the classification tree model for poverty incidence was the binary variable “Poverty” having levels “poor” and “notpoor”. Poverty incidence is indicated by a household having a per capita expenditure below a set poverty line of 7695.744 rupees, representing per capita expenditure per year in average 2003 Nepalese rupees (Haslett & Jones 2006). The set of auxiliary variables which provided the candidate predictors of poverty comprised one hundred and three variables containing household level information, census means and Geographical Information System (GIS) data. The dataset was at household level, consisted of 3912 observations and included a variable denoting household size as the number of people in the household.

An unweighted classification tree model (Section 2.4.2) for poverty rates in Nepal was built using recursive partitioning of the data into subgroups labelled "poor" or "notpoor" based upon specific conditions, commonly known as the splitting rules. The classification of each node as "poor" or "notpoor" was determined by the major class at the node. The modelling was carried out using the *rpart* function from the R software (R Core Team 2015), using the default setting for most parameters, including the control parameter $minsplit = 20$ (which specifies a minimum of twenty observations in a node before a split can be attempted), and the Gini index splitting criterion as the measure of node impurity (Hastie et al. 2001),

$$\text{Gini index} \;=\; 2\hat{p}_i(1-\hat{p}_i)\,, \tag{3.1}$$

where $\hat{p}_i$ is the proportion of poor in the i^{th} node of the tree. The default value in the *rpart* function for the threshold complexity parameter used to determine tree size is $cp = 0.01$. This value produced a tree for poverty incidence which was too sparse, having only six splits, so a cp of 0.005 was used to build an unweighted classification tree model of poverty incidence for Nepal. The value of $cp = 0.005$ was chosen because it produced a tree of manageable size, so that the associated tree diagram had no overlapping text at the deepest level of the tree. The focus of the analysis at this stage was to explore various modelling options, rather than select the "best" tree. The tree diagram for the unweighted model with $cp = 0.005$, displayed in Figure 3.1 was constructed using the *rpart.plot* package (Milborrow 2015) from the R software. For interpretation of code names for splitting variables used in the tree, as displayed in Figure 3.1, refer to Appendix A, which indicates the variable type for each predictor, and provides a description of the variable, including listing all categories of factor variables.

The tree diagram in Figure 3.1 displays important features of the unweighted model. Nodes are represented by boxes, with a rectangular shape indicating an internal node, and an oval shaped box for each leaf. Branches are labelled with their associated splitting criteria. Colour coding identifies the class assigned to each node by majority vote: green indicates a node classed as not poor, while yellow represents a node labelled as being poor. The response variable, Poverty, can be considered to have a Binomial probability distribution at the i^{th} node of the form $Y_i \sim B\,(n_i, p_i)$ (Section 2.4.2.2). The parameter values of the probability distribution at each node, the probability of being poor, p_i, and the number of observations in the node, n_i, are displayed inside the node box. For example, the root node of the unweighted classification tree in Figure 3.1 shows that the full survey dataset of 3912 households has 23% in poverty. The left hand split from the root node produces a partition of size 2040 households of which only 10% are poor. Nodes with $p_i > 0.5$, classed as poor, are coloured yellow, while green shading indicates nodes designated as not poor, i.e. having $p_i \leq 0.5$.

Using $cp = 0.005$ has produced a tree of size twenty one, with twenty splits producing twenty one terminal nodes, or leaves. The tree model is constructed so that, for each binary split, the partition with the greatest proportion of poor households is

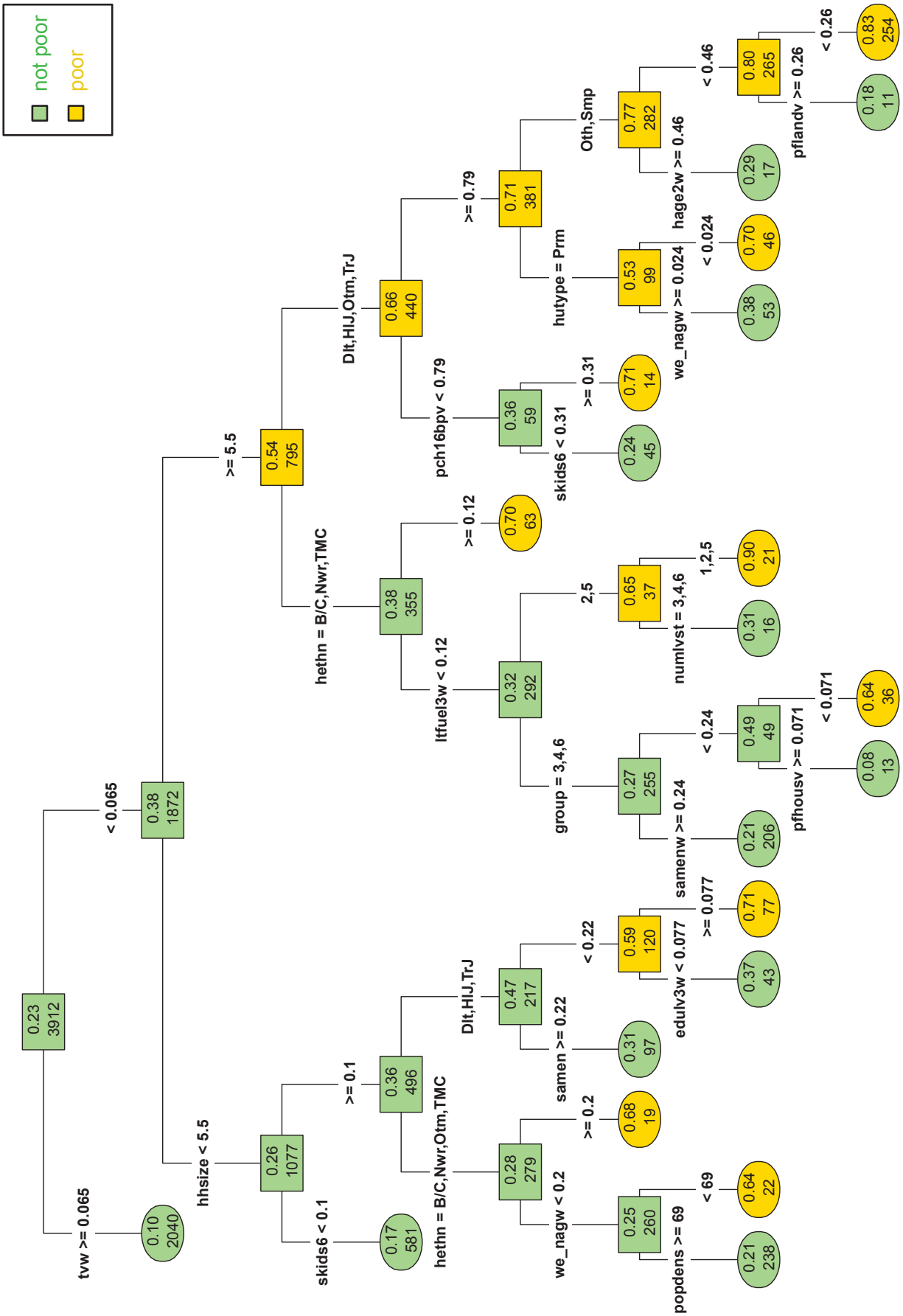


Figure 3.1: Unweighted classification tree model for poverty incidence in Nepal with $cp = 0.005$

sent to the right hand branch (Therneau & Atkinson 2000). The splitting criterion is thus formulated so that households which meet the splitting criterion are sent along the left branch. For example, at the first split in the tree in Figure 3.1, the root node, all households in a ward in which 6.5% or more of the households own a television are classified as being not poor.

We note that the first left hand split from the root node, designated as Node 2, produces a leaf. No further splitting occurs beyond this node. To produce a tree which continued to split at Node 2 required a *cp* value of less than 0.0035359. But this resulted in a tree with fifty seven splits, which was considered to be too large, in the context of balancing a reduction in tree complexity with minimisation of misclassification rate. However, no actual model tuning was carried out, for example by fitting an out-of-sample validation set to determine the optimal model. Having constructed an unweighted tree of suitable size, the next step in the exploration process is to include survey weights into the classification tree.

Recall that the sampling design for the Nepal survey incorporated a two-stage stratified random sampling approach (Section 2.2.2.6). The Primary Sampling Units (psu's) were clusters, selected at the first stage of sampling from the six strata using probability proportional to size sampling, with size representing the number of clusters in the stratum. A systematic sample of twelve households was then taken within each sampled cluster, resulting in identical weights for households in the same cluster for a specific stratum. These household weights were then adjusted to allow for missing values and post-stratification. The final weights used in the modelling process, represented by the variable *indwght*, were at individual rather then household level, and indicate that the number of individuals in the population represented by the sampled household. They were constructed by multiplying the survey household weight by household size. Model weights at individual level are required since poverty incidence measures the "proportion of poor people" rather than the "proportion of poor households".

3.2.3 Incorporating survey weights

The survey design element of weighting was included to ensure that the model built from the survey data is fully representative of the population (Section 2.2.2.3), i.e. to avoid introducing bias into the estimates (Toth & Eltinge 2011). The *rpart* function (Therneau et al. 2013), the statistical software used to build the tree, has a weight argument that incorporates survey weights into the Gini splitting criteria, and into node summaries. The estimate of the proportion of poor at the k^{th} node, comprising the subset S_k of observations, Equation (2.27), is amended to,

$$\hat{p}_k \;=\; \frac{1}{\sum\limits_{x_i \in S_k} w_i} \quad \sum_{x_i \in S_k} w_i \times I(y_i = 1)\,. \tag{3.2}$$

The weight, w_i, used for each observation is defined as,

$$w_i \;=\; \frac{N \;\times\; ind_i}{\sum_{i=1}^{N} ind_i}\,,$$

for N the total sample size of 3912. The term ind_i denotes the number of individuals in the population represented by the i^{th} observation, i.e. the i^{th} household, and comprises the values of the weighting variable used in the ELL modelling, *indwght*. It is created by multiplying the survey weights for each household by the household size, the number of people in the household. Individual-level weights are re-scaled so that they sum to the sample size, in order to maintain the same effective sample size . An accurate determination of the precision of the estimates needs to allow for the dependence structure within each cluster. The issue of standard errors of prediction using the classification tree model is addressed in Chapter 4.

A weighted classification tree model for poverty incidence in Nepal was built using the same criterion as applied to the unweighted model to determine tree size, the complexity parameter *cp*. The weighted tree now needs to be pruned to an optimal size.

3.2.4 Optimising the tree

In the machine learning approach to tree modelling, the data is divided into three distinct subsets, for training, validation and testing (Hastie et al. 2001). The training set is used to build the tree, the validation set is applied to choose the tree of optimal size, and the test set provides an independent measure of model error. The training data establishes tree structure, as determined by the splitting rules. Using the validation data the prediction error for different cp values, corresponding to different sized trees, can be obtained. The tree of optimal size is that with minimum prediction error. An alternative procedure to utilising a validation set involves efficient re-use of the training data by means of cross-validation (Breiman et al. 1984). Applying this concept provides a measure of variability in the model optimisation process but involves re-substitution of the data. A validation set supplies an independent estimate of optimal tree size but no measure of the precision of that estimate. The optimal tree size is usually determined using the cost-complexity criterion (Breiman et al. 1984),

$$R_\alpha(T) \;=\; R(T) \;+\; \alpha|T|\,, \tag{3.3}$$

where $|T|$ denotes the size of tree T, the number of leaves. The term α, the complexity parameter, measures the "cost" of adding another variable to the tree structure (Therneau & Atkinson 2013), resulting in another split in the tree. The "risk" of the tree T, denoted by $R(T)$, aggregates the risk for all terminal nodes, N_i, and is given by,

$$R(T) \;=\; \sum_i P(N_i)\, R(N_i)\,,$$

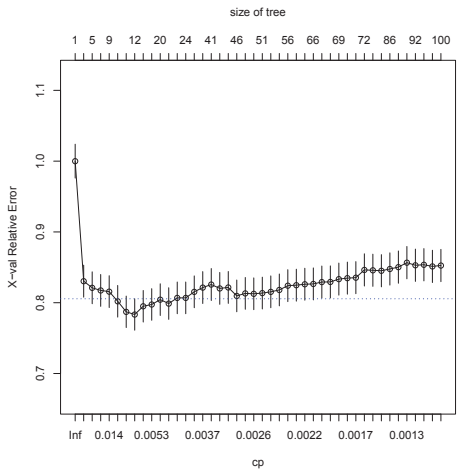
for $P(N_i)$ the proportion of observations in node N_i. When prior probabilities are set to the sample class frequencies, $R(T)$ is the proportion misclassified in tree T (Therneau & Atkinson 2013).

The process of optimisation begins by constructing a maximal tree, T_{max}, in which all the leaves are pure, and all the observations are in the same class. Pruning occurs by recursively snipping off the least important splits from the maximal tree, T_{max}, to produce a finite sequence of rooted subtrees (Venables & Ripley 2002), from T_{max} to T_0, the tree with just the root node. The cost-complexity criterion (Equation (3.3)) is applied to find the subtree T_α which minimises $R_\alpha(T)$ for a given value of α (Clark & Pregibon 1992). There can exist optimal subtrees of different sizes for a specific value of α (Venables & Ripley 2002). Breiman et al. (1984) showed that, for each value of α, there exists a unique smallest optimally pruned subtree.

While α can take any value along a continuum, the sequence of rooted subtrees produced by the pruning process is finite. Due to this finiteness, if T_α is the optimal tree for α, then it continues to be optimal as α increases until a "jump point", $\alpha^{'}$ is reached (Breiman et al. 1984), at which point a new tree, $T_{\alpha'}$, becomes optimal, and continues to be optimal until the next jump point. A *cp* plot can be produced from a classification tree model built by the *rpart* function, to illustrate cross-validation model error for the optimal tree corresponding to different values of a parameter, *cp*, which is defined as α divided by the error of the root tree (Venables & Ripley 2002). For the poverty incidence model based on a classification tree, the cost-complexity parameter was initially held at a value of $cp = 0.001$, and ten-fold cross-validation used to determine the best value of *cp* (Therneau & Atkinson 2000).

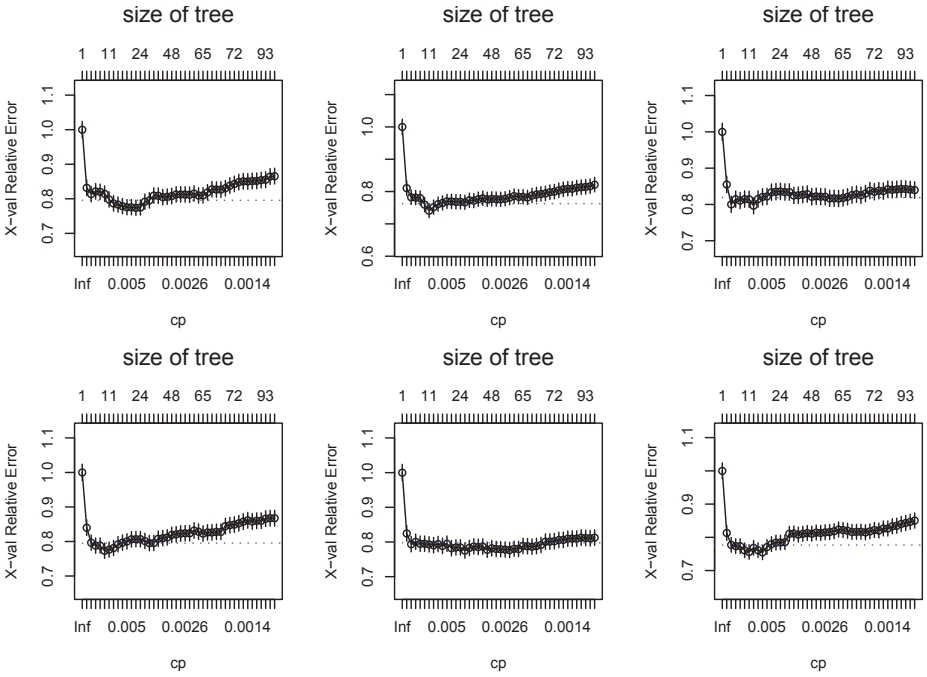
Figure 3.2 shows a typical result for the classification model of poverty incidence using ten-fold cross-validation. Since the choice of the ten cross-validation subsets is random, each different execution of the model results in a different group of ten subsets and consequently a different *cp* plot. The *cp* plots for six random ten fold cross-validation partitions of the survey training dataset are displayed in Figure 3.3.

Figure 3.2: Cp plot for the weighted classification tree



The output associated with the single *cp* plot, Figure 3.2, is displayed in Figure 3.4, which provides information on the predictive accuracy of the tree. Above the table in Figure 3.4, the root node error is given as 1206.8/3912. This non-integer value is due to the weightings applied to the response variable (Section 3.2.3), so that the sample size remains equal to the number of households in the survey, 3912, and the weight for an individual survey household equates to an averaging of all the population households which it represents. The table in Figure 3.4 lists a sequence of nested optimal trees (Breiman et al. 1984) constructed relative to decreasing values of the complexity parameter, *cp*, from the smallest tree with no splits to the largest, corresponding to $cp = 0.001$, and the corresponding measures of model error.

Figure 3.3: Cp plots for the weighted classification tree



Error values for the different trees are relative quantities (Venables & Ripley 2002), obtained by dividing absolute error by the error at the root node, i.e. the misclassification rate of 1206.8. Therneau & Atkinson (2000), the authors of *rpart*, use this scaling for the displayed values to ensure an error of one for the first node, the root node. Error computations are carried out on the absolute scale but printed on the relative scale, for ease of model comparison. The column *rel error* lists relative error for the full tree with specified number of splits. Relative error, also known as *resubstitution error rate*, measures the model error for predictions upon the same data from which the tree was built (Maindonald & Braun 2010). This quantity is not useful in determining tree size since it can never increase in value while the size increases, and so is likely to underestimate model error for a new sample. While cross-validation is also a resubstitution method, it provides a more useful measure of model performance. The columns *xerror* and *xstd* in the table from Figure 3.4 display the mean and standard deviation, respectively, for

Figure 3.4: Output of cp plot for weighted classification tree model of poverty in Nepal, with $cp = 0.001$

```
Classification tree:

Root node error: 1206.8/3912 = 0.30848

n= 3912

        CP nsplit rel error xerror   xstd
1  0.06043      0     1.000  1.000 0.0239
2  0.03648      3     0.789  0.833 0.0226
3  0.01495      4     0.752  0.812 0.0225
4  0.01398      7     0.707  0.829 0.0226
5  0.01329      8     0.693  0.831 0.0226
6  0.01058      9     0.680  0.823 0.0226
7  0.00831     10     0.670  0.817 0.0225
8  0.00566     11     0.661  0.826 0.0226
9  0.00540     15     0.637  0.826 0.0226
10 0.00519     18     0.621  0.824 0.0226
11 0.00479     19     0.616  0.819 0.0225
12 0.00451     21     0.606  0.820 0.0225
13 0.00447     22     0.602  0.815 0.0225
14 0.00447     23     0.597  0.815 0.0225
15 0.00395     32     0.547  0.811 0.0225
16 0.00349     36     0.530  0.828 0.0226
17 0.00324     40     0.515  0.832 0.0226
18 0.00305     42     0.509  0.835 0.0227
19 0.00299     44     0.502  0.823 0.0226
20 0.00277     45     0.499  0.824 0.0226
21 0.00265     47     0.494  0.830 0.0226
22 0.00263     49     0.489  0.828 0.0226
23 0.00260     50     0.486  0.828 0.0226
24 0.00257     52     0.481  0.828 0.0226
25 0.00250     53     0.478  0.831 0.0226
26 0.00238     55     0.473  0.829 0.0226
27 0.00225     57     0.468  0.835 0.0227
28 0.00219     64     0.449  0.839 0.0227
29 0.00195     65     0.446  0.849 0.0228
30 0.00190     66     0.444  0.851 0.0228
31 0.00185     67     0.443  0.848 0.0228
32 0.00184     68     0.441  0.850 0.0228
33 0.00173     69     0.439  0.850 0.0228
34 0.00163     70     0.437  0.852 0.0228
35 0.00155     71     0.436  0.853 0.0228
36 0.00152     78     0.425  0.858 0.0229
37 0.00141     84     0.412  0.858 0.0229
38 0.00136     85     0.411  0.864 0.0229
39 0.00132     86     0.410  0.864 0.0229
40 0.00122     89     0.406  0.864 0.0229
41 0.00119     91     0.403  0.861 0.0229
42 0.00113     92     0.402  0.866 0.0229
43 0.00104     98     0.394  0.874 0.0230
44 0.00100     99     0.393  0.872 0.0230
```

ten cross-validation estimates of relative model error. Figure 3.2 graphs the mean relative cross-validation error versus *cp* value, with corresponding cross-validation standard deviation displayed as error bars.

A range of *cp* values give rise to the same sized tree (Breiman et al. 1984). In Figure 3.4 , the column labelled "CP" indicates the lower limit of each range of cp values associated with a particular tree size. The complexity parameter, α can then be chosen to minimise the cross-validation error, *xerror*. Alternatively, a cp value may be chosen which has *xerror* within one standard deviation of the minimum cross-validation error on the graph, i.e. corresponding the smallest specified *cp* value of 0.001.

The choice of α is somewhat ad-hoc; any value within one standard deviation of the minimum is considered equivalent to the minimum (Therneau & Atkinson 2000). Figure 3.2 indicates a plateau of cross-validation error values within one standard deviation of the minimum *cp* value of 0.001, suggesting a large range of choice for a suitably sized tree. The standard practice for tree modelling is to use the smallest cp value which lies within one standard deviation of the minimum (Venables & Ripley 2002), but for the classification tree model for poverty incidence this would result in a tree with about eight splits, and therefore only eight predictors in the model. A compromise is made between the tree being too sparse and being unwieldy; a larger tree provides a better comparison with the ELL model. The model chosen is that with the same complexity parameter value as the unweighted classification tree model, having $cp = 0.005$, representing the tree model pruned to 19 splits, corresponding to 20 leaves. The corresponding tree diagram is displayed in Figure 3.5.

Comparing the tree diagrams of the unweighted and weighted tree models, Figure 3.1 with Figure 3.5, we note that they are of a similar size. The weighted tree with the same complexity parameter as the unweighted tree, $cp = 0.005$, has one fewer leaves than the unweighted tree. Despite having similar size, the unweighted and weighted trees have markedly different structures, although the splitting variable at the root node is *tvw* for both models. However, introducing the weights into the model has resulted in further partitioning of the first node on the left hand side of the root, Node 2. This produces a more balanced tree than the unweighted model (Figure 3.1), in which the whole left hand side of the tree is missing, because Node 2 is terminal. It also partly addresses the problem of 202 poor households misclassified as not being poor at Node 2 of the unweighted tree. The initial splitter on the right hand side of both trees is *hhsize*, but there are notable differences between the unweighted and weighted models with respect to the tree structure below this node. The discrepancies in structure between the two trees emphasise the importance of including survey weightings in the model when the data has a complex structure.

Applying correct weighting to survey observations is also necessary to ensure the survey is representative of the population, as can be seen by comparing the proportion of poor at the root nodes of the unweighted and weighted trees. In the unweighted tree (Figure 3.1), the box representing the root node lists the probability of being poor, the

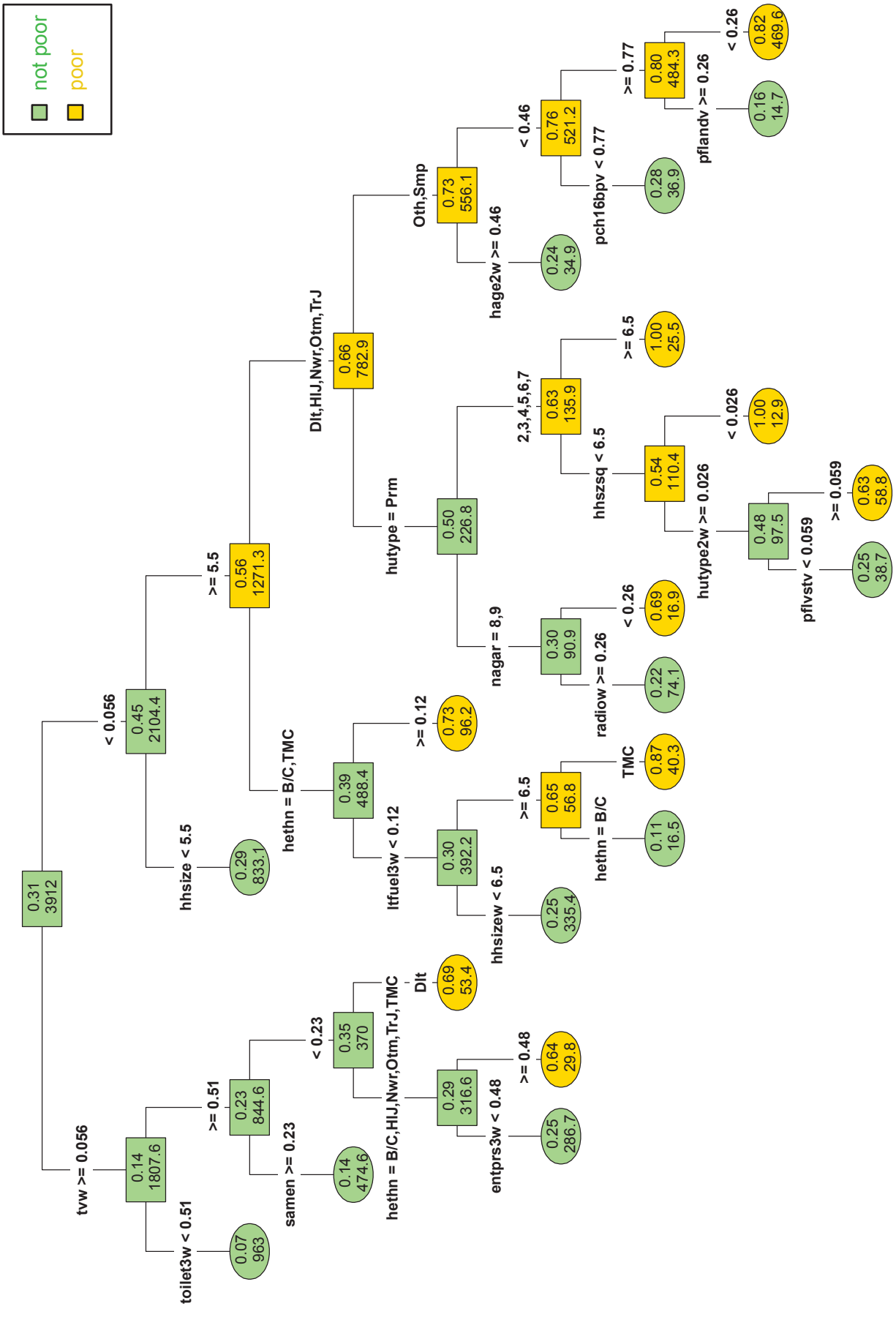


Figure 3.5: Weighted classification tree model for poverty incidence in Nepal with $cp = 0.005$

root node error, as 0.23 indicating that 23% of households in the full dataset have status of being poor. The root node error for the weighted tree, given in Figure 3.4 as 0.308, and rounded to 0.31 in the root node box, is the estimate of the proportion of individuals identified as poor. Poverty incidence for individuals is about 25% higher than the poverty status of households. This result suggests that larger households are more likely to be poor. We now examine the structure of the weighted tree model in a little more detail, to glean information about the determinants of poverty in Nepal.

3.2.5 Interpretation of the classification tree model

Another useful feature of the classification tree model is an evaluation of the usefulness of individual variables to the modelling process, which can provide another point of comparison between the ELL and tree methods. When modelling poverty incidence in Nepal, the focus is not solely upon generating predictions, but the structure of the model itself is of interest, including whether the model makes sense. A model which does "make sense" could increase the confidence that users of the model place upon the estimates. In this regard, the influence on the model of different variables is a helpful tool. In Figure 3.5, the five most important splitting variables in the pruned, weighted tree model are shown to be *tvw*, *toilet3w*, *hhsize*, *samen* and *hethn*. The first splitting variable, *tvw*, is a census mean which specifies the proportion of households in a given ward owning a television. This variable is chosen by the Gini index criterion as providing the smallest misclassification error at the initial split involving the root node. The households sent down the left hand branch of this split are those sited in a ward having television ownership of at least 5.6%. This subset of the data is further divided according to the percentage of households in the ward, above and below 51%, which do not have a proper toilet, as specified by the splitting variable *toilet3w*. The right hand side of the tree comprises households from wards with television ownership of less than 5.6%. This subset is then split according to household size. The tree continues to split until a stopping rule applies; based either on the minimum number in a node or the reduction in node impurity. The classification of each node is determined by the majority class in that node.

The terminal nodes, or leaves, represent the ultimate division of the data into distinct subsets, with all members categorised as either being poor or not poor. Two other important splitting variables are *samen*, which represents the proportion of adult men in the household, and *hethn*, denoting the ethnicity of the head of the household. The tree diagram in Figure 3.5 clearly illustrates an advantage of the tree technique over regression methodology, that the most important factors affecting poverty rates can readily be identified. The households comprising a particular leaf are represented by a pathway consisting of a unique set of splitting rules, which describe the characteristics of that collection of households. For example, a household is classified as being not poor if at least 5.6% of households in the ward own a television and more than 49% of households in the ward have a proper toilet. In addition, for those wards with television ownership rates of less than 5.6%, a household with 5 or fewer people is also classified as not being

poor, evidence that poor families tend to be larger. Even if the rate of tv ownership in the ward is at least 5.6%, a household is designated as poor if the proportion of households in the ward with no proper toilet is more than 51%, the proportion of adult men in the household is less than 23% and the ethnicity of the household head is Terai Middle Caste.

On first examination, television ownership might seem a surprising and trivial depicter of poverty. The weighted tree model is rebuilt with data excluding *tvw* as a predictor; the structure of this tree is displayed in Figure 3.6. To produce a tree of similar size, to aid comparison, the complexity parameter for the weighted tree excluding *tvw* is set as $cp = 0.006$. When *tvw* is removed from the set of predictors, it is replaced in its position of first splitting variable by *ltfuel2w*, a predictor representing the proportion of households in a ward which had kerosene as lighting fuel . In rural areas without electricity, kerosene is used to power generators which provide household electricity. Thus, a rural household can be considered to be "not poor" if it can afford the kerosene to run a generator, for lighting and the operation of a television set. Ownership of a television had no significant effect in urban areas in the ELL model of poverty status in Nepal.

The set of predictors for the regression model of poverty incidence included an interaction between tv ownership and whether the household was in a rural or urban area (Haslett & Jones 2006). In a classification tree model for poverty incidence in Nepal, tv ownership and urban/rural effect would enter as separate predictors, since the recursive feature of the tree algorithm automatically creates interaction between predictors. However, the urban/rural effect variable has not appeared in the tree model, suggesting that the urban/rural effect has been captured by other predictors, such as location variables.

The explanation for *ltfuel2w* acting as first splitting variable when *tvw* is removed from the set of predictors is related to the feature of competing splits in tree-based models (Breiman et al. 1984). Figure 3.7 displays the splitting criterion selected for the root node of the classification tree for poverty incidence in Nepal with the first five competing splits. Competing splits provide the next best splitting variables for the node, in terms of information gain, improving the misclassification rate, or minimising impurity. The first competing split for *tvw* at the root node is *ltfuel2w*, and so the first choice of splitter at the root node when *tvw* is not a choice of splitting variable. We note that the second competing split is *toilet3w* which appears in the weighted tree (Figure 3.5) as the first left hand split.

One advantage of the classification tree for poverty incidence over the regression model is clearly illustrated: the tree structure displays the variables in order of importance. Initially, the most important variables identify individuals who are not poor: for example, people in households with a tv, in smaller households and with a larger proportion of adult men in the household. The greatest difficulty of the modelling occurs in identifying people at the margins of poverty, those in households with per capita expenditure close to the poverty line. Having first established which conditions indicate financial advantage, the determinants of poverty can be more easily identified. Factors which contribute most to poverty appear as splitting variables in the upper levels of the unweighted and weighted

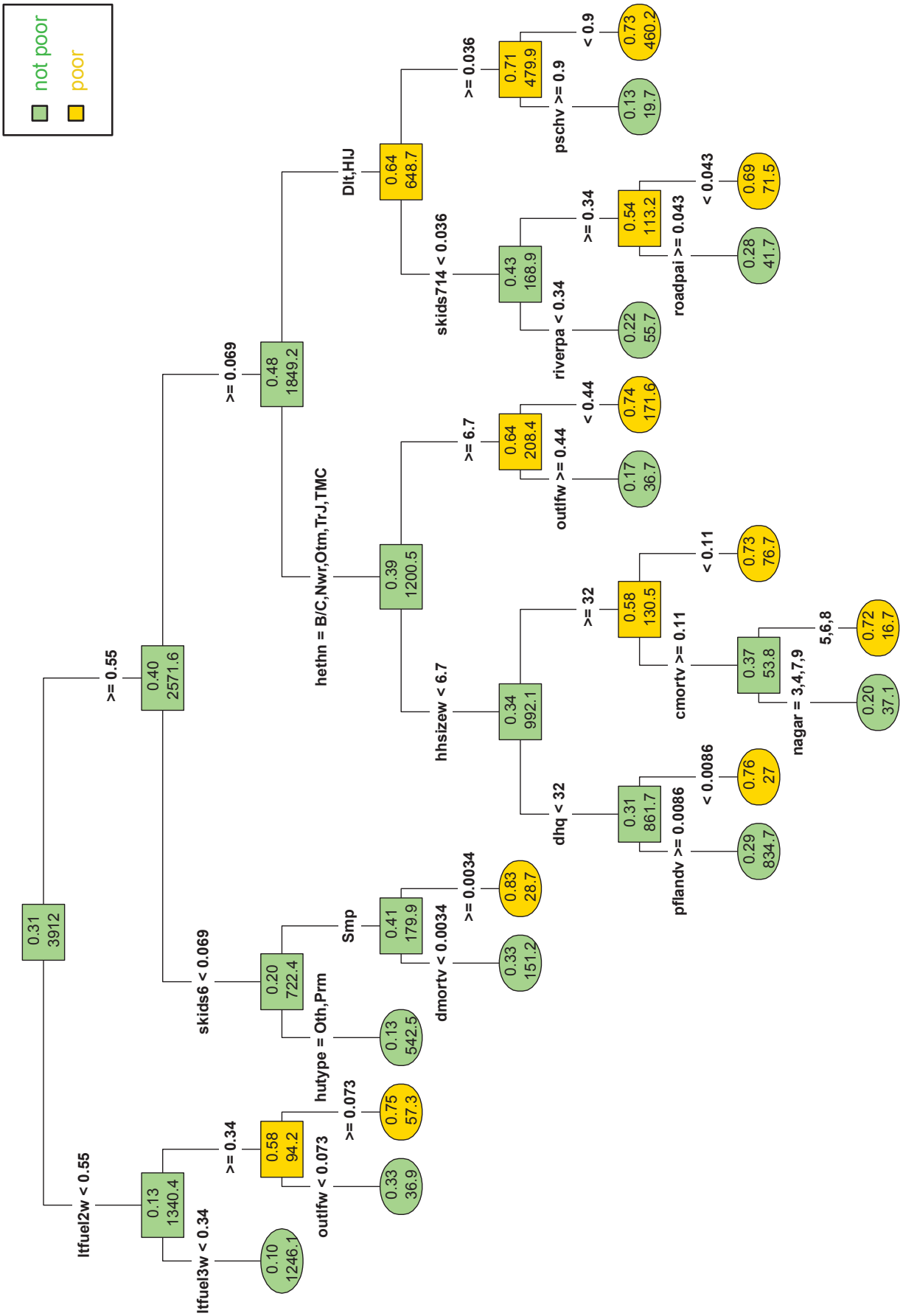


Figure 3.6: Weighted classification tree model for poverty incidence omitting *tww*, $cp = 0.006$

Figure 3.7: Competing splits for root node of weighted classification tree for poverty incidence

```
Node number 1: 3912 observations,
  predicted class=notpoor  expected
    class counts: 2705.24 1206.76
   probabilities: 0.692 0.308

  Primary splits:
      tvw      < 0.0560797  to the right, improve=182.4457, (0 missing)
      ltfuel2w < 0.550246   to the left,  improve=125.2889, (0 missing)
      toilet3w < 0.6319075  to the left,  improve=121.3443, (0 missing)
      skids6   < 0.1213236  to the left,  improve=118.1610, (0 missing)
      skids6w  < 0.1569847  to the left,  improve=115.7572, (0 missing)
      edulv5w  < 0.023703   to the right, improve=114.8974, (0 missing)
```

tree models. These include: having no proper toilet (*toilet3w*); a high proportion of children under five in the household (*skids6*); ethnicity of the household head (*hethn*); semi-permanent and temporary housing (*hutypeb*). A measure of variable importance for all predictors can be extracted from the model.

3.2.6 Variable importance and surrogates

An overall measure of contribution to the tree model can be provided for each predictor in the dataset. Ranking of variables in order of importance needs to consider variables which have high splitting power, but do not appear in the final tree structure because they are being masked by variables selected by the tree algorithm as providing the optimal splits. The best approach to account for masked predictors is based on surrogate splits (Breiman et al. 1984). Surrogate variables are designed to provide replacement splitting criteria when the chosen splitting variable has missing values (Berry & Linoff 2004). Surrogates are automatically constructed by *rpart*, even when all predictors have no missing values, as occurs in the Nepal dataset.

Figure 3.8 displays details of the first left hand split at the root node in the classification tree for poverty incidence in Nepal, together with the first five surrogate variables, the default option in *rpart*. The split at the root node is based upon the two categories " tvw < 0.0560797 " and " tvw ≥ 0.0560797 ". The first surrogate, *ltfuel2w*, is selected as the alternate independent variable which best predicts the two categories, " tvw < 0.0560797 " and " tvw ≥ 0.0560797 " (Therneau & Atkinson 2013). If *ltfuel2w* also had missing values, then the second surrogate variable, *ckfuel3w*, would be chosen to classify those observations with missing values, etc. An optimal split point and misclassification error are computed for each possible surrogate. We note that *ltfuel2w* also acts as the first competing split for *tvw*. Surrogates and competitor splitting variables provide different information. Competitor splits are those predictors which provide the same number of correct classifications as the primary split, whereas surrogates are selected to produce the same classification of the observations, a stricter criterion (Therneau & Atkinson 2000).

The technique of surrogate variables used in *rpart* is a variation of a common approach that uses other independent variables to estimate missing data points (Therneau & Atkinson 2013).

Figure 3.8: Splitting criterion and surrogate variables for root node in classification tree

```
Node number 1: 3912 observations,
  predicted class=notpoor  expected
    class counts: 2705.24 1206.76
   probabilities: 0.692 0.308

  Primary splits:
      tvw < 0.0560797  to the right, improve=182.4457, (0 missing)
  Surrogate splits:
      ltfuel2w < 0.696154   to the left,  agree=0.778, adj=0.520, (0 split)
      ckfuel3w < 0.0177675  to the right, agree=0.766, adj=0.493, (0 split)
      edulv5w  < 0.029566   to the right, agree=0.747, adj=0.453, (0 split)
      hethn6w  < 0.0005423  to the right, agree=0.741, adj=0.440, (0 split)
      motvehw  < 0.0056708  to the right, agree=0.739, adj=0.435, (0 split)

Node number 2: 2076 observations,    complexity param=0.005662089
  predicted class=notpoor  expected loss=0.1437145  P(node) =0.4620772
    class counts: 1547.86 259.785
   probabilities: 0.856 0.144
  left son=4 (1416 obs) right son=5 (660 obs)
  Primary splits:
      toilet3w < 0.5083335  to the left,  improve=25.07266, (0 missing)
  Surrogate splits:
      huown2w  < 0.9640862  to the left,  agree=0.860, adj=0.699, (0 split)
      ckfuel3w < 0.021432   to the right, agree=0.834, adj=0.644, (0 split)
      ckfuel4w < 0.077935   to the right, agree=0.826, adj=0.627, (0 split)
      pschv    < 0.7801231  to the right, agree=0.821, adj=0.617, (0 split)
      skids6w  < 0.1604274  to the left,  agree=0.819, adj=0.613, (0 split)
```

The importance score for a predictor is computed by taking the sum of its contribution to the model as a primary split or a surrogate variable, in terms of improvement in classification rate. This measure of reduction in impurity is represented in the summary output (Figure 3.8) by the term *improve*, which equates to information gain provided by the split multiplied by the number of observations in the node (Section 2.4.2.2) . So, at Node 1, the root node, a score of 182.4 is assigned to the importance measure of the variable *tvw*. The splitting variable at Node 2, *toilet3w*, receives 25.1 points towards its importance score.

Allocation of points for surrogate variables is calculated from $improve \times adj$ where *adj* quantifies the gain in purity when compared with the 'go with the majority' rule (Therneau & Atkinson 2000). Thus, the variable *ltfuel2w* gains $182.446 \times 0.52 = 94.87192$ points from its contribution as a surrogate at Node 1. The surrogate predictor *ckfuel3w* is assigned $182.446 \times 0.493 + 25.073 \times 0.644 = 106.0929$ points from Nodes 1 and 2 towards its importance score. The contribution of surrogate variables to the model input must be included in the assessment of variable importance to account for redundant variables (Therneau 2011). In the Nepal context, owning a tv set (*tvw*) is dependent, in

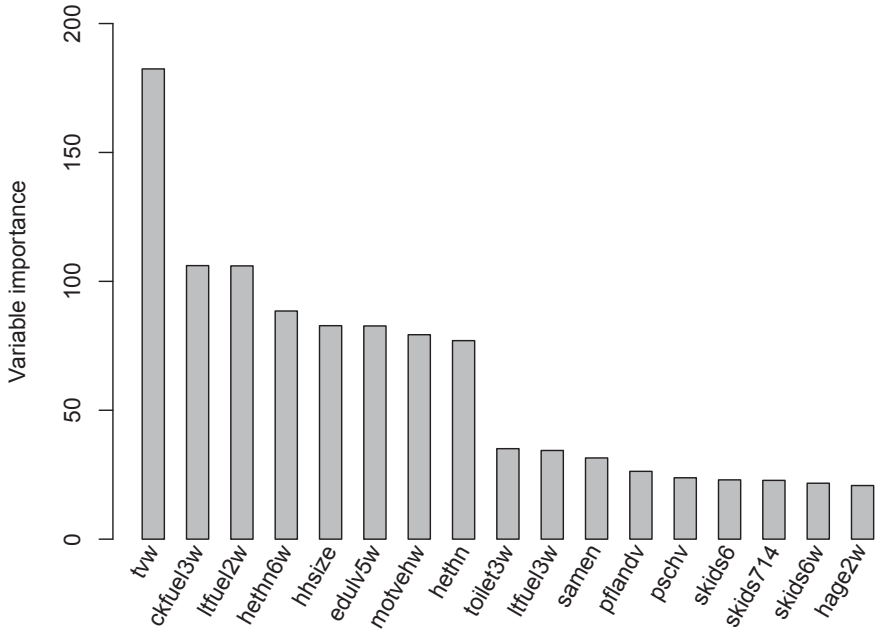
rural areas, on being able to afford kerosene to run a generator for lighting etc. (*ltfuel2w*). These two variables provide very similar information, and either variable might be chosen at a particular split. If surrogates were not included in the importance measure, the importance value would be split between *tvw* and *ltfuel2w*, and the true contribution to the model of each predictor would be undervalued (Therneau & Atkinson 2013). Thus, the measure of variable importance includes the contribution of a predictor at each level of the tree in which it appears as a splitting variable, and also its effect as a surrogate variable. A variable is assigned a score to contribute to its measure of importance only if it does better than the "blind rule", 'go with the majority', which essentially reproduces the splitting proportions of the primary split. If p is the proportion of observations which meet the splitting criterion, then the "blind rule" has misclassification error of $\min(p, 1-p)$. Table 3.1 lists the importance scores, in decreasing order, for the seventeen most important primary and surrogate splitting variables in the classification tree model for poverty incidence in Nepal (Figure 3.5). For this model, there are sixty-eight predictors with non-zero importance scores, so the table is restricted to values above 20. Since the value listed for *tvw* in the importance table (Table 3.1), 182.4, represents its contribution to the model at the first split, it is clear that *tvw* does not act as a primary split or surrogate at any other node in the tree.

Table 3.1: Scores for the seventeen most important predictors in the weighted classification tree: *hh* means household

Variable	Description	Importance
tvw	% hh in ward own televison	182.4
ckfuel3w	% hh in ward use LP/gas for cooking	106.1
ltfuel2w	% hh in ward use kerosene for lighting	106.0
heth6w	% in ward hh head ethnicity Terai Jajajatis	88.5
hhsize	household size	82.8
edulv5w	% in ward age 15+ with 11+ years schooling	82.7
motvehw	% in ward own motor vehicle/ motorbike	79.3
hethn	ethnicity of hh head	77.0
toilet3w	% hh in ward with flush toilet	35.1
ltfuel3w	% hh in ward don't use kerosene or electricty for lighting	34.4
samen	% of adult men in hh	31.5
pflandv	% in VDC with land-owning females	26.3
pschv	% in VDC aged 6 - 16 attending school	23.8
skids6	% children aged 0 - 6 in hh	23.0
skids714	% children aged 7 - 14 in hh	22.8
skids6w	% children in hh aged 0 - 6, ward average	21.7
hage2w	% in ward hh head aged 30 -44	20.8

A plot of importance scores for the eighteen most influential predictor variables is provided in Figure 3.9. The proportion of households in the ward which own a television is quantified by the classification tree model as the most significant determinant of poverty status. We recall from Section 3.2.5 that tv ownership is related to being wealthy enough to run a kerosene generator (*ltfuel2w*) for electricity production, especially in rural wards.

Figure 3.9: Plot of variable importance for classification tree with $cp = 0.005$



The other predictors listed fall into two categories; those with importance score between 70 and 110, and variables with scores between 20 and 40. Reduction in impurity value at the root node is as least twice that for any other node in the tree, and consequently, all surrogate splitters at the root node appear in the top seven most important predictors in the model.

These eighteen most influential variables (Figure 3.9) include indicators of wealth, such as owning a tv, having a kerosene generator for the household, owning a motor vehicle (*motvehw*), the household head having had eleven or more years of schooling (*edulv5w*) and a high proportion of adult men in the house. Household poverty is seen to be related mainly to larger households, the number of young children in the household, poorer options for cooking fuel such as wood and dung, no proper toilet and the ethnicity of the head of the household. However, these variable importance measures should be interpreted with caution. Breiman et al. (1984) stakes no claim that the method applied in the CART algorithm is intrinsically the best. In addition, different models can provide very different importance scores for the same predictors. We test this assertion by examining the effect of applying alternate pruning methods to optimising the cost-complexity parameter. Utilising *cp* as the pruning criterion produced branches which terminate at the first level for the unweighted tree and at the second level for the weighted tree, where the root node is at level zero, or tree depth of zero. Another option for restricting tree size is to set $cp = 0$ and specify the tree depth (Section 2.4.2.4). Taking this approach produced a classification tree for poverty incidence which continued to split past the first and second levels. Figure 3.10 displays a tree with $cp = 0$ and tree depth of 4. This particular depth is chosen for ease of readability.

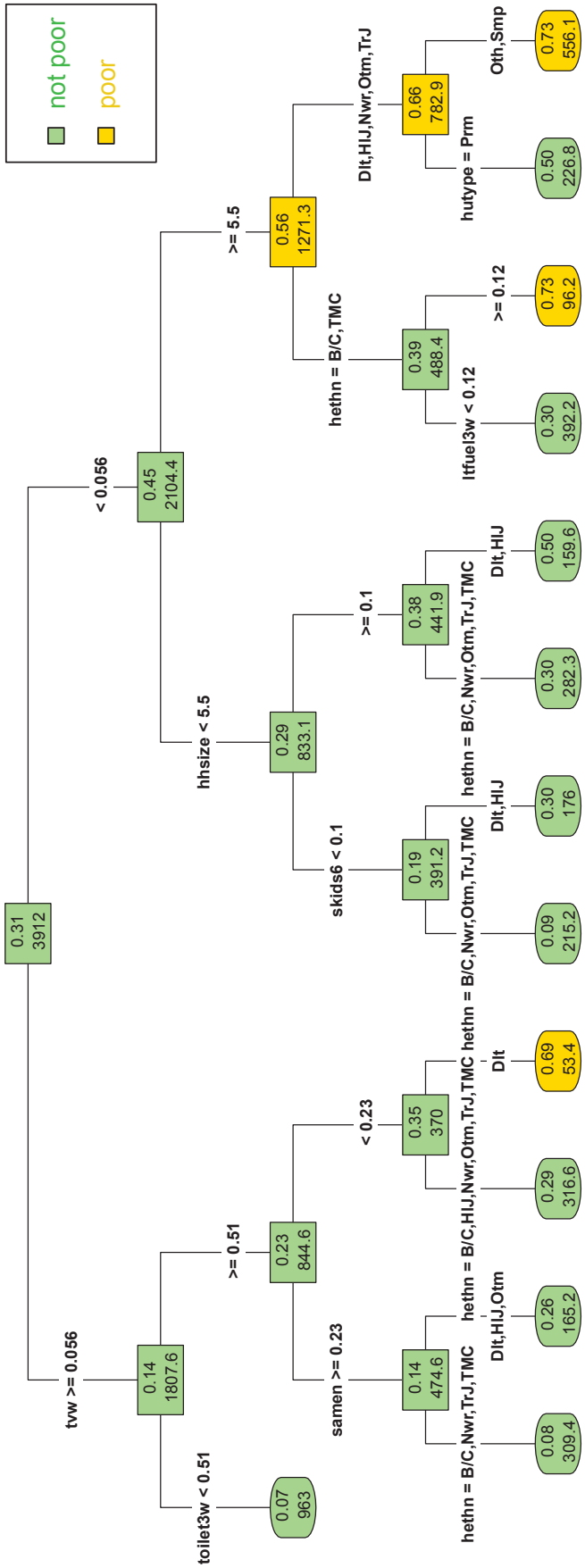
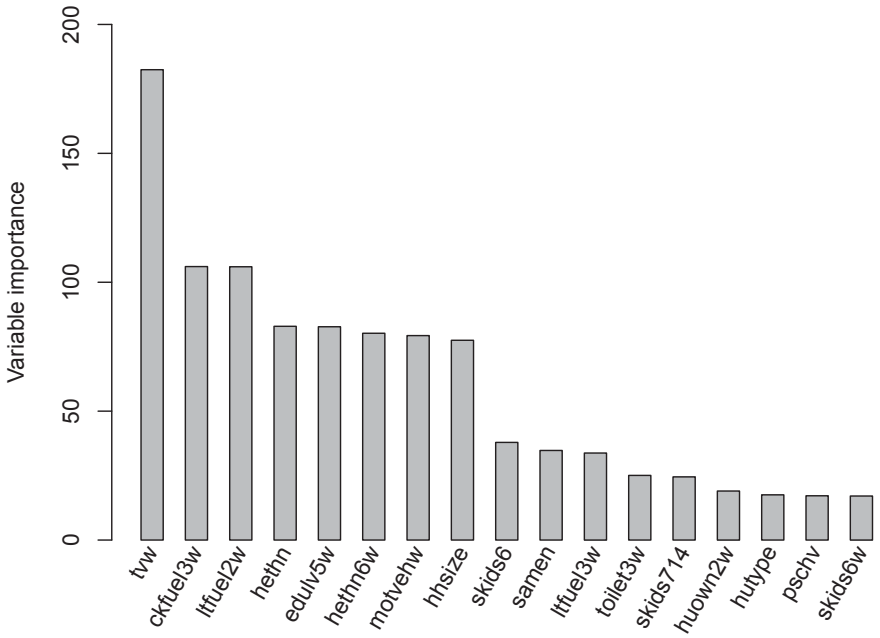


Figure 3.10: Classification tree model for poverty incidence with $cp = 0$ and tree depth 4

The tree diagram for a tree with depth 5 has terminal nodes with text which overlaps, and so is difficult to decipher. The graph displaying eighteen highest importance scores for the tree model having depth of 4 is given in Figure 3.11. Despite Breiman's (1984) reservations, the weighted classification trees built using different methods for restricting tree size have produced very similar importance values (compare Figure 3.9 with Figure 3.11). The only differences are in some rearrangement of the order of variables within the two groupings, scores between 70 and 110, and scores between 20 and 40.

Figure 3.11: Plot of variable importance for classification tree with $cp = 0$ and depth 4



It is useful to investigate whether the variables identified by the classification model as being influential in determining poverty status are also important predictors in the ELL model. Darlington (1968) discusses various measures of variable importance in the context of linear regression models. Two variance importance measures are the beta weights and "usefulness". The usefulness of a predictor is the decrease in $\bar{R}^2$ when the predictor is removed from the regression equation, where $\bar{R}^2$ represents the square of the population multiple correlation. The beta weight, β_j, is the weight assigned to predictor X_j when all variables have been adjusted to unit variance. Inspection of beta weights for the ELL model of log expenditure (Haslett & Jones 2006) shows that many of the variables with the largest beta weights are also listed as the most important in the tree model. The main discrepancies are *tvw*, which has reduced importance in the ELL model, and the variable *dmortv*, the ward mortality rate due to infectious disease, which has the largest beta weight, by a considerable distance. Due to correlations between predictors variables, the beta weights are not a reliable measure of variable importance in the regression context (Darlington 1968).

The differences in variable importance between the ELL and tree models may also be due to the intrinsic structure of the two techniques. In the linear regression model, the beta weights are constructed so as to adjust for the effect of other variables present in the model, using Type 3, or partial, sums of squares (Draper & Smith 1998). In contrast, the recursive algorithm in the tree model is akin to applying Type 1, or sequential, sums of squares. Adjustment at a particular node in a tree is with respect to splitting variables above that node, a subset of variables which make up the same branch. Type 1 regression sums of squares measure the additional variability explained when a new predictor is added to the model (Steel et al. 1997). A comparison of Figure 3.5 with Figure 3.6, the weighted classification tree model with and then without *tvw* as a predictor, illustrates a disadvantage of the classification tree technique, the inherent instability of the tree structure (Breiman et al. 1984). Removing a single variable, *tvw*, has changed the structure of the tree markedly, with several new splitting variables introduced into the tree. This characteristic, the instability of tree models, becomes problematic when generating standard errors of prediction using a classification tree model for poverty incidence. This issue is discussed in detail in Chapter 4. Having chosen a suitable model, the next step is an examination of the goodness-of-fit of the model.

3.2.7 Assessing model fit

As discussed in Section 3.2.4, the usual practice in tree modelling is to divide the data into a training set for model building and a test set to assess model accuracy. However, reducing the size of the dataset can lead to model instability (Section 3.2.5) which makes the choice of training set problematic. In addition, the test set must be independent of the training set, so the dependence structure within each cluster must be taken into consideration when constructing a test set. This issue can be addressed through the use of replicate subsamples, partitions of the data which replicate the survey design.

The Nepal dataset comprises 326 clusters as primary sampling units (psu's) which are known as "ilakas". Each ilaka consists of 12 households. A replicate sample of size 326 can be constructed by randomly sampling a single household from each ilaka. Hence, 12 independent replicate subsamples each of size 326 can be created by random sampling without replacement from the 326 clusters. Each replicate can act as a test set, with the remaining 11 replicates comprising the training set. This procedure provides 12 independent assessments of model accuracy as well as a measure of variability.

Let R_i represent the i^{th} replicate subsample for $i = 1, 2, \ldots, 12$. A classification tree model is built using the i^{th} training set which consists of the dataset excluding R_i. This model is then applied to the corresponding test set, R_i, to obtain a prediction of poverty status for each household in R_i. However, the i^{th} replicate contains a variable describing the actual poverty status for each household, so that actual poverty status can be compared with predicted poverty status to provide an estimate of the misclassification rate of the model. The information can be displayed in a *confusion matrix* (Han et al. 2012), which consists of the following four measures:

- True positives (TP) which is the total number of poor households that were correctly classified as poor
- True negatives (TN) which is the total number of nonpoor households that were correctly classified as not poor
- False positives (FP) which is the total number of nonpoor households that were incorrectly classified as poor
- False negatives (FN) which is the total number of poor households that were correctly classified as being not poor.

A confusion matrix can be displayed in a 2 x 2 table as shown in Figure 3.12.

Figure 3.12: Layout of a confusion matrix

	Predicted Poverty	
Actual Poverty	NotPoor	Poor
NotPoor	TN	FP
Poor	FN	TP

A confusion matrix was generated for each of the test sets constructed from the 12 replicate subsamples R_i. The mean and standard deviation of these 12 estimates of the four measures of classification accuracy (TP, TN, FP and FN) were then computed, and are displayed below. When the replicate subsamples were used to provide training and test sets, the classification tree model produced a misclassification rate (Figure 3.13) for non-poor households which is reasonable at 8%, but a 59% misclassification of poor households seems large. Modelling poverty incidence in Nepal using the ELL regression method provided a goodness-of-fit measure of $R^2 = 55\%$ (Haslett & Jones 2006), a value which is considered acceptable in models for poverty mapping (Demombynes et al. 2007, Jamal 2005).

Figure 3.13: Aggregated measures of classification accuracy from models based upon replicates

Means of classification accuracy measures

	PredictedPoverty	
ActualPoverty	NotPoor	Poor
NotPoor	0.92	0.08
Poor	0.59	0.41

Standard deviations of classification accuracy measures

	PredictedPoverty	
ActualPoverty	NotPoor	Poor
NotPoor	0.022	0.022
Poor	0.069	0.069

Different approaches have been suggested to address a class imbalance problem in tree models: applying a cost measure for incorrect classification of the minority class, the class of interest (Elkan 2001); under-sampling the majority class or over-sampling the minority class (Kotsiantis et al. 2006); boosting (Freund & Schapire 1999), which involves successive iterations of the tree with misclassified observations in one iteration given a higher weighting in the next iteration. Reducing the misclassification rate of poor households in the survey dataset by direct means is not a focus of this research, but could be an important direction for future research.

We complete the chapter on introducing classification trees for poverty incidence by applying the model built from the Nepal survey data to census data to generate point estimates of poverty incidence for a specific district in Nepal. Two types of prediction are produced, hard and soft, and compared with the corresponding estimates obtained from the ELL modelling of poverty incidence in Nepal.

3.3 Generating small area estimates of poverty incidence

3.3.1 Hard and soft predictions

Poverty incidence is defined as the proportion of individuals in poverty across a particular domain, obtained by aggregating the poverty status of individuals in that domain. The Foster, Greer and Thorbecke identity for poverty incidence has the form (Section 1.2),

$$P_0 \;=\; \frac{1}{N}\sum_{n=1}^{N} \mathrm{I}\left(\mathcal{E}_n < \mathrm{z}\right) \;, \tag{3.4}$$

where I represents the indicator function, $\mathcal{E}_n$ is the estimate of per capita expenditure for the n^{th} individual and z denotes the poverty line, which equates to 7695.744 in average 2003 Nepalese rupees in the Nepal analysis. The ELL technique for poverty mapping models log per capita expenditure from the survey data and generates predictions of log per capita expenditure from the census data, which are then exponentiated to provide values of $\mathcal{E}_n$ to be applied to Equation (3.4). A classification tree model provides a simpler process for generating poverty estimates by modelling poverty directly. When generating small area estimates of poverty incidence, *P0*, the natural approach is to obtain a prediction of poverty incidence for each census household and then aggregate these predictions across the small domains of interest. However, the poverty status of individuals rather than households is the preferred indicator of poverty incidence, so the classification tree predictions at household level are weighted by the household sizes before amalgamating to the small area estimates.

New cases from the census data are classified by the tree according to the leaf at which they terminate. A majority rule is applied, so that each leaf is designated as *poor* or *not poor* based on the status of the majority of survey households which migrate to that leaf. In addition to being allocated a poverty status, the k^{th} leaf can be assigned a

posterior probability of being poor, p_k (Section 2.4.2.2) defined as the proportion of poor households in the k^{th} leaf. When $p_k \geq 0.5$, then the k^{th} leaf is classed as poor. Thus classification trees can provide two types of prediction, in this research labelled *hard* and *soft*. For a hard type of tree prediction, the i^{th} census household is assigned the class designation of the leaf at which it terminates; either $Y_i = 1$ if the leaf has classification of "poor", or $Y_i = 0$ if the leaf designation is "not poor". When a terminal node of the tree model is classified as poor, every household in that terminal node is designated as being poor. A hard prediction of poverty incidence for an individual household is then analogous to assigning the mode of a leaf to each household in the leaf. Let S_k represent the subset of census households which emerge at the k^{th} leaf, and n_i denote the household size, i.e. total number of people in the household, for the i^{th} census household, where $i \in S_k$. Then a hard classification tree estimate of poverty incidence, $P0^{(h)}$, the proportion of poor, for a given small area is obtained by summing the number of poor people, $n_i\, Y_i$, over all census households allocated to a particular leaf, summing across the leaves and then dividing by the total number of people in the small area, as follows:

$$P0^{(h)} \;=\; \frac{\sum\limits_{k}\sum\limits_{i\in S_k} n_i\, Y_i}{\sum\limits_{k}\sum\limits_{i\in S_k} n_i}\,. \tag{3.5}$$

A soft tree prediction for the i^{th} census household, where $i \in S_k$, is p_k, the posterior probability of being poor for households in the k^{th} leaf. Then, $P0^{(s)}$, the soft classification tree estimate of poverty incidence for the small area being considered is defined as:

$$P0^{(s)} \;=\; \frac{\sum\limits_{k}\sum\limits_{i\in S_k} n_i\, p_k}{\sum\limits_{k}\sum\limits_{i\in S_k} n_i}\,. \tag{3.6}$$

At the prediction stage, the poverty status of each household in the census is unknown but is estimated from the tree, which is fixed, and so the Y_i's can be considered as Bernoulli random variables. For the i^{th} census household, such that $i \in S_k$, $Y_i \sim \text{Bern}(p_k)$. Consequently, $P0^{(s)}$ is equivalent to the expected value of poverty incidence across the leaves of the tree, the posterior mean of the proportion of poor individuals in the small area, since:

$$\text{E}\left[\frac{\sum\limits_{k}\sum\limits_{i\in S_k} n_i\, Y_i}{\sum\limits_{k}\sum\limits_{i\in S_k} n_i}\right] \;=\; \frac{\sum\limits_{k}\sum\limits_{i\in S_k} n_i\, p_k}{\sum\limits_{k}\sum\limits_{i\in S_k} n_i}\,. \tag{3.7}$$

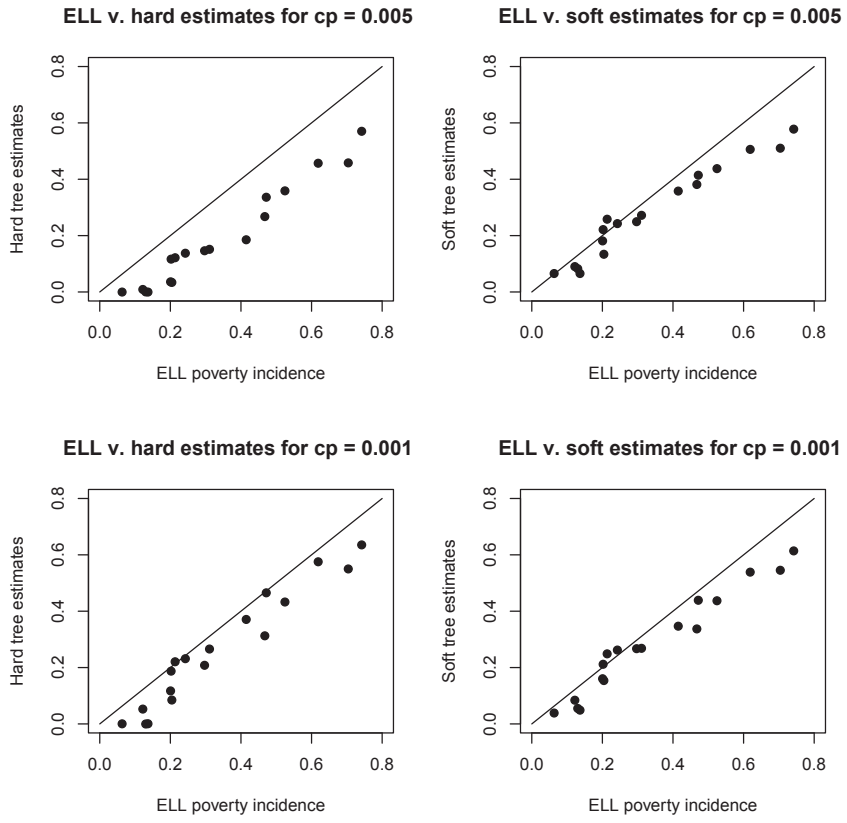
3.3.2 Small area estimates of poverty incidence for a district in Nepal

Hard and soft small area estimates of poverty incidence in Nepal was generated using the *R* function *predict* (R Core Team 2015), according to Equations (3.5) and (3.6). The data used for predictions comprised a subset of the 2001 Population Census of Nepal, representing a district in Nepal chosen to have a wide range of poverty incidence estimates from the Haslett & Jones (2006) study. The district comprises twelve ilakas, the small

domains for estimation purposes. Census data from the district consisted of the same set of common variables used in the survey dataset to build the classification tree model (Section 3.2.1).

Figure 3.14 displays plots of the ELL district predictions against hard and soft predictions from the tree modelling, for the original weighted tree model chosen, with $cp = 0.005$, and also for a larger tree, having $cp = 0.001$. The hard tree estimates are consistently lower than the corresponding estimates from the ELL method. This pattern is worse for the smaller tree, having the larger cp value of 0.005. An explanation for the "apparent" underestimation of poverty levels is the discrete nature of hard tree prediction. A classification of poor is assigned to terminal nodes with proportions of poor households being between 50.01% and 100%. For example, a group of 40 households emerging at a terminal node with posterior probability of 40% would all be classed as not poor using a hard estimate type, whereas each household has probability of 0.4 of being poor, for a soft estimate. So, the expected number of poor households for that leaf will be 16. The term "apparent" is used because there is no "gold standard" (Francq & Govaerts 2014, Kang et al. 2013) for small area estimates of poverty incidence in Nepal, against which to measure the accuracy of the tree estimates, and indeed the ELL estimates.

Figure 3.14: ELL predictions compared with hard and soft tree predictions for two cp values



The soft predictions, the expected value of poverty incidence across an ilaka, are closer to the ELL estimates than the hard predictions. The discrepancy between ELL and tree predictions tends to be worse for the higher levels of poverty. Using a bigger tree ($cp = 0.001$) slightly improves the agreement between estimates from the different types of modelling.

3.4 Conclusions

Results so far from the classification tree modelling have proved promising, in providing estimates of poverty incidence similar to those obtained from the ELL technique. These two methodologies are essentially regression techniques, so the tree model gives some insights into the ELL model. The tree based model has some advantages over the ELL regression type approach. The tree provides a model which is easy to interpret. Variable interactions are automatically catered for in the tree, whereas they must be tested individually in the linear regression model, a tedious process with a large set of possible predictors. The most important variables are chosen first in the tree model, and are clearly identified in the tree structure. In the poverty incidence model, the algorithm has separated out households which are not poor, to more effectively identify households in need.

The structure of the two methodologies are similar, in that a model is fitted to survey data and then applied to census information to generate predictions. The difference is in the type of model fitted, a regression model in ELL versus a tree based model. A concern about the ELL method is that the standard errors of prediction produced by the model are conditional on the model structure being correct, i.e. that the correct predictors are used in the model and that reasonably accurate parameter estimates result. Building a classification tree for poverty incidence has utilised a model with a completely different structure to the ELL model, and yet has produced very similar poverty estimates. However, generating only point estimates for poverty measures is not sufficient, an indication of the precision of these estimates is also required. Using tree based models to produce standard errors of prediction has not to date been achieved, regardless of the survey design of the data, whether a simple random sampling design or a complex survey structure. Developing a method for providing standard errors from tree based models is a core component of the thesis.

The main disadvantage of trees is the intrinsic instability when the dataset changes, either in terms of predictors or observations. The unstable nature of trees occurs because the tree structure depends upon the first split. In modelling a single tree, the classification tree technique has made no provision for estimating sampling errors. However, applying a variance estimation technique to build multiple trees and provide multiple estimates can be problematic due to tree instability. Variance estimation requires resampling of the survey data. Different subsamples can produce very disparate tree structures, and unstable estimates of poverty incidence. This problem is discussed in the next chapter in the context of using replicate samples for estimation of standard errors.

Chapter 4

Tree instability under resampling

4.1 Introduction

Chapter 3 has outlined the construction of a weighted classification tree model from the Nepal survey data, which was then applied to census data from a district in Nepal to provide estimates of poverty incidence over eighteen ilakas which comprise the district. The small area estimates of poverty incidence in each ilaka were found to be similar to published estimates produced by ELL modelling (Haslett & Jones 2006). Tree-based models have to date been used solely for prediction, but poverty mapping involves not only generating point estimates for poverty but also standard errors of prediction. This chapter outlines the problems which arose when the classification tree methodology was extended to provide a measure of uncertainty for small area estimates.

The survey data used for poverty mapping in Nepal (Central Bureau of Statistics, Nepal 2004a,b) has a complex structure (see Section 2.2.2), comprising stratification and clustering. With a complex survey structure in the data, the variance of the estimator does not have a tractable mathematical form or an adjustment for weightings, so some type of variance estimation procedure was required .

Resampling schemes for variance estimation, such as jackknife, bootstrap and replication, involve generating subsamples of the original survey sample to provide multiple estimates, from which an estimate of mean and variance can be computed. The structure of the Nepal survey dataset lends itself to the creation of replicate subsamples for variance estimation since each sampled psu has twelve households, so that selecting one household per psu provides twelve replicates. To estimate the standard error of prediction for poverty incidence using the tree model, the jackknife method of variance estimation was applied to replicate samples of the survey data.

Generating multiple estimates from jackknife subsamples of replicates proved problematic. Large estimated standard errors resulted from tree instability caused by the small number of observations in the replicates and jackknife subsamples. Tree instability is mainly due to the hierarchical nature of the algorithm, since an error in an upper split continues on down to successive splits and increases (Hastie et al. 2001). This chapter

discusses types of tree instability and possible explanations. The issues examined here included the difficulties in using *cp* for tree pruning, competing splits, and the effect of weights on minimum split. For conciseness, only the hard type of classification tree estimate is discussed in this chapter.

4.2 Variance estimation for poverty incidence in Nepal

In the poverty mapping scenario the primary sampling unit, psu, is a cluster of households, each cluster tending to be fairly homogeneous. The statistical dependence between households in each cluster needs to be accounted for when estimating variance. The linear regression mixed model used in the ELL methodology allows for the dependency of households within a cluster by incorporating a random component in the model to represent the clusters, the psu's, in addition to weighting the observations to adjust for unequal selection probabilities. Variance estimation in ELL is achieved through applying the bootstrap technique to three sources of variability: regression coefficients, cluster effects and household effects (see Section 2.3.3). The task of the thesis is to develop a suitable procedure for estimating the precision of predictions generated by the tree model. Variance estimation procedures involve some type of replication of the original data to generate multiple estimates, in order to provide a point estimate and error measurement of the quantity of interest. Bootstrap and jackknife are the two most commonly used variance estimation techniques, but the structure of the Nepal survey data lends itself to an alternative resampling procedure.

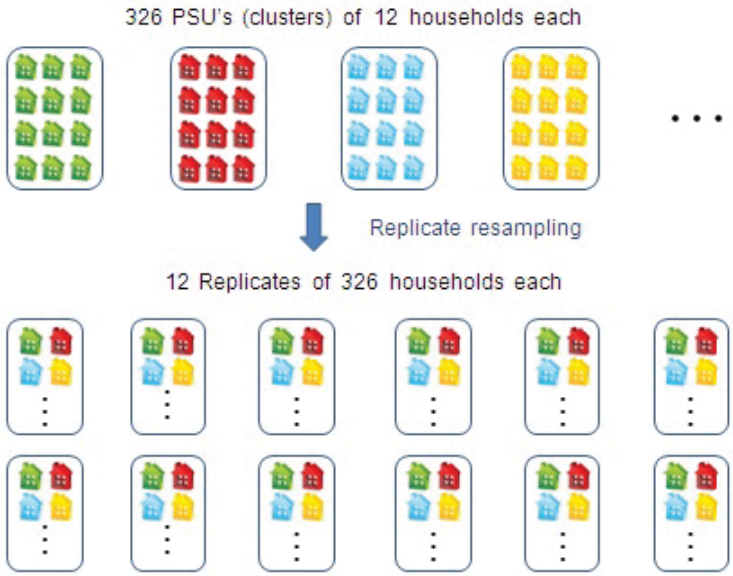
The survey sampling design for the Nepal data involves probability proportional to size (PPS) selection of clusters, the primary sampling units, and systematic sampling of twelve households within each chosen cluster. Random sampling within clusters of a fixed number of households is a common survey sampling scheme which gives rise to an *epsem* selection structure, an equi-probability selection method at household level. This type of survey design provides a neat resampling scheme for variance estimation, by creating replicates which are mirrors of the full sample.

4.2.1 Replicate subsamples

The effect of constructing replicates is to reduce the complex survey design to a set of simple random samples, an example of inverse sampling as described by Hinkins et al. (1997). Replicate subsampling in the Nepal context, selection of a single household at random from each cluster in the original complex survey sample, corresponds to the application of inverse sampling (Section 2.2.3.7) to clustered data, as proposed by Hoffman & Weinberg (1998), and discussed in Rao & Scott (2000). The Nepal dataset comprises 326 psu's each comprising 12 households. Replicate subsamples were created by random allocation of a single household from each psu to a particular replicate, thus providing 12 replicates each containing 326 households.

Selection of the replicate subsamples was achieved as follows. Each household in a psu is randomly assigned a number between 1 and 12. Then Replicate 1 comprised each household with the index 1, Replicate 2 was made up of all households with index 2, etc. This process resulted in subsamples, the replicates, which contain independent observations, and so standard variance estimation formulae could be applied. Each cluster is fairly homogeneous, households within a cluster are similar, but the clusters are independent units. So, households from different clusters are design independent. Figure 4.1 illustrates the construction of the replicates. Association between households in a cluster is indicated by all households within the same psu being coloured alike. Households within a replicate have different colours, representing the independence of observations within the replicate subsample. The replicate subsamples constructed from the Nepal survey data were used to estimate the variability of predictions of poverty incidence.

Figure 4.1: Construction of replicate subsamples



4.3 Variance under inverse sampling

The variance formula under inverse sampling (Section 2.2.3.7) has the form (Rao et al. 2003).

$$\hat{V}_{IS} = \frac{1}{g}\sum_{r=1}^{g}\hat{V}_r^* - \frac{\sum_{r=1}^{g}\left(\hat{\theta}_r^* - \hat{\theta}_{IS}\right)^2}{g} \tag{4.1}$$

where $\hat{\theta}_r^*$ and $\hat{V}_r^*$ denote, respectively, the estimate of the parameter of interest θ and its variance derived from the r^{th} inverse subsample, and $\hat{\theta}_{IS}$ is the average of the estimates $\hat{\theta}_r^*$ computed from the g inverse subsamples,

$$\hat{\theta}_{IS} \;=\; \frac{1}{g}\sum_{r=1}^{g}\hat{\theta}_r^{*}\,.$$

The form of the variance estimator reflects the fact that the estimates generated from individual inverse subsamples are conditional on the original sample survey, and is derived from the standard relationship between conditional and unconditional expectations (Rao & Scott 2000). In the context of the Nepal survey data, replicate samples were obtained by random selection of a single observation from each cluster. When each replicate sample was constructed the within-cluster effects were removed. Variability between replicate estimates, the $\hat{\theta}_r^*$'s, is represented by the second term in Equation 4.1, which also expresses the variation within clusters. The first term in Equation 4.1 measures the average of the $\hat{V}_r^*$'s, variability within each replicates, and, equivalently, between clusters. In most applications of the variance estimator under inverse sampling, an explicit formula will exist for the $\hat{V}_r^*$ terms, but not when using a classification tree model.

Variability between the replicate estimates, the $\hat{\theta}_r^*$'s, was estimated by building a tree model from each replicate, then using that model to generate predictions from census data, at household level. These household level estimates were aggregated to provide a small area estimate of poverty incidence, $\hat{\theta}_r^*$, for each replicate subsample, and thus an estimate of the between replicate error. However, in calculating the within variability for each replicate, the $\hat{V}_r^*$'s, the first term in Equation 4.1, a problem arose, since each replicate sample provided only one tree model and thus only a single prediction. In addition, no analytical method exists to provide standard errors of prediction from a tree model. So, some type of double-sampling of the replicates was needed in order to estimate the variability within each replicate, $\hat{V}_r^*$. This was achieved using jackknife subsampling of the replicates. The between replicate variability was investigated first. A single weighted classification tree model was built from each replicate subsample, including adjustment of household weightings to account for the structure of the replicate subsamples, and to ensure each tree was representative of the whole population.

4.4 Replicate weights

For surveys such as the Nepal Living Standards Survey, the use of probability proportional to size sampling of clusters within strata, in conjunction with the selection of an equal number of households per cluster, results in equal weights for the selected households within each chosen cluster. The survey household weights indicate the number of households in the population represented by a particular household selected. However, the weights used in the tree modelling process need to be at person level, so that the sum of the household weights in the model is a measure of the total population size. When a replicate subsample is taken, by randomly selecting one household per cluster, the model weights of the selected households must be adjusted to reflect the fact that each chosen household now represents a whole cluster. There are several ways of doing this. The household weight could be multiplied by twelve, since each household in the replicate rep-

resents the twelve households comprising the cluster from which it was selected. However, since the statistic being estimated is a proportion (mean), multiplication of each weight by the same value would retain the same proportions of representation as would be achieved by using no scaling. In addition, this rescaling scheme does not allow for differences in household size between the single household selected for a replicate and the remaining households in the cluster it is intended to represent. An alternative approach is to scale up the household weight using the total weight for the cluster from which it was taken.

The approach taken initially was to use the most precise method, scaling up the household weights using the total cluster weight. Let w_{ij} denote the weighting assigned in the tree model to the i^{th} household in the j^{th} cluster, C_j, so that w_{ij} indicates that number of individuals in the population represented by that household. Then the adjusted replicate weight for the household was,

$$w^*_{ij} = \frac{\text{total model weights for replicate}}{\text{model weight for household}} = \frac{\sum_{i \in C_j} w_{ij}}{w_{ij}} \tag{4.2}$$

In addition, the weights were normalised so that they approximated to the sample size, the number of observations used to build the tree model, 326 for a replicate. The other aspect of tree building that needed consideration was how large to make the trees.

Since the replicate subsamples are being used to provide information about the variability in the model, trees based on the replicates should be constructed to be "similar" to the tree built from the full survey dataset, so that the variability across the replicate trees is representative in some way of the inherent variability in the full tree. How to determine "similarity" is problematic. Is it represented by similar splitting variables and tree structure? Section 3.2.5 discussed predictors which acted as competing splits for the chosen splitting variable at the root node of the weighted classification tree, and showed that they provide similar information, in terms of misclassification rate, to the primary splitter. So, the issue of similarity being based on splitting rules does not seem to be critical, except to note that the "greedy" nature of the CART algorithm used in *rpart* means that each split in the tree is dependent only on the criterion used in the previous split.

Tree complexity could be used to indicate similarity. The next question that arises is deciding upon an appropriate measure of tree complexity. Possible options are fixing the number of splits (and leaves) in the tree or using the same stopping rules. The *rpart* algorithm (Therneau et al. 2013) has no provision for predetermining the number of splits in the tree, but does provide various stopping criteria, used to prune the tree. These include setting the value of the complexity parameter, specifying the minimum number of observations in a node before a split can occur and fixing the depth of the tree (see Section 2.4.2.4). The complexity parameter is the most commonly used stopping rule, so this method was applied to the replicate subsets in order to produce classification tree models from the replicates which were "similar" to the the weighted tree model (Figure 3.5) for the full Nepal survey dataset.

4.5 Using the complexity parameter for tree pruning

Pruning the tree reduces the model to a suitable size to avoid overfitting. Tree pruning using the complexity parameter involves balancing the complexity or size of the tree with node impurity, the amount of misclassification at each node. As discussed in Section 2.4.2.4, this is achieved by minimising the cost-complexity measure $R_\alpha(T)$ for a given tree T (Breiman et al. 1984),

$$R_\alpha(T) \;=\; R(T) \;+\; \alpha|T|\,, \tag{4.3}$$

where $R(T)$ is the misclassification cost, based on the Gini index, $2p\,(1-p)$. Tree complexity, the tree size in terms of the number of leaves, is denoted by $|T|$ and α is the cost-complexity parameter.

To ensure that only the complexity parameter determines tree size, the minimum split was set at 3 and tree depth was set at the default value, 30 (Therneau 2011). The approach taken to select a suitably sized tree is similar to that described in Section 3.2.4. The Nepal survey dataset was subdivided into twelve replicate samples, as outlined in Section 4.2.1. A weighted classification tree model was built from data in Replicate 1 only, using a small value of the complexity parameter, $cp = 0.001$, and the results of this model used to determine the optimal tree size and corresponding *cp* value. Figure 4.2 displays a section of the output for the model with $cp = 0.001$.

Figure 4.2: Table of *cp* values and associated cross-validation error for different tree sizes

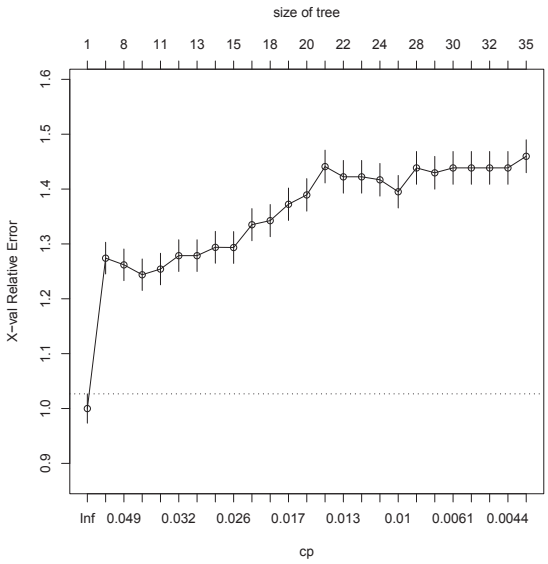
```
Root node error: 74.118/326 = 0.22735

n= 326

          CP nsplit rel error xerror    xstd
1  0.0769799      0 1.0000000 1.0000 0.10210
2  0.0513279      4 0.6612165 1.2917 0.11095
3  0.0470379      7 0.4864633 1.2222 0.10912
4  0.0416201      8 0.4394254 1.2115 0.10883
5  0.0337252     10 0.3561851 1.2278 0.10928
6  0.0308066     11 0.3224598 1.2297 0.10933
7  0.0286007     12 0.2916532 1.2534 0.10996
.........................................

23 0.0056494     31 0.0126439 1.3855 0.11316
24 0.0034973     32 0.0069945 1.3793 0.11302
25 0.0010000     34 0.0000000 1.3793 0.11302
```

The *rpart* algorithm has produced a sequence of nested trees, with corresponding *cp* values, from the smallest tree consisting only of a root node to the largest having $cp = 0.001$. Ten-fold cross validation was used to estimate model error for trees of different sizes (see Section 2.4.3), expressed in terms of the number of splits in the tree, one less

than the number of leaves. The third column of the model output shown in Figure 4.2 lists misclassification error for the full tree of a specified size. The next two columns tabulate average misclassification rate and standard error for the cross validation subsets. Relative error values are displayed, obtained by dividing absolute error by misclassification rate at the root node, a scaling carried out in order that the root node has relative error of 1.0.

In Figure 4.3 the cross-validation error is plotted against *cp* value for the different sized weighted classification trees listed in Figure 4.2. The expected pattern in *cp* values is a sharp drop in cross-validation error (*xerror*) followed by a fairly flat plateau (Therneau 2011), and the optimal *cp* value is taken to be the smallest value within one standard deviation of the minimum (as represented by the dotted line in the plot). Instead, the cross-validation errors for all sized trees constructed from subsamples of Replicate 1 are greater than that for the tree with only a root node. This result suggests that the best model is a tree with no splits, since a tree with any branches overfits the data, but is probably due to the small sample size, 32 or 33, of the cross-validation subsets.

Figure 4.3: Plot of *cp* values against cross-validation error for model with cluster weights



The subsets used for cross validation need to be of similar size but also to be mutually exclusive. With a sample size of 326, creating 10 cross validation subsets will result in 4 of these subsamples being of size 32 and the rest of size 33. This issue of *xerror* being greater than 1.0 is also associated with a small sample size, as seen in the prostrate cancer example given in Section 6.1.2 of Therneau & Atkinson (2000).

We conclude from the analysis above that the complexity parameter is unsuitable as a method of tree pruning for the replicate subsamples. Instead, minimum split and tree depth only were considered for optimising weighted classification tree models from the replicate subsamples. The trees built from replicate samples were then used to provide between-replicate variability, the second term in Equation (4.1), the inverse sampling formula for variance.

4.6 Estimating between replicate variance

The suitability of minimum split specification and tree depth as methods for controlling tree complexity and providing stable predictions was investigated. Setting the complexity parameter to zero eliminates its influence in determining optimal tree size, so that only minimum split and tree depth are considered as stopping criteria. A minimum split criterion of $minsplit = n$ stops further splitting at a node when the node comprises less than n observations. The value of n could be the same as used in the full tree or set to reflect the size of the replicate subsamples proportional to the full survey dataset, one twelfth. The default value in *rpart* is $minsplit = 20$, the value used in the weighted classification tree model (Section 3.2.4). So, a one twelfth proportion is about 2, but this was considered too small to be practical, so the smallest value of minimum split used was 3. In the modelling process to estimate between replicate variability, minimum split was set as the only stopping rule applied to the trees, and the tree depth left at the default of 30. The weighted classification tree generated from Replicate 1 using a minsplit of 3 is shown in Figure 4.4. The structure of the weights used to build this tree were described in Section 4.4.

In comparing similarity between the weighted classification tree built from Replicate 1 data in Figure 4.4 and that constructed from the full survey dataset, Figure 3.5, we note first that the depth of the full tree is eight, while the tree built from Replicate 1 data is slightly larger with depth of ten. However, making a comparison on this basis is not particularly useful since, although the minimum split values are proportional, 20 for the full tree as against 3 for the replicate tree, the *cp* values are very different, $cp = 0.005$ for the full tree and $cp = 0$ for the replicate tree. A more interesting observation is that *skids6*, the number of children in the household aged six and under, and *skids6w*, the proportion across the ward of children in the household aged six and under, provide the first three splits in the tree built from Replicate 1 data.

Figure 3.7 displays details of the first six competing splits for the root node in the weighted classification tree build from the full Nepal dataset (Figure 3.5). Comparing this information with the tree diagram of the model built from Replicate 1, Figure 4.4, it can be seen that both of these predictors, *skids6* and *skids6w*, are competing splits for *tvw*, the primary splitting variable in the weighted classification tree model built from the full Nepal survey dataset. New splitting rules, such as the variable *bratev* representing the birth rate for adult women, have appeared in the tree built from the Replicate 1 data. Two additional primary splitters are *avmwhv*, indicating the average number of months worked for households across the ward, and *avanuwsv*, the proportion of adults not working due to disability. The determinants of poverty for the households selected in Replicate 1 are concerned mainly with the presence of young children and the working status of adults in the household.

A weighted classification tree model, using the complexity control parameter minimum split assigned the value 3 and $cp = 0$, was built for all the twelve replicates. Each replicate tree model was applied to the census data from Ilaka 1 in a particular district of Nepal. A small area estimate, $\hat{\theta}_r^*$, was generated for the r^{th} replicate, and these

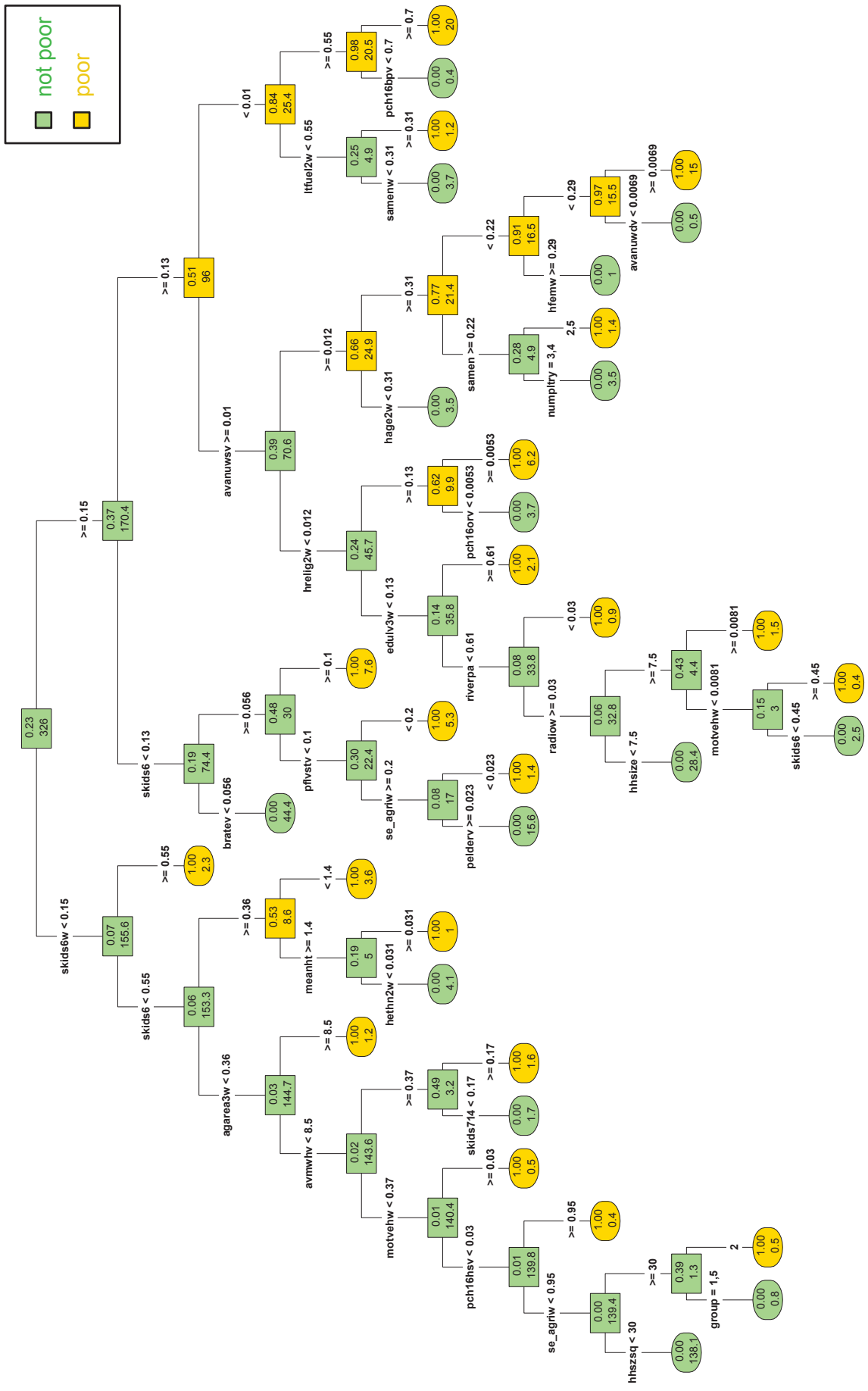


Figure 4.4: Tree diagram for weighted classification model using only data from Replicate 1

$\hat{\theta}_r^*$ values used to provide an estimate of between-replicate variance, the second term in Equation (4.1), the estimation of variance under inverse sampling.

Table 4.1 lists the values of $\hat{\theta}_r^*$ for each of the twelve replicate subsamples. The mean of these estimates is 0.575, and standard deviation of 0.149, giving a between replicate variance estimate of 0.022. These statistics indicate quite a large variability in replicate estimates of poverty incidence. The other component of variability in the inverse sampling formula for variance is the first term in Equation (4.1) which measures the average of the $\hat{V}_r^*$'s, the estimates of within replicate variability. An approach to computing the $\hat{V}_r^*$ terms is discussed in the next section.

Table 4.1: Estimates of poverty incidence for Ilaka 1 using replicate subsamples

Replicate	$\hat{\theta}_r^*$	Replicate	$\hat{\theta}_r^*$	Replicate	$\hat{\theta}_r^*$
1	0.712	5	0.669	9	0.649
2	0.812	6	0.622	10	0.697
3	0.465	7	0.646	11	0.538
4	0.304	8	0.364	12	0.421

4.7 Jackknife variance estimation of within replicate variability

The within replicate variance component, the first term in Equation (4.1), denotes the average of the within replicate variability for each replicate, the $\hat{V}_r^*$'s, and is required to allow for the conditional dependence of the replicate estimates on the original sample survey data. Since a tree model provides a single estimate for each replicate dataset, computation of the $\hat{V}_r^*$'s requires an additional resampling scheme to be applied to each replicate. The variance estimation method chosen for the task was jackknife resampling for simple random sample data (Section 2.2.3.3), with variance estimator of the form,

$$Var_{JK}(\hat{\theta}) \;=\; \frac{n-1}{n} \sum_{j=1}^{g} \Big(\hat{\theta}_{(j)} \;-\; \hat{\theta}_{(\cdot)} \Big)^2 , \tag{4.4}$$

where $\hat{\theta}_{(j)}$ is the estimate of θ obtained from the j^{th} jackknife sample, and $\hat{\theta}_{(\cdot)}$ the average of the $\hat{\theta}_{(j)}$'s. The use of Equation (4.4) for within replicate variance estimation is appropriate, since the purpose of the inverse sampling was to create samples with a structure akin to simple random sampling (Hinkins et al. 1997). Delete-a-group jackknife was initially chosen for the variance analysis on replicate samples, with a group size of two, providing 163 jackknife samples. Since the factors of 326, the replicate size, are 1, 2, 163 and 326, a group size of 2 was the only practical option. Employing a delete-1 jackknife method with tree models may be problematic, since the delete-1 jackknife is inconsistent for non-smooth statistics (Miller 1974). This inconsistency can be mitigated by the delete-group jackknife (Efron & Tibshirani 1993). In addition, working with 326 jackknife subsamples was considered to be cumbersome.

A tree model was built on each of the 163 jackknife samples of the data in Replicate 1. Each tree model was then applied to the census data in Ilaka 1 of the chosen district in Nepal, to generate 163 jackknife predictions, which provide a measure of within replicate variability based on the jackknife estimates. The model weightings used were the w_{ij}^* defined in Equation (4.2). No further adjustment was made to the household weights, and consequently the sum of weights for a jackknife sample only approximated the sample size of 324. Tree complexity was controlled by setting $cp = 0$ and a minimum split value of three.

Figure 4.5 displays a frequency table for the 163 jackknife predictions of poverty incidence for the model employing minimum split of 3 as the method of tree optimisation. The mean of these 163 estimates was 0.809, which is about 14% larger than the value computed from the tree built from all the data in Replicate 1 (Table 4.1). The table in Figure 4.5 indicates one main repeated small area estimate value for poverty incidence in Ilaka 1, a mode of 0.8305 generated by 137 of the 163 jackknife tree models. The jackknife estimate of variance (Equation (4.4)) is a scaled version of s^2, the sample variance of the $\hat{\theta}_{(j)}$, so estimates which are outliers have a big effect on the estimated variance.

Figure 4.5: Table of estimates of poverty incidence in Ilaka1 using 163 jackknife subsamples of Replicate 1

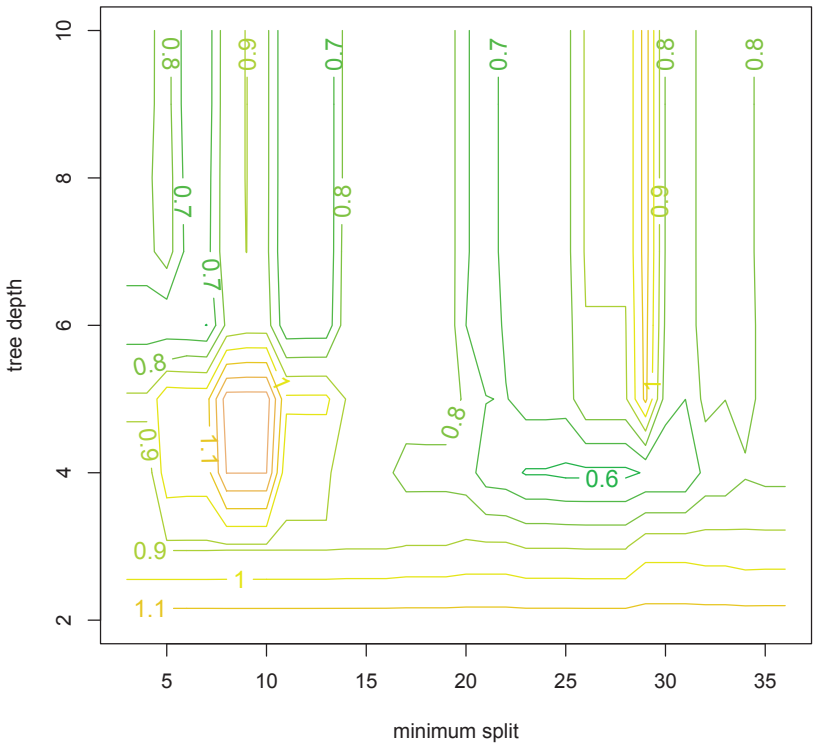
```
P0

0.4742 0.4751 0.5074 0.5417 0.5677 0.5923 0.6056 0.6178 0.6632 0.6955
     1      1      1      1      1      1      1      1      1      2

0.7069 0.7125 0.7153 0.7768 0.7842 0.7915 0.8297 0.8305 0.8381 0.8504
     1      1      1      6      1      1      1    137      2      1
```

Inconsistency in the jackknife method when the statistic is not smooth is evident here, even for a delete-group jackknife, although a group size of 2 may not be large enough to mitigate instability in the jackknife method. Since each household is classified as either poor or not poor, the hard classification tree prediction is a discrete statistic, and therefore not smooth. The inconsistency of this statistic is demonstrated in Figure 4.5 which displays values ranging from 0.47 to 0.85. The standard deviation of these 163 estimates is 0.0666 which, by Equation (4.4), equates to a jackknife standard error of 0.845. This represents a very large value for standard error, since the quantity being estimated is a proportion, which should have a maximum value of one. The main cause of the instability appears to be one main repeated estimate value, and considerable variability in the remaining estimates, resulting in an unacceptably large variance in the estimates. Minimum split alone as a stopping rule seems inadequate, so tree depth should also be considered as a means of optimising the tree. The analysis was extended to investigate the effect of a range of values for minimum split and tree depth on the stability of poverty estimates.

4.8 Effect of minimum split and tree depth on tree stability

The next stage of the investigation examined how different combinations of minimum split and tree depth affected the variability in small area estimates of poverty incidence using jackknife subsamples. Different model options were explored, with the objective of finding a combination of values of minimum split and tree depth which would provide a stable tree model, one exhibiting a reasonable variability across the jackknife subsamples. A succession of models were run, for minimum split values of 3 to 36, and tree depth from 2 to 10. A maximum tree depth of 10 was selected, since this is the greatest depth of the tree model built using all the data in Replicate 1 (see Figure 4.4). A jackknife estimate of within replicate variance was generated for each combination of minimum split and tree depth. A contour plot of the jackknife estimates of variability, expressed as standard deviations, for each of the varying values of minimum split and tree depth is given in Figure 4.6.

Figure 4.6: Contour plot of jackknife standard deviation values for varying minimum split and tree depth

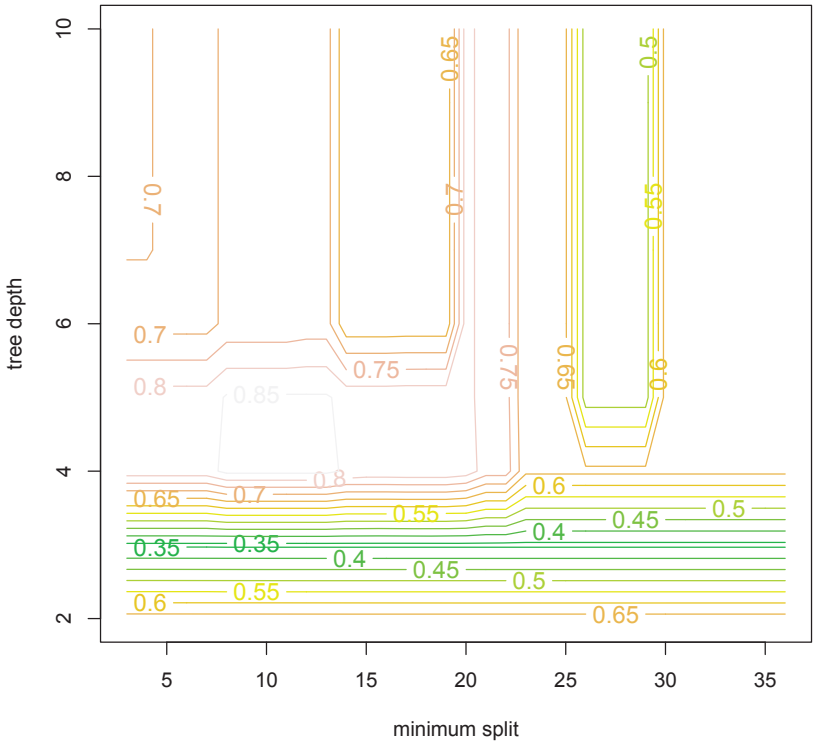


The contour plot in Figure 4.6 indicates a range of standard deviations between 0.6 and 1.2. Since the quantity of interest in the analysis is a proportion, this range represents variability which is exceedingly large, and indicates a considerable amount of instability in the jackknife hard tree predictions, for all combinations of minimum split and tree depth. Instability in the estimates tends to decrease, as expected, for larger trees, which have greater depth. For tree depth of 2 and 3, the level of instability is consistent

across all sizes of the tree. A minimum occurs in the contour plot for a tree depth of 4 and minimum split values approximately 22 to 30.

Figure 4.7 displays a contour plot of the mode of the estimates for varying minimum split and tree depth. The first point to note is the wide range of these estimates, from approximately 0.35 to 0.8. The full tree model for Replicate 1, built using all 326 observations in Replicate 1, provides a small area estimate for poverty incidence in Ilaka 1 of 0.831. The contour plot in Figure 4.7 displays some unusual patterns of instability. For tree depths of 2, 3 and 4, the poverty estimate values remains fairly constant as minimum split levels increase. However, estimate values decrease as tree depth increases from 2 to 3, and then increase as the tree is extended to 4 levels. The notable feature of Figures 4.6 and 4.7 is regions of very steep change, i.e. "flat valleys and steep cliffs", which illuminates this interesting issue of instability in the hard small area estimates using jackknife subsamples.

Figure 4.7: Contour plot of jackknife mode estimate values for varying minimum split and tree depth



Thus, finding a combination of minimum split and tree depth which can provide a variance model, using replicates and jackknife resampling within replicates, to generate a reasonable measure of standard error of predictions for poverty incidence has not been achieved, because the estimates of standard error are too large, some even exceeding 1.0. A single model was then examined in greater depth in order to understand the reasons for the tree instability and such divergent estimates. The weighted tree model built using replicate weights, as per Equation 4.2, and model options of $cp = 0$, minimum split of 3

and depth of 4, was chosen because it highlights some of the most problematic aspects of using replicates for variance estimation, namely a great majority of estimates, 137 of the 163, having identical value of 0.8305, and a wide range of predictions, between 0.474 and 0.850.

More in depth analysis involved comparing tree diagrams of models using minsplit of 3 and depth of 4, firstly the model based on all the data in Replicate 1, then the model built from data in jackknife subsample #25. This particular jackknife sample was chosen because it exhibited extreme jackknife estimates of poverty incidence. It was found that an extreme prediction was usually associated with a significant change in tree structure. The first point of note is the influence of competing splits on these changes in tree structure.

4.9 Competing splits

Tree instability occurs when a slight change in input data favours a different splitting rule (Breiman 1996*b*), and consequently different tree structure and prediction. Li & Belford (2002) propounded an "Instability theorem" which proved that the existence of almost equally good splitting criteria for a node, i.e. providing very similar decreases in node impurity, could result in the splitting rule chosen for that node being sensitive to small changes in the training data. These "almost equally good" splitting criteria are known as competing splits (Section 3.2.5). An example of how competing splits contend for the position of primary split when the input data is slightly perturbed is outlined in this section.

The algorithm used to build the tree includes a splitting function which determines which predictor variable provides the best discrimination of the observations at each node in the tree. A splitting rule separates the observations into two distinct groups based upon a single cut-off value for a continuous predictor variable, and combinations of the factor levels for a categorical predictor. The splitting function scrolls through each possible splitting rule to assess the best partition of the data. Gini splitting criterion, the most commonly used measure of node impurity (Section 2.4.2.2) was applied to determine the best splitting rules for the tree. For binary partitioning, which for poverty status involves division into two classes of "poor" or "notpoor", the Gini index takes the form,

$$\text{Gini index} = 2\, p_{iA}\ [1\ -\ p_{iA}]\ , \tag{4.5}$$

where p_{iA} denotes the posterior probability that the i^{th} observation in node A is assigned the class of interest, which in the context of poverty incidence equates to a household being categorised as "poor". When the observed class frequencies in the sample are used as the prior probabilities, then (Therneau & Atkinson 2013) the posterior probability has the form,

$$p_{iA}\ =\ \frac{n_{iA}}{n_A}\ ,$$

for n_A the total number of observations in node A, and n_{iA} the number of observations

classed as being "poor" in node A. For a weighted tree model, the terms n_A and n_{iA} incorporate model weights for each observation and are designated as "class counts", as discussed later. When prior probabilities are assigned on the basis of observed class frequencies, Equation 4.5 equates to

$$\text{Gini index} \;=\; 2\,\frac{n_{iA}}{n_A}\left(1-\frac{n_{iA}}{n_A}\right) \;=\; 2\,\frac{n_{iA}}{n_A}\left(\frac{n_A-n_{iA}}{n_A}\right).$$

Since n_{iA} denotes the number of poor in node A, then $n_A - n_{iA}$ is the number of households in node A classed as not being poor. In devising notation to show incorporation of weights into the Gini splitting criterion, we adapt the terminology used in the *rpart* output where the term "class counts" refers to the sum of weights in a specified class in a node. For example, the class count for poor households in a node is the total weight for all households in the node which have predesignation "poor". Suppose c_{iA} represents the sum of weights of all observations in node A classed as "poor", and c_A the total weights in node A. Also, let c_L denote the number of observations in the left split from node A, and c_R represent number of observations in the right split from node A. Then, c_{iL} and c_{iR} indicate the sum of weights of observations classed as "poor" which are sent into the left hand split and right hand split, respectively. The splitting rule chosen for a particular node is that which has the greatest information gain (IG), where

$$\text{IG} \;=\; 2\,\frac{c_{iA}}{c_A}\left(\frac{c_A-c_{iA}}{c_A}\right) - 2\left[\frac{c_L}{c_A}\frac{c_{iL}}{c_L}\left(\frac{c_L-c_{iL}}{c_L}\right) + \frac{c_R}{c_A}\frac{c_{iR}}{c_R}\left(\frac{c_R-c_{iR}}{c_R}\right)\right] \tag{4.6}$$

The *rpart* summary output for a tree model (see Figure 3.8) displays a statistic labelled *improve*, which quantifies the reduction in impurity resulting from a particular split as the information gain for that split multiplied by the number of observation in the node being split, such that $improve = c_A \times \text{IG}$.

Figures 4.8 and 4.9 compare the tree diagrams of the model based on the full Replicate 1 dataset to that built from a delete-2 jackknife subsample #25, which omitted households with ID numbers 85 and 281 from the replicate data. The root node split for the full replicate tree is determined by the variable *skids6w*, the proportion of children 6 years and under in the household, averaged over the ward in which the household is situated. The splitting variable at the root of the jackknife tree is *edulv4w*, the proportion of people aged fifteen and over who have had between eight and ten years of schooling. Thus, the tree structure has changed considerably with the omission of only two households. The first left hand split involves the same variable, *skids6*, for both the full replicate tree and the jackknife tree, but with different splitting rules, $skids6 < 0.55$ for the full tree from Replicate 1 and $skids6 < 0.13$ for the jackknife tree. Otherwise, the two trees have very different structures.

A portion of the summary output for the weighted classification tree model built using all the Replicate 1 data is displayed in Figure 4.10. The weights used in the modelling are those defined in Equation 4.2. Note that the number of observations in the node

Figure 4.8: Tree diagram for model using all data from Replicate 1

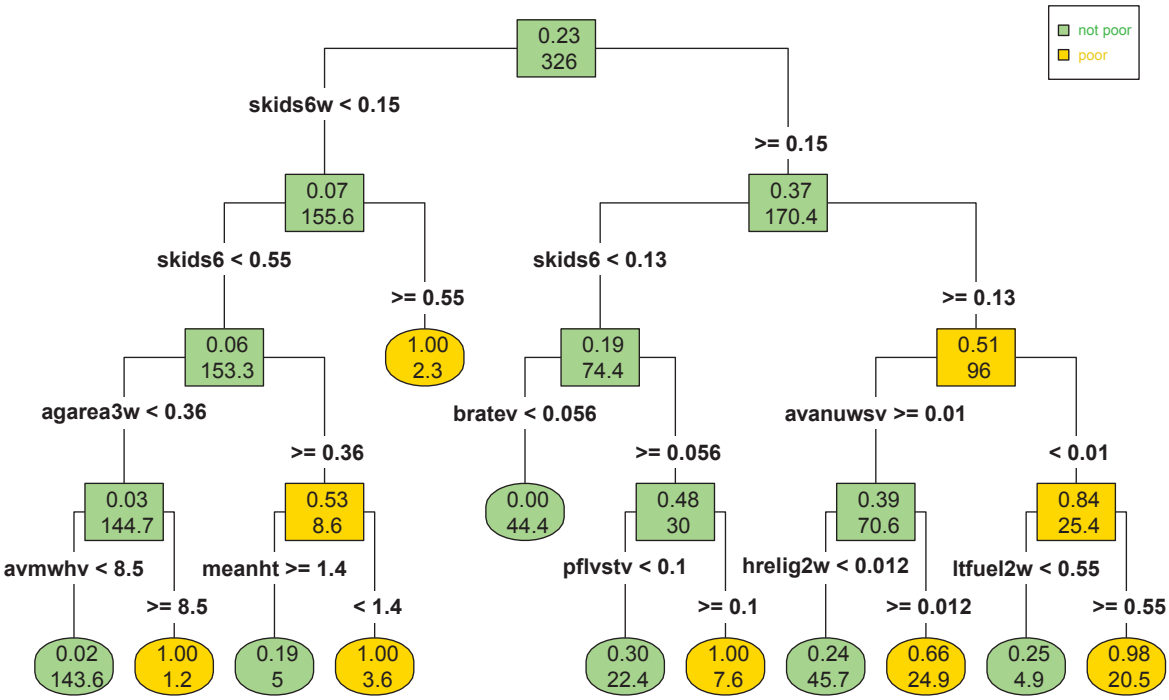


Figure 4.9: Tree diagram for model using data from jackknife subsample #25 of Replicate 1

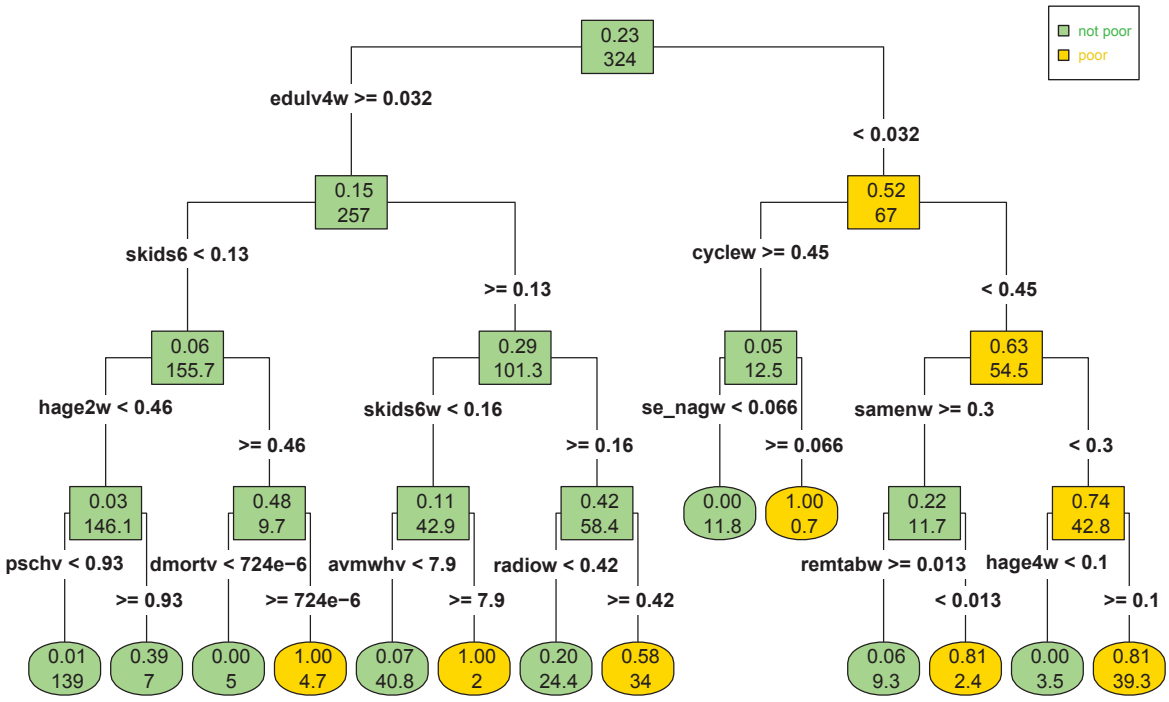


Figure 4.10: Summary of Node 1 for model on full replicate sample, cp=0, split=3, depth=4

```
Node number 1: 326 observations,    complexity param=0.07697987
  predicted class=notpoor  expected loss=0.2273548  P(node) =1
    class counts: 251.882 74.1177
   probabilities: 0.773 0.227
  left son=2 (153 obs) right son=3 (173 obs)
  Primary splits:
      skids6w  < 0.1539799  to the left,  improve=14.57771, (0 missing)
      skids6   < 0.1339285  to the left,  improve=13.86400, (0 missing)
      edulv4w  < 0.032498   to the right, improve=13.56240, (0 missing)
      bratev   < 0.0492612  to the left,  improve=12.57349, (0 missing)
      toilet3w < 0.4083555  to the left,  improve=11.70362, (0 missing)
      popdens  < 496.5419   to the right, improve=11.43255, (0 missing)
```

summary denotes the total number of households in the node, which is used, by *minsplit*, to determine whether a further split can occur, but not for assessing node purity. The class counts, 251.882 and 74.1177, represent the sum of model weights for households which are not poor and poor respectively. The *improve* values for the chosen splitting rule and the first five competing splits at the root node are also shown in the summary output, Figure 4.10. We note that the variable *skids6w* is the splitting criterion chosen at the root node for the tree built using all data in Replicate 1, and *edulv4w* is the second competing split for this model. Figure 4.11 displays the the summary for Node 1 in the tree built from jackknife sample #25. The primary splitting variable is now *edulv4w*, while *skids6w* is relegated to a position as the first competing split.

To elucidate the underlying cause of instability in tree structure when the jackknife sample #25 was used build the model instead of the full replicate dataset, the values of the relevant variables for the two households omitted from the jackknife sample were examined. The variables which determine the change of primary splitting rule at the root node are household poverty status, model weight for each household, and the values of *skids6w* and *edulv4w* for the omitted observations, having ID's of 85 and 281. This information is displayed in Table 4.2.

Figure 4.11: Summary of Node 1 for model on JK #25 subsample, cp=0, split=3, depth=4

```
Node number 1: 324 observations,    complexity param=0.0968018
  predicted class=notpoor  expected loss=0.2273796  P(node) =1
    class counts: 249.524 73.434
   probabilities: 0.773 0.227
  left son=2 (254 obs) right son=3 (70 obs)
  Primary splits:
      edulv4w < 0.032498   to the right, improve=14.73419, (0 missing)
      skids6w < 0.1539799  to the left,  improve=14.70889, (0 missing)
      skids6  < 0.1339285  to the left,  improve=13.96244, (0 missing)
      bratev  < 0.0492612  to the left,  improve=12.77508, (0 missing)
      popdens < 496.5419   to the right, improve=12.10873, (0 missing)
```

Table 4.2: Predictor values for households omitted from jackknife sample #25

Rep 1 ID	**Poverty**	**Weight**	**skids6w**	**edulv4w**
85	not poor	2.3585	0.2089	0.0268
281	poor	0.6837	0.1875	0.1428

Based upon the information in Table 4.2, the pathways taken by households 85 and 281 for both possible splitting variables *skids6w* and *edulv4w* can be determined. These pathways are displayed in Table 4.3. The root node is designated as Node 1, the first left hand split as Node 2, and the first right hand partition as Node 3. An abbreviated model summary for both models, using all the data in Replicate 1 and then only jackknife #25 data, i.e. with observations 85 and 281 omitted, in provided in Appendix B. These reduced summaries include Nodes 1, 2 and 3, the root node and first left and right splits, of both models.

Table 4.3: Predictor values for households omitted from jackknife sample #25

Predictor	**Splitting value**	**Pathway for 85**	**Pathway for 281**
skids6w	< 0.154	right into Node 3	right into Node 3
edulv4w	≥ 0.032	right into Node 3	left into Node 2

When the jackknife subsample #25 was used to build the tree, the datapoints representing households 85 and 281 were omitted from the dataset. Omission of these datapoints changed the class counts for each node, which affected the information gain and improve values for each split. Table 4.4 demonstrates how the improve function for the original splitting variable *skids6w* changed at the root node when households 85 and 281 were omitted from the dataset, to create jackknife subsample #25. Changes are emphasised with bold font, and *improve* values were calculated using Equation (4.6). The expressions c_{iL}, $c_L - c_{iL}$, c_{iR} and $c_R - c_{iR}$ denotes class counts for classification as "poor" in the left split, "notpoor' in the left split, "poor" in the right split and "notpoor" in the right split, respectively. It is sufficient to display only the class counts for poor and non poor at the left and right splits of the node being partitioned. The values of the other terms in Equation (4.6) can be calculated from these four quantities. Computation of the *improve* values was achieved using a hand written *R* function, *Improve*, as shown in Appendix C.1.

Table 4.4: Class counts and *improve* functions using *skids6w* as first split

Dataset	c_{iL}	$c_L - c_{iL}$	c_{iR}	$c_R - c_{iR}$	**Improve**
Replicate 1	11.033	144.588	63.084	107.294	14.577
JK 25	11.033	144.588	**62.401**	**104.936**	14.709

Table 4.3 indicates that, when *skids6w* is the splitting variable at the root node for the model based on all Replicate 1 data, the path which household 85 takes from the root node is right into Node 3. Thus, when this household, which carries label of

"notpoor", is omitted from the dataset, and the model is built from jackknife sample #25, the class count for "notpoor" in Node 3, the first right split, decreases by 2.3585, the weight associated with household 85 (Table 4.2). This equates to a reduction in class counts of "notpoor" in the first right split from 107.294 to 104.936 (Table 4.4). Household 281, which has poverty status of "poor", also travels right into Node 3, so its omission from the dataset decreased the class count for "poor" in Node 3 by 0.6837, the model weight for household 281, a reduction from 63.084 to 62.401. As a consequence, the improve value for *skids6w* as the first splitting variable in the tree built from the jackknife sample #25 is 14.709, as compared with an improve value of 14.577 for its function as root node splitting criterion for the tree built from all Replicate 1 data (Table 4.4).

Using *edulv4w* as the initial splitting variable results in a different pattern of change in class counts from the full Replicate 1 tree to the jackknife #25 tree. From Table 4.5 we see that when *edulv4w* is the splitting variable at the root node for the full Replicate 1 tree, the household 85 is still sent right into Node 3, but household 281 is sent left into Node 2. Household 85 has class label of "notpoor", so its inclusion in the dataset used to build the full tree results in an increase in class count of "notpoor" in the first right split from 31.836 to 34.195, the difference being the weight of household 85, which is 2.3585. Similarly, household 281, with class label of "poor", takes the left hand branch from the root node into Node 2. Thus its inclusion in the full dataset increases the class count of "poor" in the first left split of the tree from 38.497 to 39.181, a difference of 0.6837, being the model weight of household 281. The *improve* value for *edulv4w* as root node splitter in the full tree, built from all the data in Replicate 1, is 13.562, compared to an *improve* value of 14.734 as primary split for the jackknife #25 tree.

Table 4.5: Class counts and *improve* functions using *edulv4w* as first split

Dataset	c_{iL}	$c_L - c_{iL}$	c_{iR}	$c_R - c_{iR}$	**Improve**
Replicate 1	**39.181**	217.688	34.937	**34.195**	13.562
JK 25	38.497	217.688	34.937	31.836	14.734

The omission of two households to create the jackknife sample has adjusted the *improve* values sufficiently so that the second competing split from the tree based on all the Replicate 1 data, *edulv4w*, now provides the best discrimination between observations in the reduced tree, built from the jackknife sample which omits households 85 and 281. The calculated *improve* values displayed in Tables 4.4 and 4.5 are confirmed in the summary outputs for the Replicate 1 and jackknife models, as shown in Figures 4.11 and 4.10. In the jackknife tree, the splitting variable *edulv4w* now has a slightly higher *improve* value than *skids6w*. A different initial splitting rule has resulted in a completely different tree structure, and produced very different predictions. The normalised weight of household 85 is 2.3585, indicating that this household represents proportionally two and a half times more individuals in the population than the "average" household in the Replicate 1 dataset. This detail may explain why *edulv4w* has such a marked effect on the tree building process when households 85 and 281 are removed. It jumps two places in priority ranking as the

primary splitting variable at the root node, from second competing split for the tree based on all the data in Replicate 1 to primary split for the model built from the jackknife reduced dataset. It is interesting to note that the splitting values, the cut-off points, for *skids6w* and *edulv4w* are the same for both models, the tree built from all Replicate 1 data, and the jackknife tree which omits households 85 and 281.

Other approaches to modelling were tried. The over-representation of census means as the most important splitting variables (see Section 3.2.6) suggests that these predictors might contribute to the problems of lack of continuity in the data. A model was rerun using only the 20 household level variables, but this produced no significant reduction in estimate instability. In addition, predictions were generated for another ilaka in the selected district of Nepal, chosen because its estimate of poverty incidence was around 0.23, the estimated level of poverty for all of Nepal. Modelling was also carried out using jackknifing on different replicates, but the instability problem was not eliminated, and in one case actually increased.

4.10 Conclusions

Inverse sampling seemed an attractive option for variance estimation using replicates samples since it takes account of the clustering in the survey data. However, applying the inverse sampling variance estimation procedure on the replicate subsamples, including using jackknifing to generate an estimate of within replicate variability, has not provided a suitable measure of the standard error of prediction for poverty incidence in Nepal. The problems highlighted in this chapter relate to the use of hard tree estimation and instability of the tree structure when observations were omitted to construct jackknife samples for variance estimation.

Classification trees are inherently unstable (Last et al. 2002). The usual approach to this problem is to generate multiple trees and take an average (Breiman 1996*a*). However, the aggregation method discussed in this chapter, jackknife resampling with hard tree estimation, has not corrected the difficulty of tree instability. The jackknife technique is inconsistent for estimators which are not smooth, such as the classification tree. Using the hard type of tree estimate in modelling exacerbates this problem because the hard estimator is discrete, since it predicts households to be either poor or not poor. Resampling can change the tree structure to such an extent that the deletion of two households might result in the reclassification of a large proportion of the remaining households from poor to not poor, or vice versa. Utilising the inverse sampling variance estimation method to create replicate subsamples was also problematic because of the small sizes of the replicates, much smaller than the subsamples produced by jackknife or bootstrap resampling.

The soft tree estimate, the probability of being poor, would be expected to perform better than the hard estimate type, being a more smooth estimator. A bootstrap resampling approach might ameliorate the problem of inconsistency since bootstrapping would be less affected by the non smooth nature of the tree model. The next chapter

discusses a designed experiment set up to compare the effectiveness of hard versus soft types of tree estimation and jackknife versus bootstrap resampling methods in providing reasonable standard errors of prediction of poverty incidence. The designed experiment also included sample size as a factor, to investigate the minimum sample size for estimate stability.

Chapter 5

A study in stability

5.1 Introduction

Generating valid standard errors of prediction for poverty incidence using a classification tree model has proved to be a challenge, due to the inherent instability of the decision tree algorithm. Tree instability occurs when a small change in a dataset produces a very different tree structure and prediction Breiman et al. (1984). When survey data comprises more than one component of complex survey design, some type of re-sampling based variance estimation procedure is needed to estimate standard errors of prediction for the quantity of interest. In Chapter 4, the problem of tree instability became apparent when the resampling method employed for variance estimation, the jackknife, produced unrealistically large standard errors .

Stable classification algorithms are important since they ensure repeatability of results (Turney 1995). The issue of instability in decision tree models is a well known and long-term problem (Kotsiantis 2013). For example, from their studies of regression trees Toth & Eltinge (2011) observed instability of splits and the resulting aggregates. Instability in recursive partitioning arises when different classification rules are generated by slightly different training samples (Dwyer & Holte 2007). This phenomenon is described as being counter-intuitive by Li & Belford (2002), who provide fundamental theorems for instability in decision tree algorithms. However, tree instability can be explained very simply, it is due to the inherent nature of the tree building algorithm which chooses the best split for each node. Section 4.7 illustrates a situation in which changes in tree structure occurred when a slightly different dataset favoured a competing split in preference to the original splitting variable.

Tree instability is evident with small datasets but is also an issue when resampling techniques are applied to decision trees (Pérez et al. 2004). Several types of ensemble methods have been devised to address the problem of tree instability, such as bagging (Breiman 1996*a*), boosting (Freund & Schapire 1997), random subspaces (Ho 1998) and Random Forests (Breiman 2001). However, these approaches focus upon achieving stable predictions, whereas the scope of the thesis is valid standard errors of prediction. In this

chapter we describe a Monte Carlo simulation study to investigate the factors affecting tree stability under resampling, with the objective of devising a method which will provide valid standard errors from a classification tree model.

5.2 Monte Carlo simulations

A Monte Carlo simulation is a process in which artificial datasets are generated to replicate the properties of a real life situation. This technique provides a means by which researchers can choose suitable analytical methods when the underlying model assumptions do not hold (Serlin 2000), for example if the normality requirement is not met. As a methodology to evaluate the properties of statistical procedures, Monte Carlo studies have a long history, including William Sealy Gosset's introduction to his Student-t-test (Student 1908).

The purpose of a Monte Carlo simulation is to create a simulated population with characteristics which mimic those of the real life population. The rationale for applying a simulation is that the performance of the simulated distribution would be close enough to that of the true distribution for valid inferences to be made on the actual population. In addition, the expectation is that a statistical procedure with properties which are robust under a range of ideal distributions would perform well in practice. Monte Carlo simulations have been used for a wide range of analyses, including a study by Stangenhaus & Narula (1991) to investigate the smallest sample size required to build robust confidence intervals for parameters in L_1 regression. The simulation study outlined in this chapter began as an exercise to determine the smallest sample size needed for stable tree estimates under jackknife resampling, but expanded into a designed experiment which examined how estimate stability was affected by the resampling method, type of tree estimate, size of survey used to construct the tree, and tree complexity as specified by the minimum split criterion.

A major use of simulation studies is the examination of the properties of a particular statistical model or procedure. Many artificial datasets are generated to replicate the known properties of a real world situation (Ólafsdóttir & Mudelsee 2014). For each dataset generated, a confidence interval is constructed and the actual, or empirical, coverage is compared with nominal coverage (Barton et al. 2014, Paul & Zhang 2014). Actual coverage is defined to be the proportion of simulations which generate a confidence interval containing the true parameter value. Until computers became available to facilitate the generation of random numbers, the number of iterations in Monte Carlo studies was around 1000 (Serlin 2000). With the development of sophisticated computers very complex simulations have become feasible, but are often very computationally expensive (Jones & Waller 2013, Ali et al. 2014). The Monte Carlo studies described in this chapter involved 100 or 1000 iterations, since a larger study was found to require too much computation time. The purpose of the simulation exercise was to develop a method which provided stable predictions of poverty incidence utilising classification tree models.

5.3 Source of instability

We recall how the instability problem arose. Since the NLSS survey data contained elements of complex survey design, a variance estimation procedure was required to obtain an estimate of standard error for predictions of poverty incidence. The structure of the NLSS data, 12 households selected from each PSU chosen for the survey, suggested constructing replicate subsamples by random selection of one household from each PSU (Section 4.2.1) . Since the replicates thus formed are not independent, the variance estimation procedure is akin to inverse sampling. However, the formula for variance under inverse sampling (Equation 4.1) includes an estimate of variance for each replicate, $\hat{V}_j^*$. Each replicate sample provided only a single tree, and thus only a single prediction of poverty across a given region. Some type of double-sampling was required to be applied to the replicates for estimating variability within each replicate. The initial approach was to use jackknife subsampling of the replicates and compute hard tree estimates of poverty.

However, poverty estimates obtained through this method were extremely variable, sometimes resulting in standard errors of the proportion of poverty greater than one. When hard tree estimates were computed with delete-1 or delete-2 jackknife resampling, most of the jackknife samples gave exactly the same tree structure and identical estimates, but some subsamples produced wildly varying estimates of poverty, as a consequence of very different tree structures. The initial approach to solving this problem was a simulation study to investigate the minimum survey size required to produce sufficiently stable estimates.

5.4 Outline of simulation study

The purpose of the simulation exercise was to examine which aspects of the variance estimation and classification tree processes would provide sensible standard errors from a classification tree. Initial investigations focused on the minimum size of the survey dataset needed for tree stability using jackknife resampling and hard estimates, since the results from Chapter 4 suggest that a sample size of about 300 is too small to produce stable estimates. Thus, the first consideration was to find the optimal size of survey dataset to produce trees which were stable enough when resampling was employed. Simulated survey datasets of increasingly larger sizes were generated to determine at which point instability was no longer a problem.

Instability in the tree algorithm is unrelated to complex survey design, so for the initial analysis the simulated data was completely independent, having no clustering or stratification effects. The Monte Carlo study focused on unconditional variance estimation, which involved refitting the tree at each iteration, so that tree structure was not fixed but allowed to vary with different survey samples. In contrast, the ELL methodology conditions upon the assumption that the regression model and its parameter estimates are correct. A single model is fitted, and standard errors of prediction are generated using

variability in the parameter estimates, as well as cluster level and household level variability. So, variance estimation using tree based models contains an extra unconditioning aspect not included in the ELL technique. In the simulation process, survey and census datasets were simulated, and then used to test how changing the survey size affected the stability of jackknife hard estimates. Since the study investigated predictions across a single small area, it was not necessary to simulate the full census data, which would have required much greater computational effort. The simulated survey and census datasets were generated from the same model based upon NLSS data, rather than taking a random sample from the census. Construction of the simulated datasets is described in the next section.

5.5 Simulating the datasets

The simulation was conducted using the same method of introducing auxiliary information to improve estimate precision as applied in the ELL linear regression modelling. A survey dataset was simulated and used to build a classification tree, then predictions of poverty incidence were generated by passing simulated census data through the tree. These predictions were then amalgamated across small domains of interest to provide small area estimates of poverty. For ease of computation, the set of common variables for survey and census was restricted to only those predictors used in the final model of poverty incidence in Nepal. The simulated survey and census datasets were constructed from the same linear model,

$$Y_i \;=\; \boldsymbol{X}_i\boldsymbol{\beta} \;+\; \epsilon_i \;, \tag{5.1}$$

where Y_i denotes log expenditure, $\boldsymbol{X}_i$ the vector of predictor values and $\boldsymbol{\beta}$ the vector of model coefficients. This structure was achieved by generating matrices of standard normal random variables which were then modified to have the same mean and covariance structure as the original NLSS survey data, with the option of changing the number of observations, n. This process was carried out according to the following equation,

$$\mathbf{Y}_{sim} \quad = \quad \mathbf{Z}\,\boldsymbol{\Sigma}^{\frac{1}{2}} \;+\; \boldsymbol{\mu}\,, \tag{5.2}$$

where, $\mathbf{Y}_{sim}$ represents the simulated survey or census dataset, $\mathbf{Z}$ is the matrix of simulated standard normal random variables, $\boldsymbol{\Sigma}^{\frac{1}{2}}$ denotes a square root of the covariance matrix estimated from the Nepal survey dataset, and $\boldsymbol{\mu}$ is a matrix of constants. Each column of $\boldsymbol{\mu}$ is a vector of constant values in which each element is equal to the mean value for the variable in the corresponding column of the Nepal survey dataset.

Modelling was based upon the simplest situation, independently and identically normally distributed data. Thus the matrix $\mathbf{Z}$ comprised twenty six columns, each of which was a standardised normal random vector of a specified length, n. These twenty six columns correspond to the twenty six variables utilised for the ELL model of poverty incidence in Nepal, the response log expenditure, and twenty five predictors. The log

transformation was applied because the original expenditure data is very right skewed. The matrix $\mathbf{\Sigma}^{\frac{1}{2}}$ is derived using spectral decomposition (eigen-decomposition) of the covariance matrix of these twenty six variables. Matrix multiplication of $\mathbf{Z}$ and $\mathbf{\Sigma}^{\frac{1}{2}}$ results in a simulated matrix with the same covariance structure as that derived from the twenty six variables in the Nepal dataset as follows,

$$\begin{aligned} \text{var}\left(\mathbf{Y}_{sim}\right) &= \text{var}\left(\mathbf{Z}\,\mathbf{\Sigma}^{\frac{1}{2}} + \boldsymbol{\mu}\right) \\ &= \mathbf{\Sigma}^{\frac{1}{2}}\,\text{var}\left(\mathbf{Z}\right)\mathbf{\Sigma}^{\frac{1}{2}} \\ &= \mathbf{\Sigma}\,. \end{aligned}$$

Since $(\mathbf{Z})$ is a matrix of standardised normal random variables, then $\text{var}\left(\mathbf{Z}\right) = 1$.

A covariance matrix can be constructed only from numeric variables, so the factor variables of the original NLSS dataset were first converted into numeric integer predictors. Each level of a particular categorical variable was assigned an integer rather than character label, then the class of the categorical variable changed from factor to integer. In the final stage of the simulation process, the matrix of means of the NLSS variables, μ, was added to ensure the simulated survey data had the same mean structure as the original NLSS data. Since the matrix $\mathbf{Z}$ was generated from standard normal random variables, each of which has mean of zero, incorporating the means of the original predictors into the simulated datasets ensures that they are more realistic. The simulation model, as described by Equation (5.2), does not incorporate complex survey design, makes no allowance for survey variables which are not continuous, and the assumption of a multivariate normal distribution for model variables is not upheld.

5.6 Simulation process

The following process was used to generate poverty estimates from the simulated survey and ilaka datasets;

1. simulate a survey dataset of size $n \times 26$ according to Equation (5.2)
2. simulate a fixed ilaka dataset of 6000 observations according to Equation (5.2)
3. build a classification tree from a subsample of the simulated survey with tree options of $cp = 0$, minimum tree depth of 5 and minimum split of 20 observations
4. pass the simulated ilaka data through the classification tree built in Step 3.
5. generate predictions for each household in the simulated ilaka dataset
6. aggregate these predictions over all households in the ilaka to provide a measure of poverty incidence, the proportion of poor in the ilaka
7. repeat Steps 3 - 6 for multiple subsamples of the simulated survey data, to obtain multiple estimates of poverty incidence for the ilaka

8. the mean and standard deviation of these multiple predictions provide a small area estimate of poverty incidence, $\hat{\theta}_i$ with associated standard error, $\hat{\sigma}_i$

9. run 100 iterations of the Monte Carlo simulation outlined in Steps 1 to 8

The objective of the simulation exercise was to study the distributions of poverty incidence, $\hat{\theta}_i$, and its standard error (s.e.), $\hat{\sigma}_i$, under different methods of variance estimation. Since the simulated data did not contain clustering or stratification effects it resembled data with a simple random sampling structure.

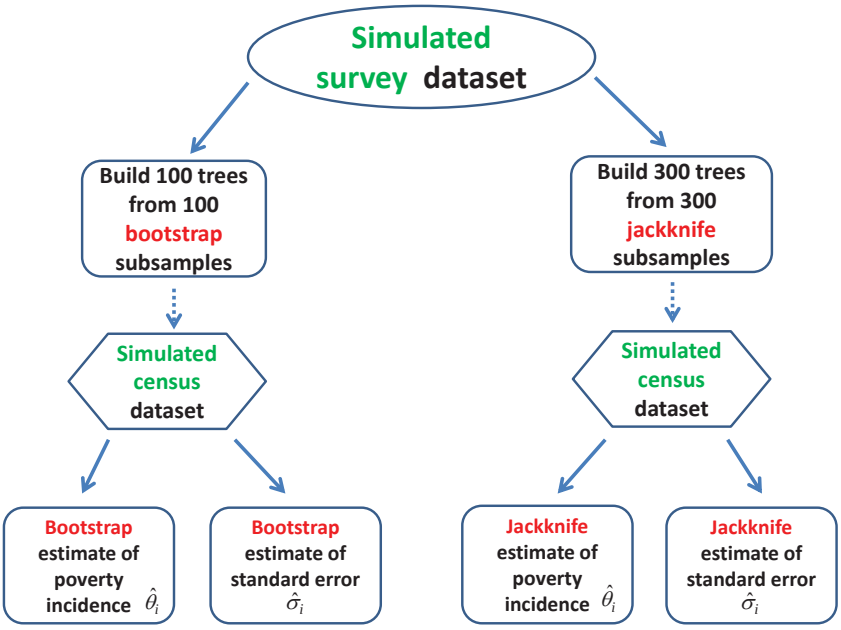
Survey datasets used to build the tree varied according to the subsample, whereas a single fixed small area dataset was used to generate predictions for each iteration. Utilising a different small area dataset for each subsample is equivalent to taking a superpopulation approach. Instead, the initial approach was use a single simulated ilaka dataset which represented one domain in the population. The response variable from this ilaka dataset, log expenditure, which indicated the level of household expenditure, was omitted in the prediction process, but provided a "true" measure of poverty incidence of $\theta = 0.1962$. As a measure of the true poverty incidence for the ilaka, θ became the gold standard by which to judge the accuracy of predictions generated from each subsample.

Initial simulation modelling involved jackknife resampling with hard tree estimates, using survey datasets of size $n = 300$, 3000 and 30000, with delete-1 and delete-5 jackknife resampling. Surprisingly, results showed that, even with a large survey dataset size of 30000, jackknife hard estimates were still producing unstable predictions. The analysis was extended to compare jackknifing variance estimation with bootstrapping for survey sizes $n = 300$ and 3000, as outlined in Figure 5.1. These values were chosen because 300 is close to the size of replicates used in Chapter 4, while 3000 is fairly similar to the actual Nepal survey size of 3912. For a simulated survey size of 300, a delete-1 jackknife was used, and for survey size 3000 a delete-10 jackknife was employed, to retain the same number of jackknife subsamples, 300, for each value of n. A hundred bootstrap subsamples were generated for both values of n. The results of the simulation involving jackknife and bootstrap resampling is discussed in the next section.

5.7 Results of simulations using jackknife and bootstrap resampling

The simulation exercise examined the distribution of predictions of poverty under jackknife and bootstrap resampling. Statistics computed were the estimate of poverty incidence, $\hat{\theta}_i$ and the standard error of prediction, $\hat{\sigma}_i$. Prediction bias was also computed as the difference between $\hat{\theta}_i$, predicted P0, and the "true" poverty level of the simulated ilaka, 0.1962, so that $Bias = \hat{\theta}_i - 0.1962$. Another relationship investigated was how the type of prediction extracted from the tree affected results, with hard estimates being contrasted with soft estimates. A soft estimate measures the probability of a household's being poor, as opposed to labelling a household as poor or not poor, a hard estimate. Since the quantity

Figure 5.1: Flowchart describing the simulation process



of interest is the proportion of poor households in a small area, poverty incidence (P0), averaging hard estimates to obtain a proportion of poor is comparable to taking the average of soft estimates, the probability of being poor. Table 5.1 lists average prediction bias, $\hat{\theta}_i - 0.1962$, and mean of the standard errors of prediction, $\hat{\sigma}_i$, across the 100 simulations for each of the four variance estimation methods, jackknife hard, jackknife soft, bootstrap hard and bootstrap soft, for simulated survey sizes of 300 and 3000 respectively.

Table 5.1: Average prediction bias and s.e. from 100 simulations, for two different survey sizes

Estimate Type	**n = 300**		**n = 3000**	
	Bias	**SE**	**Bias**	**SE**
JK hard	-0.04411	0.26660	-0.08798	0.22646
JK soft	-0.00094	0.07215	0.00138	0.02402
BS hard	-0.02743	0.04368	-0.07521	0.02643
BS soft	-0.00403	0.02442	-0.00003	0.00726

Soft predictions of poverty were reasonably close to the true poverty level of the ilaka, but hard estimates showed large biases. Bias also increased with increasing survey sample size. Increasing the sample size has tended to reduce the standard error of prediction, and bootstrap estimates were less variable than jackknife predictions. However, the key issue here is not in producing the smallest possible standard error, but whether the standard errors are reliable. This was tested by building confidence intervals, at a specified nominal confidence level, for each simulation and each variance estimation method, and seeing what proportion of these one hundred intervals contained the true census value of poverty incidence, as discussed in the next section.

5.8 Validity of estimated standard errors

An essential feature of the modelling process is finding a variance estimation method which will produce estimates of variability which are reasonable. The validity of a method can be tested by examining the coverage of the standard errors it generates, where coverage is defined as the proportion of intervals built which contain the true poverty incidence. In this simulation the true level of poverty is known, it is poverty incidence for the simulated census data, $\theta = 0.1962$. A robust estimation method will produce standard errors which are valid, but not necessarily very small. Validity of standard errors equates to good coverage. Since the objective of the simulation process is to generate predictions, then the intervals under discussion should more properly be referred to as *prediction* intervals.

From each survey dataset, hard and soft estimates of poverty, $\hat{\theta}_i$, and their associated standard error $\hat{\sigma}_i$ were computed, using jackknife and bootstrap resampling methods. Prediction intervals at various levels of confidence were constructed from these statistics for each survey dataset. The actual coverage of one hundred jackknife, or bootstrap, intervals was examined to see how well it matched the nominal coverage. Nominal coverage describes the specified confidence level of the interval, e.g. 95%, 90%, 80% and 68%. For example, a prediction interval at a nominal level of 95% is constructed from $\hat{\theta}_i \pm 1.96 \times \hat{\sigma}_i$. Perfect coverage is an unreal expectation, since this requires a symmetric, normally distributed sampling distribution. However, coverage provides a means of assessing whether the "standard errors" of prediction are approximately the right size.

Using a hundred simulated survey datasets, one hundred intervals with nominal coverage of 95%, $\hat{\theta}_i \pm 1.96 \times \hat{\sigma}_i$, were generated from predictions of poverty incidence, $\hat{\theta}_i$, and associated standard error, $\hat{\sigma}_i$, for each of the four estimation methods. Figures 5.2 and 5.3 display patterns of actual coverage, for nominal coverage of 95%, of prediction intervals generated from simulated survey sample sizes of 300 and 3000 respectively. The four estimation methods compared employed jackknife and bootstrap resampling, with both hard and soft estimate types. Tree models were built using minimum split value of 20. Delete-1 jackknife resampling was employed for a survey size of 300, and delete-10 jackknife for simulated survey size of 3000. The use of a single small area dataset for each simulation provided the same estimated value of "true" poverty incidence for each prediction interval, so that the 100 intervals could be plotted on the same graph. The true level of poverty for the simulated census dataset, $\theta = 0.1962$, is indicated in the plots by a red line on each graph. The actual coverage for each estimation method, the proportion of intervals that cross the red line, is displayed in blue.

For meaningful comparison of interval widths for each estimation method, the X-axis scale was kept constant for all variance estimation methods except the jackknife hard procedure. Since the hard jackknife estimates are much more variable than the other three types of estimates, the X-axes for those graphs have a different scale. Contrasting the graphs for survey size of 300 (Figure 5.2) with those for survey size of 3000 (Figure 5.3) provides visual evidence of the increasing prediction bias with hard as opposed to soft estimates, and increased precision when bootstrapping is used. The coverage under jackknife

Figure 5.2: Actual coverage of a 100 intervals for a nominal level of 95% for survey size 300

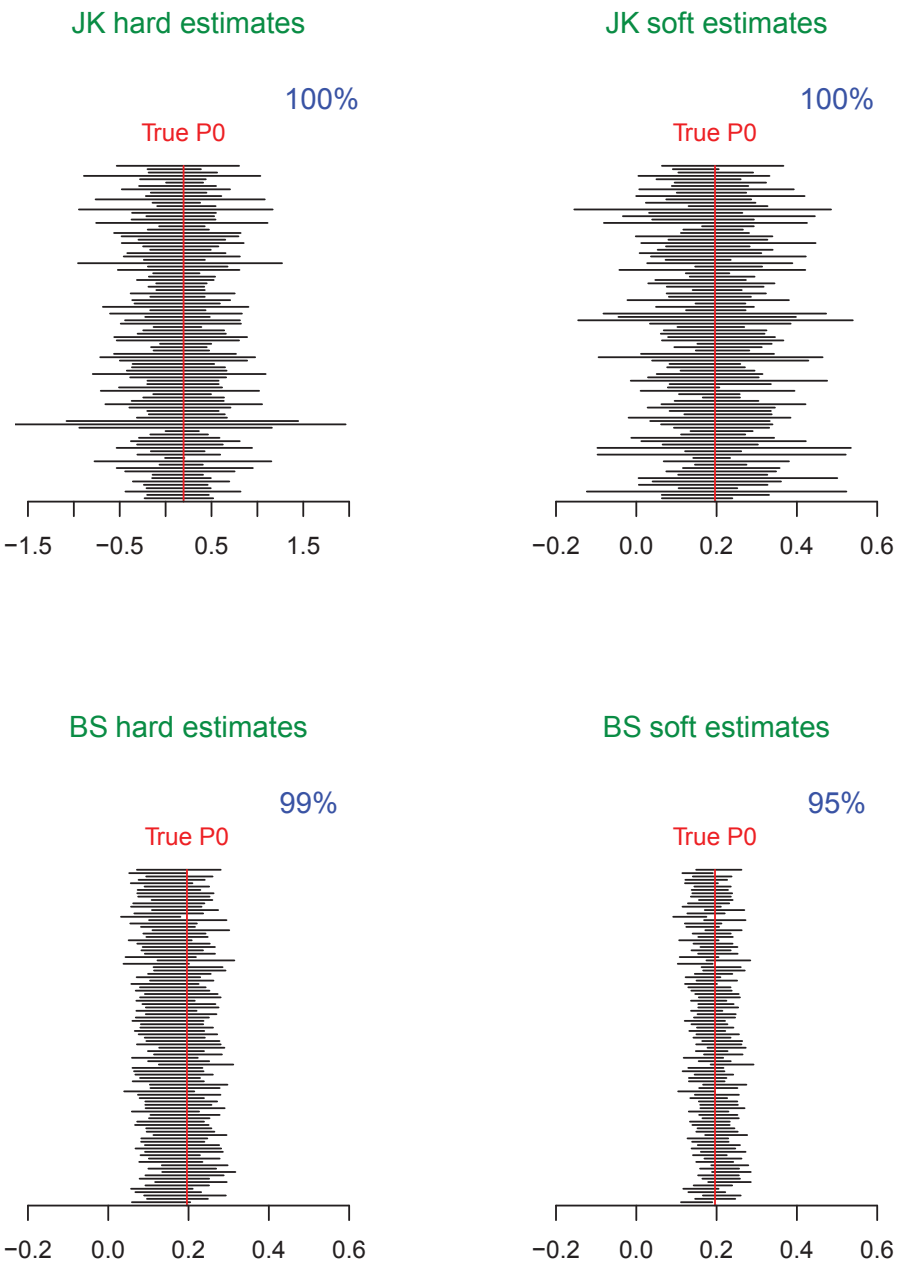
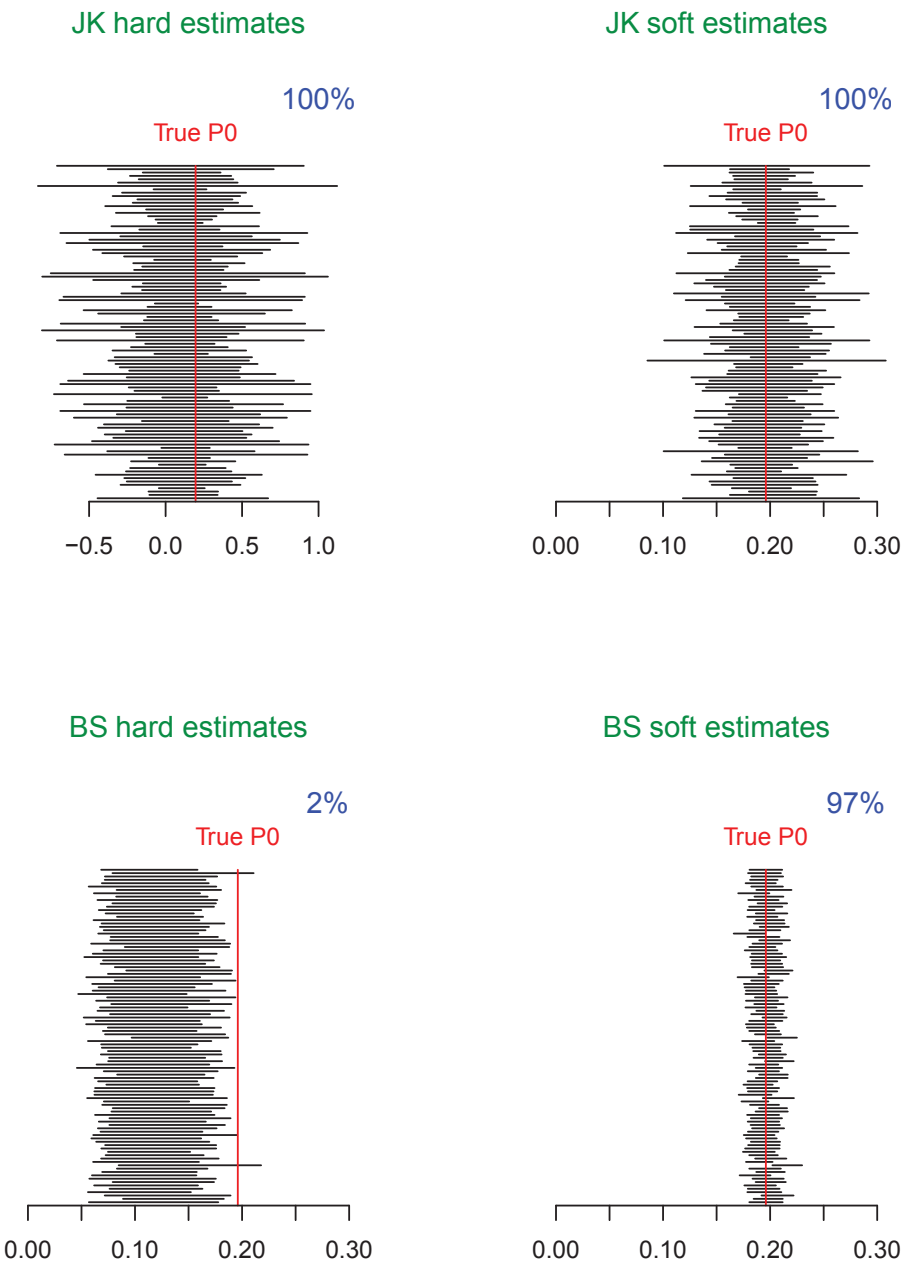


Figure 5.3: Actual coverage of a 100 intervals for a nominal level of 95% for survey size 3000



resampling was a hundred percent, even with very biased poverty estimates, an indication that estimated standard errors of prediction are much larger than actual standard errors of prediction. However, good coverage due to very large variance is not useful. Bootstrap hard estimates had wildly differing coverage levels, 99% for survey sample sizes of 300 and 2% with 3000 observations. The latter result is due to biased estimates in conjunction with small standard errors. Bootstrap soft estimation, with actual coverage of 95% and 97%, had consistently small bias and high precision.

One striking feature of these plots is the severe undercoverage with bootstrap hard methodology for survey size of 3000, which demonstrates that bias in predictions is worse with the larger sample size. The bias occurs because the fixed small area dataset provides a fixed value for true poverty, whereas varying the survey dataset for each simulation generates a distribution of model estimates of poverty which vary about the mean of the distribution. As a general illustration, suppose we generate several sets of training data for a particular model from a $N(0,1)$ distribution and also a single prediction dataset from a $N(0,1)$. Then the mean of the prediction data will be fixed, whereas the means of the training sets will vary about 0. The bias resulting from using a fixed small area is dependent upon the size of the small area, in terms of both the number of clusters and the number of households. In modelling of the Nepal data, bias is particularly affected by the small number of clusters.

Not only were the jackknife errors larger than those for bootstrapping, but they were also more variable. These results suggest that jackknife resampling is a poor choice of variance estimation method for tree based models, an outcome not entirely unexpected. Although the asymptotic consistency of the jackknife estimator of variance has been demonstrated by Miller (1964),The grouped jackknife is often used to address this problem, but has not been successful for the hard classification tree predictions in this study. The tree algorithm is not a smooth process, being a "greedy" top-down binary partitioning method.

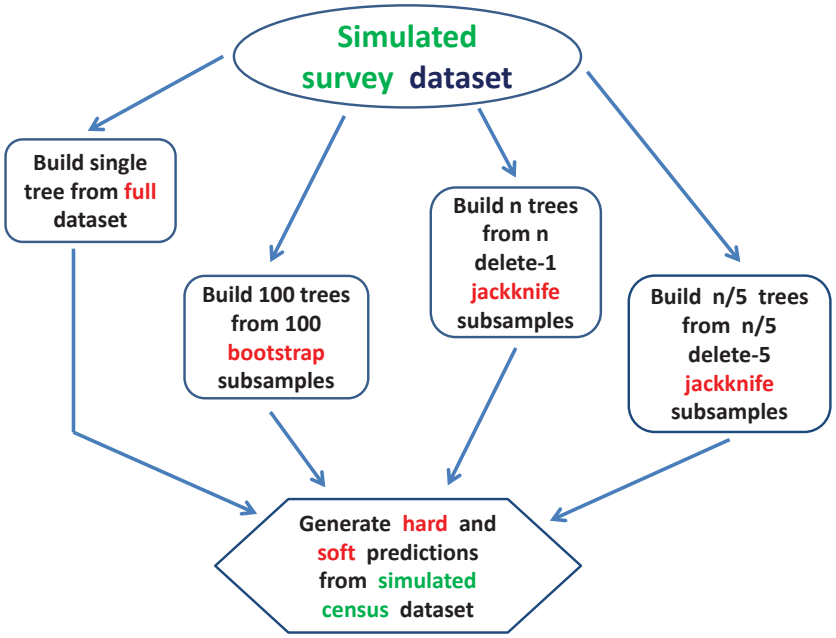
As shown in the plots in Figures 5.2 and 5.3, the granular nature of a hard estimate, classifying a household as poor or not poor, also seems to contribute to the larger bias and less precision of this type of tree prediction, as compared with employing soft estimation, the probability of being poor. The inconsistency of hard estimation is particularly evident in boundary situations, when poverty incidence at a node is close to 0.5. A small change in the data passing through a node can change the prediction completely. The simulation exercise suggests that instability of hard jackknife tree estimates is worse with larger survey sample sizes. The analysis to this point provides evidence of bootstrap resampling with soft tree estimates as a valid method of estimation for data with a simple random sampling structure. A designed experiment was set up to assess this evidence.

5.9 Experimental design

5.9.1 Outline of the designed experiment

The purpose of the designed experiment was to broaden the simulation exercise and investigate which factors were driving the precision and accuracy of poverty estimates using the classification tree method. The specific factors examined were the variance estimation method applied, jackknife or bootstrap resampling, and the estimate type, hard or soft. Delete-1 and delete-5 jackknife methods were included in the investigation. Figure 5.4 provides a flowchart of the algorithms used in the designed experiment. The optimal size of survey dataset may also be important, so three levels were included in the design; 300, 1500 and 3000 observations. The experiment also considered what would be the best minimum number of observations in a node for a split to occur, the levels selected being 10, 20 and 40 observations.

Figure 5.4: Flowchart of algorithms used in the designed experiment



The accuracy of each estimation method was tested by examining bias of the estimates, and precision in terms of relative standard error, the latter statistic being the ratio of estimated standard error to true standard error. Validity of the standard errors was also tested by examining coverage of one hundred prediction intervals. Bias in the estimates was computed as the difference between poverty as predicted by the model, $\hat{\theta}_i$, and the true poverty level, θ. Poverty incidence of the simulated census dataset provided a measure of the true poverty level, θ. From the response variable, log expenditure, for each survey dataset a measure of poverty, θ_i, was obtained by categorising a household as poor if it was below the poverty line of log expenditure, which equals 8.95.

Since the modelling used only a single census dataset, a measure of true standard error, σ, was obtained from the hundred survey datasets generated. Each different simulated survey provided a different value of θ_i, thus the variability of these one hundred predictions of poverty, i.e. their standard deviation, was used to provide a measure of the true standard error, σ, of poverty predictions under the model. This approach is feasible since the simulated survey and census datasets are independent and generated from the same model. Then the relative standard error for each variance estimation method was calculated as the jackknife or bootstrap estimated standard error of predictions, $\hat{\sigma}_i$, divided by the true standard error, σ. True standard error, σ, for each of the three survey sizes is listed in Table 5.2.

Table 5.2: True standard error for different survey sizes

Simulated survey size	300	1000	3000
True standard error	0.0247	0.0134	0.0061

Each of the 100 simulated survey datasets supplied different estimates of poverty incidence, $\hat{\theta}_i$, and associated standard error, $\hat{\sigma}_i$, for each combination of the levels of all four factors, the variance estimation method, estimate type, survey sample size and minimum split value. From these statistics were computed measures of bias, standard error, relative standard error and coverage. Jackknife or bootstrap estimates of poverty, $\hat{\theta}_i$, and associated standard errors, $\hat{\sigma}_i$, from the one hundred survey datasets were used to build a hundred prediction intervals to investigate coverage for nominal coverage levels of 95%, 90%, 80% and 68%, across each combination of factor levels. The statistics for the hundred survey datasets were then aggregated to provide overall measures of bias and relative error for each combination of factor levels.

The factor variable incorporating variance estimation type also included a level, "Full", representing no variance estimation. A single tree model was built from the full simulated survey dataset, i.e. without using jackknife or bootstrap resampling, and the simulated census data passed through this "Full" tree to provide an estimate of poverty incidence, $\hat{\theta}_i$ not based on resampling. For each full tree model, statistics quantifying bias, standard error, and relative standard error were also computed. The full tree model did not provide an estimate of the standard error of prediction, $\hat{\sigma}_i$, so this quantity was derived from the standard deviation of estimates of poverty incidence, $\hat{\theta}_i$, derived from the hundred survey samples. In contrast, $\hat{\sigma}_i$ under the resampling methods was calculated as the average of three hundred jackknife or a hundred bootstrap estimates of error, for each simulated survey dataset. Thus, the full tree model did not provide a measure of coverage. The purpose for generating estimates from all the survey data was to provide a level of comparison by which to judge the performance of estimates under jackknife and bootstrap resampling.

The statistics computed from the hundred survey datasets display patterns of accuracy and precision for poverty estimates under various estimation methods. A good

technique will produce estimates of poverty incidence similar to the true poverty incidence of the census dataset. It will also demonstrate standard error akin to true standard error, σ, provided by the survey datasets, through a relative standard error of approximately 1. Analysis of variance models were run for the response variables bias, relative standard error and coverage. Predictors for the models comprised variance estimation method, estimate type, survey sample size and minimum split value. Results of the designed experiment for bias, relative standard error and coverage are discussed separately.

5.9.2 ANOVA results for bias

All the analysis of variance models built for response variables had very high R^2 values, greater than 98%. The ANOVA output for the analysis of bias in predictions of poverty is displayed in Figure 5.5, and shows that estimate type is the most significant factor in determining bias, accounting for 85% of the model variability. The significance of other, less important factors, is due to a very small residual variability.

Figure 5.5: ANOVA table for analysis of bias of variance estimation methods

```
Analysis of Variance Table

Response: Bias
                  Df   Sum Sq  Mean Sq    F value    Pr(>F)
Method             3 0.000553 0.000184   115.8304 2.066e-14 ***
Type               1 0.071941 0.071941 45203.5903 < 2.2e-16 ***
n                  2 0.004142 0.002071  1301.4468 < 2.2e-16 ***
split              2 0.000499 0.000249   156.7424 1.673e-14 ***
Method:Type        3 0.000998 0.000333   209.0247 < 2.2e-16 ***
Method:n           6 0.000014 0.000002     1.4971  0.221514
Type:n             2 0.005402 0.002701  1697.0622 < 2.2e-16 ***
Method:split       6 0.000075 0.000013     7.8973 9.040e-05 ***
Type:split         2 0.000398 0.000199   125.0375 2.033e-13 ***
n:split            4 0.000481 0.000120    75.5327 3.056e-13 ***
Method:Type:n      6 0.000038 0.000006     3.9742  0.006689 **
Method:Type:split  6 0.000035 0.000006     3.6168  0.010685 *
Type:n:split       4 0.000349 0.000087    54.7888 1.010e-11 ***
Residuals         24 0.000038 0.000002
```

Soft estimation is the driver of small bias, as indicated in the section of the table of coefficients in Figure 5.6. The intercept for the ANOVA model of bias represents the bootstrap resampling method with soft estimation, survey size of 300 and minimum split of 10, which has a bias of approximately -0.0044. The bias of soft estimates has reduced under jackknife resampling for variance estimation, and the method that did not use resampling, by approximately 0.006. Bias under soft estimation also decreases with increasing survey sample size, but minimum split value has no appreciable effect. However, when the hard estimation method is employed with bootstrap resampling, there is a very large increase in bias, around 2.0.

Figure 5.6: Table of coefficients for analysis of bias of variance estimation methods

```
Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                       -4.369e-03  9.258e-04  -4.719 0.000498 ***
Methoddel-1  JK                    6.522e-03  1.265e-03   5.156 0.000239 ***
Methoddel-5  JK                    6.188e-03  1.265e-03   4.892 0.000371 ***
MethodFull                         6.381e-03  1.265e-03   5.044 0.000287 ***
TypeHard                          -2.049e-02  1.171e-03 -17.498 6.60e-10 ***
n1000                              3.408e-03  1.242e-03   2.743 0.017817 *
n3000                              5.613e-03  1.242e-03   4.519 0.000703 ***
split20                            7.418e-04  1.242e-03   0.597 0.561485
split40                            2.449e-03  1.242e-03   1.972 0.072109 .
Methoddel-1  JK:TypeHard          -1.861e-02  1.512e-03 -12.309 3.64e-08 ***
Methoddel-5  JK:TypeHard          -1.720e-02  1.512e-03 -11.378 8.73e-08 ***
MethodFull:TypeHard               -1.942e-02  1.512e-03 -12.842 2.26e-08 ***

.........................

Multiple R-squared: 0.9996,     Adjusted R-squared: 0.9987
```

Applying jackknife resampling or no resampling with hard estimation almost doubles the increase in bias seen with the bootstrap method, with effect sizes of 1.7 to 1.9 (Figure 5.6). The soft estimation process is clearly the most important factor in reducing bias in the estimates, $\hat{\theta}_i$, of poverty incidence.

5.9.3 ANOVA results for relative standard error

Results of the ANOVA model for the analysis of relative standard error of the estimates are given in Figure 5.7. The analysis of variance table in Figure 5.7 indicates that minimum split has a very small effect on relative standard error, and only through its interaction with estimate type. Resampling method, estimate type and survey sample size, through their main effects, 2-way and 3-way interactions, explain 99.9% of the variability in relative standard error. The proportions of variability attributable to the individual effects are listed in Table 5.3.

Table 5.3: Percentage variability explained by by Method, Type, Survey size and their interactions

Effect	Method	Type	n	Method:Type	Method:n	Type:n	Method:Type:n
% SS	25.50	21.50	11.90	15.50	9.10	9.20	7.04

Since the model for relative standard error contains a complex structure of interactions, only portions of the output of coefficients are displayed in the coefficients table, in Figure 5.8. The intercept term indicates that soft bootstrap estimates from the model with survey size of 300 and minimum split of 10 have average relative error of 1.1. A value close to unity indicates reasonable precision in the estimates. There is no significant effect on the relative error under bootstrap soft estimation by increasing survey size or minimum split error.

Figure 5.7: ANOVA table for analysis of relative s.e. for variance estimation methods

```
Analysis of Variance Table

Response: Rel.se
                  Df  Sum Sq Mean Sq    F value    Pr(>F)
Method             3 2886.90  962.30  8338.4171 < 2.2e-16 ***
Type               1 2440.57 2440.57 21147.8027 < 2.2e-16 ***
n                  2 1347.90  673.95  5839.8565 < 2.2e-16 ***
split              2    0.17    0.09     0.7407  0.487355
Method:Type        3 1780.84  593.61  5143.7044 < 2.2e-16 ***
Method:n           6 1029.92  171.65  1487.3992 < 2.2e-16 ***
Type:n             2 1042.49  521.25  4516.6515 < 2.2e-16 ***
Method:split       6    0.10    0.02     0.1380  0.989761
Type:split         2    3.83    1.92    16.6113 2.963e-05 ***
n:split            4    1.26    0.31     2.7230  0.053229 .
Method:Type:n      6  798.15  133.03  1152.6749 < 2.2e-16 ***
Method:Type:split  6    2.41    0.40     3.4866  0.012719 *
Type:n:split       4    2.24    0.56     4.8601  0.005152 **
Residuals         24    2.77    0.12
```

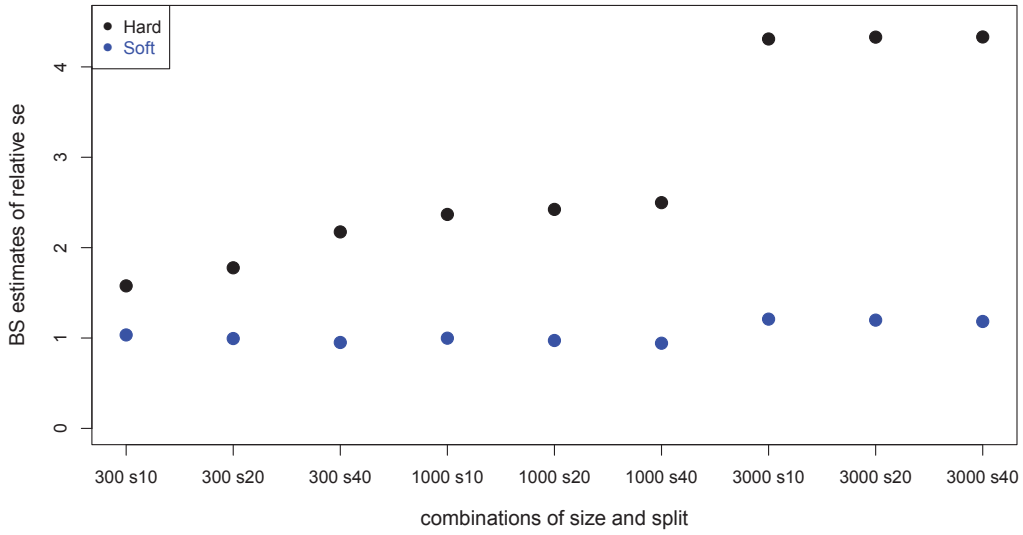
Using the full tree model instead of bootstrap resampling to obtain poverty estimates also does not affect relative standard error of predictions. Bootstrapping with hard estimates has no effect for survey size of 300, but the relative standard errors of bootstrap hard estimates increase with increasing survey size, and also with minimum split of 40. Using the jackknife approach, relative standard errors under soft estimation, are much larger than those for the bootstrap method, around 4 for the delete-1 method and 3 for the delete-5 method.

Figure 5.8: Table of coefficients for analysis of relative s.e. of variance estimation methods

```
Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                      1.10662    0.27738   3.990 0.000541 ***
Methoddel-1  JK                  2.31345    0.35809   6.461 1.11e-06 ***
Methoddel-5  JK                  1.64898    0.35809   4.605 0.000113 ***
MethodFull                      -0.03658    0.35809  -0.102 0.919493
TypeHard                         0.20323    0.39227   0.518 0.609145
n1000                           -0.14073    0.33971  -0.414 0.682368
n3000                            0.06587    0.33971   0.194 0.847886
split20                         -0.12864    0.33971  -0.379 0.708253
split40                         -0.20846    0.33971  -0.614 0.545231
Methoddel-1  JK:TypeHard         6.27677    0.50642  12.395 6.38e-12 ***
Methoddel-5  JK:TypeHard         4.30558    0.50642   8.502 1.06e-08 ***
MethodFull:TypeHard             -0.08312    0.50642  -0.164 0.871005
.....................................................

TypeHard:n1000                   1.25874    0.48043   2.620 0.015008 *
TypeHard:n3000                   3.14335    0.48043   6.543 9.11e-07 ***
 .....................................................

TypeHard:split20                 0.48420    0.48043   1.008 0.323583
TypeHard:split40                 1.45491    0.48043   3.028 0.005801 **
 .....................................................

Multiple R-squared: 0.9998,     Adjusted R-squared: 0.9993
```

The grouped jackknife has reduced the instability to some degree, but not enough. When the hard estimation method is used with jackknife resampling, relative standard error becomes extremely large, with the value at survey size of 3000 being 50 for the delete-1 jackknife and 41 for the delete-5 jackknife. The patterns of change are better illustrated in the scatterplot in Figure 5.9 which graphs relative standard error of hard and soft estimates for different survey sizes and minimum split values using the bootstrap (BS) method. Relative standard error in soft bootstrap estimates is fairly constant for all survey sizes and minimum split values. Hard bootstrap estimates of relative standard error increase with increasing survey size, but are not affected by changing minimum split, except for minimum split of 40 with a survey size of 300. As indicated by the coefficients table (Figure 5.8), there is a significant difference between soft and hard bootstrap estimates of relative standard error for survey sizes of 1000 and 3000, and for minimum split of 40 with survey size 300.

Figure 5.9: Relative standard error for BS method for different sample sizes and minimum split values



A very similar pattern was seen when relative standard error of hard and soft estimates generated by the jackknife sampling methods were plotted against survey size and minimum split. The variability of soft estimates remained fairly stable, but relative standard error for the hard estimates increased markedly with increasing survey size. However, relative standard error for estimates from the full tree differed very little from corresponding bootstrap estimates, i.e. for the same estimate type and survey size. In summary, bootstrapping is the best resampling method for generating reasonable standard errors, since it produces relative standard error close to 1. In contrast, jackknifing resulted in relative standard errors which were much larger than 1, some being extremely large.

5.9.4 ANOVA results for coverage

Actual coverage was investigated for four levels of nominal coverage, 95%, 90%, 80% and 68%. Figure 5.10 displays the table of coefficients for the ANOVA of actual coverage for a nominal level of 95%; results for the other three nominal coverage levels were very similar. Minimum split was not included in the model since it had no effect on coverage. The patterns of actual coverage for the four nominal levels matched those seen in the plots in Figures 5.2 and 5.3, which each display a hundred confidence intervals simulated for nominal coverage of 95% using jackknife and bootstrap resampling, with hard and soft estimates and survey sizes of 300 and 3000. The intercept in Figure 5.10 represents bootstrap soft estimation for survey size of 300. Bootstrap soft estimation produced reasonable coverage, generally a few points above the nominal value. Bootstrap hard estimates provided good coverage for survey size of 300, but coverage decreased markedly as survey size increased. The jackknife resampling method consistently produced overcoverage for both soft and hard types of estimates, with actual coverage of 94% to 99% even for a nominal coverage of 68%. Bootstrap resampling with soft estimates provided the best coverage, and also produced small standard errors, which were in the same order of magnitude as the published standard errors generated by the ELL method (Haslett & Jones 2006).

Figure 5.10: Table of coefficients for analysis of coverage of variance estimation methods

```
Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                      96.0000     0.7158 134.119  < 2e-16 ***
Methoddel-1  JK                   3.3333     1.0123   3.293  0.00223 **
Methoddel-5  JK                   3.3333     1.0123   3.293  0.00223 **
TypeHard                          2.0000     1.0123   1.976  0.05589 .
n1000                             2.3333     1.0123   2.305  0.02704 *
n3000                             1.0000     1.0123   0.988  0.32981
Methoddel-1  JK:TypeHard         -1.3333     1.4316  -0.931  0.35786
Methoddel-5  JK:TypeHard         -1.3333     1.4316  -0.931  0.35786
Methoddel-1  JK:n1000            -1.6667     1.4316  -1.164  0.25199
Methoddel-5  JK:n1000            -1.6667     1.4316  -1.164  0.25199
Methoddel-1  JK:n3000            -0.3333     1.4316  -0.233  0.81720
Methoddel-5  JK:n3000            -0.3333     1.4316  -0.233  0.81720
TypeHard:n1000                  -26.6667     1.4316 -18.628  < 2e-16 ***
TypeHard:n3000                  -96.0000     1.4316 -67.059  < 2e-16 ***
Methoddel-1  JK:TypeHard:n1000   25.6667     2.0245  12.678 7.69e-15 ***
Methoddel-5  JK:TypeHard:n1000   26.0000     2.0245  12.842 5.25e-15 ***
Methoddel-1  JK:TypeHard:n3000   95.3333     2.0245  47.089  < 2e-16 ***
Methoddel-5  JK:TypeHard:n3000   95.3333     2.0245  47.089  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Multiple R-squared:  0.998,     Adjusted R-squared:  0.997
```

5.10 Conclusion

The Monte Carlo simulations indicated that bootstrap sampling using soft estimates was found to be the best method of variance estimation for data with a simple random sampling structure. There was no apparent effect of split and no consistent effect of sample size across the three criteria, bias, relative standard error and coverage. Soft estimation minimised bias, and bootstrap sampling minimised standard error. Bootstrap sampling with soft estimation had the smallest relative error, around 1. It also provided the best coverage, actual estimates being consistently only a few points above the nominal value. Results of the designed experiment confirm bootstrapping with soft estimates as the best method of variance estimation for classification trees under simple random sampling. The next step is to extend the bootstrap soft estimation method for data containing elements of complex survey design, such as stratification and clustering. This is examined in the subsequent chapter.

Chapter 6

Adapting classification trees for complex survey data

6.1 Introduction

The scope of the thesis is adaptation of tree based models for small area estimation of poverty, utilising survey and census data from Nepal. To achieve this goal, the complex survey design elements of clustering and stratification need to be incorporated into classification tree methodology. The Monte Carlo study, described in Chapter 5, was set up to examine the coverage properties of several variance estimation methods applied to classification tree models. The objective was to determine minimum sample size for stability of tree predictions, and to compare the performance of jackknife and bootstrap resampling methods, as well as hard and soft tree estimates, for models built from data with a simple random sampling structure. Simulation results suggested that the method of bootstrapping with soft tree predictions provided reasonable coverage. The next step is to expand this methodology for complex survey data, and test its validity by means of a Monte Carlo study. In this chapter, the methodology applied to simple random sample data is modified to investigate the coverage properties of bootstrap soft estimation when applied to clustered data.

An important application of Monte Carlo simulations is in answering a research question for which there is no reference measurement, or "gold standard" (Kang et al. 2013, Francq & Govaerts 2014). In the context of small area estimation of poverty in Nepal, there is no "gold standard" measurement of poverty incidence for each small area. When adapting tree models for complex survey data, the validity of the chosen variance estimation method was examined by employing a Monte Carlo simulation to generate datasets which mimicked the properties of the Nepal survey data.

6.2 Monte Carlo simulation for clustered data

The Monte Carlo study of Chapter 5 investigated stability in tree predictions for data collected through simple random sampling. In this chapter, the methodology described in Section 5.6 is extended to clustered data by introducing a random cluster effect into the structure of the simulated datasets, applying the cluster bootstrap procedure for resampling and incorporating cluster effects into predictions. Coverage properties of this amended procedure for bootstrap resampling with soft estimates was examined using Monte Carlo simulation.

6.3 Introducing clustering into the model

The simulation exercise outlined in Chapter 5, carried out under a simple random sampling structure which assumed independence of the observations, generated survey and small area datasets based on a linear model, as outlined in Steps 1 and 2 of the simulation process described in Section 5.6. Simulated survey datasets had 3000 observations and small area datasets were of size 6000. Adapting the bootstrap soft method for clustered data firstly requires amending Equation (5.1) to include a random cluster effect,

$$Y_{ij} \;=\; X_{ij}\boldsymbol{\beta} \;+ cl_j \;+\; \epsilon_{ij}\,, \tag{6.1}$$

where Y_{ij} represents the response variable, log expenditure, and the cluster effect, cl_j, is assigned the same value for all households in the j^{th} cluster. Households in the simulated datasets were grouped into blocks to represent the clusters, and the same random effect, cl_j, appended to the log expenditure value for each household in the j^{th} block. Survey and small area datasets were then generated from the same linear mixed model represented by Equation (6.1).

To simulate a clustering effect in the data structure, extra variability must be introduced at the cluster level, so that the error structure in the simulated data has the form;

$$\sigma^2_{total} \;=\; \sigma^2_{cl} + \sigma^2_{\epsilon}\,, \tag{6.2}$$

where σ^2_{cl} represents the random variation due to clustering and σ^2_{ϵ} denotes random variation at the household level. A cluster effects parameter, $k = \sigma_{cl}$, was introduced into Steps 1 and 2 of the simulation process outlined in Section 5.6, to represent the cluster variability, σ^2_{cl}, in Equation (6.2). Modelling was carried out using different values of $k = \sigma_{cl}$, chosen so as to correspond approximately to a specified intracluster correlation, ρ. Values of the intracluster correlation coefficient used in the modelling were selected to be representative of the ranges of actual values found in real poverty datasets. The original intraclass coefficient, ρ, proposed by Ronald Fisher, is an unbiased but complex formula (Lohr 1999). A simpler but slightly positively biased estimator of intraclass correlation, ρ, based on the random effects model, has the form;

$$\begin{aligned} \rho &= \frac{\sigma^2_{cl}}{\sigma^2_{total}} \\ &= \frac{\sigma^2_{cl}}{\sigma^2_{cl} + \sigma^2_{\epsilon}} \,. \end{aligned} \tag{6.3}$$

The value of σ^2_ϵ used in the simulations was extracted from the relationship between the response variable and predictors for the simulated datasets. When the response, log(expenditure), was regressed against the 25 predictor variables, the model residuals were found to have a residual standard error of around 0.5. Thus a value of 0.25 was assigned to σ^2_ϵ for the simulation. The value of ρ, the ratio of cluster variation to total residual variation (Equation 6.3), for the ELL modelling of poverty incidence in Nepal was approximately 0.12 (Haslett & Jones 2006). This is equivalent to a cluster effect parameter of approximately $k = 0.18$. Several values were chosen for k, larger and smaller than 0.18, based on the degree of clustering found in real poverty datasets. These values of k, the clusters effects parameter, and approximate corresponding values of ρ are displayed in Table 6.1. The value $k = 0$, representing data without clustering, was included to investigate the effect of applying the cluster bootstrap when there is no clustering in the data.

Table 6.1: Cluster effect values and corresponding intracluster correlations

k	0	0.04	0.08	0.12	0.16	0.20	0.24
ρ	0	0.006	0.025	0.054	0.093	0.138	0.187

The procedure for constructing the random effects in the model was;

1. generate a standard normal variable $\boldsymbol{Z} = (Z_1, Z_2, \cdots, Z_n)$ of length n where $Z_i \sim N(0, 1)$
2. multiple $\boldsymbol{Z}$ by $k = \sigma_{cl}$ to produce a random cluster effect $k\boldsymbol{Z}$
3. add kZ_j to the value of response variable log expenditure for each observation in the j^{th} cluster, to obtain

$$Y^*_{ij} \;=\; Y_{ij} \;+\; kZ_j \,,$$

where Y_{ij} denotes the initial value of log expenditure for the i^{th} household in the j^{th} cluster, and Y^*_{ij} the amended value used in building the tree.

The cluster size for the NLSS dataset was 12. Since the size of the simulated survey used in the simulation was 3000, which equates to 250 clusters of 12 households each, then the length of $\boldsymbol{Z}$ for the simulated survey was 250. For the simulated small area data, clusters of size 150 were established, since this is approximately the average size of clusters in the Nepal census dataset (Haslett & Jones 2006). The simulated small area

data had 6000 observations, so 40 clusters were selected, each of size 150, resulting in the length of $\boldsymbol{Z}$ for the simulated small area being 40.

It is noted that increasing the size of the random cluster effect, kZ_j, appended to the log expenditure variable, Y_{ij} for both survey and small area data, produced a corresponding increase in the level of poverty for both survey and small area datasets. Incorporating clustering into the model as an adjustment to the log expenditure variable, Y_{ij}, resulted in a wider range of values for the amended expenditure variable, Y^*_{ij}, with more households having a value below the poverty line, and so a greater level of poverty incidence overall.

Simulated survey datasets of 3000 observations were used since this represents a typical size for a survey. A survey size of 300 households was included in the analysis in Chapter 5 using data simulated to have a simple random sampling structure, to mimic the scenario in which replicates were used as the variance estimation method. However, to examine the validity of bootstrap soft estimation for clustered data, a simulated survey of 3000 was used, since it more closely resembles the Nepal dataset which includes clustering effects.

Bootstrap resampling with soft tree estimation was then applied to each simulated survey dataset, to build a classification tree model which was then used to predict for the simulated small area data. However, the classical bootstrap method (Efron 1979) assumes that the data is independently and identically distributed, which doesn't hold with clustered data (Antal & Tillé 2011). To maintain the complex survey design of the simulated data, and take account of the dependence structure in the data (Field & Welsh 2007), sampling with replacement needs to be applied to clusters of households rather than individual households (Cameron et al. 2008).

6.3.1 Bootstrapping the clusters

The usual procedure for using bootstrap sampling of data with cluster effects is to bootstrap the clusters rather than individual households (Field & Welsh 2007). Several methods exist for bootstrapping clustered data. One technique used for bootstrap resampling of data having a clustering structure, known simply as the *cluster bootstrap*, involves selecting clusters using simple random sampling with replacement. A bootstrap sample of clusters is chosen and the ultimate cluster principle applied (Wolter 2007), in which each household in a selected cluster is included in the bootstrap sample, multiple times if its associated cluster is selected more than once. The cluster bootstrap was incorporated into Step 3 of the simulation process outlined in Section 5.6, the tree building stage.

To investigate the performance of the bootstrap soft estimation method amended for clustered data, a thousand Monte Carlo simulations were run for values of the cluster effects parameter, $k = 0,\ 0.04,\ 0.08,\ 0.12,\ 0.16,\ 0.20,\ 0.24$. The simulation results are discussed in the next section.

6.3.2 Performance of the bootstrap soft method for clustered data

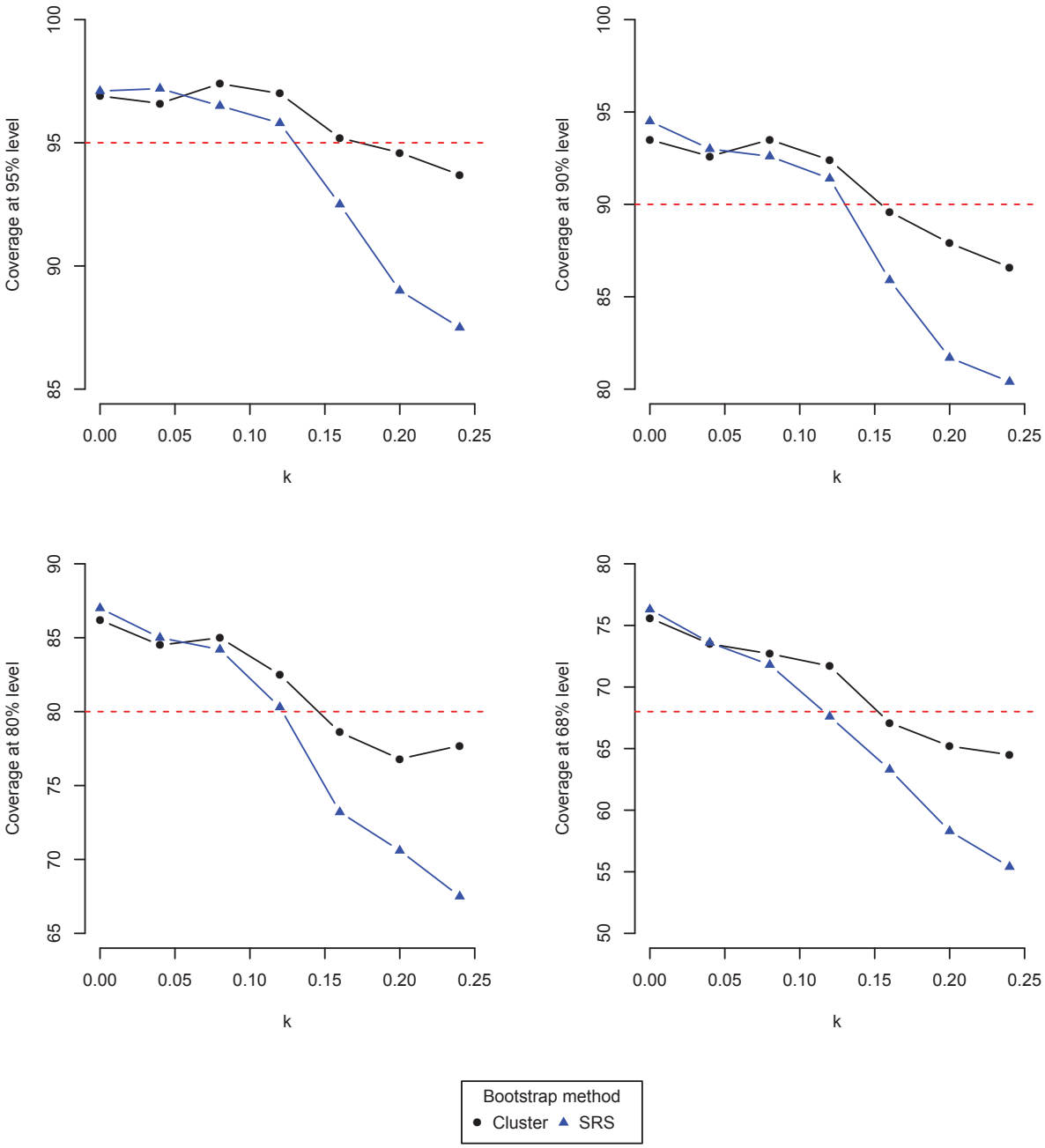
The simulation study to investigate the performance of bootstrap soft estimation when there is clustering in the data was initially carried out using a fixed small area dataset. A different survey dataset was constructed to build a different classification tree model for each of the 1000 simulations, but predictions were based on a single, fixed simulated small area dataset. For each of the 1000 simulations, a bootstrap point estimate of poverty incidence,$\hat{\theta}_i$, and its associated standard error, $\hat{\sigma}_i$, were calculated from 100 bootstrap samples. From these statistics, a thousand prediction intervals were constructed to examine empirical coverage at four different nominal levels, $95\%, 90\%, 80\%$ and 68%. In addition, estimates of bias and relative error were obtained. Since the fixed small area dataset used is the same dataset employed for predictions in Chapter 5, bias was measured as $\hat{\theta}_i - 0.1962$, where 0.1962 is the poverty level of the simulated small area dataset. Relative error was computed as the ratio $\hat{\sigma}_i/\sigma_i$, where σ_i denotes the "true" standard error and is estimated by calculating the standard deviation of the 100 bootstrap estimates of poverty, $\hat{\theta}_i$, obtained from each simulation.

The naive bootstrap method for variance estimation, which utilises sampling with replacement of households rather then clusters as discussed in Chapter 5 , was also included in the simulation process. The reason for applying a simple random sampling bootstrap to the clustered data was to test the assumption that a cluster bootstrap is needed to obtain valid estimates from clustered data. For both types of bootstrapping, bias was small, between 0.0013 and 0.0017, and relative standard error close to 1. However, small standard error may not necessarily result in good coverage, which is the key test of the usefulness of the method. The plots in Figure 6.1 display actual coverage relative to a specified nominal coverage, $95\%, 90\%, 80\%$ and 68%, for different levels of clustering in the data structure, k. The dashed red line on each plot represents the specified nominal coverage level, which was included to provide a reference line by which to assess empirical coverage.

The cluster bootstrap and simple random bootstrap both produced overcoverage when the data was devoid of actual clustering effects. This behaviour reflects the patterns seen in the coverage plots in Section 5.8, using simulated data with a simple random sampling structure. For all levels of clustering, the cluster bootstrap method provided better coverage, with actual coverage values reasonably close to the nominal coverage values. In contrast, actual coverage under the naive bootstrap procedure is significantly less than actual coverage under the cluster bootstrap. Not employing the cluster bootstrap when clustering is present in the data tends to produce undercoverage, the effect being more pronounced as the level of clustering increases.

Using a fixed small area dataset but generating a different survey with each simulation run was also utilised in Chapter 5, to investigate the effectiveness of bootstrap soft variance estimation when the data has been collected using simple random sampling. This approach is akin to design-based modelling (Rao 2003), in which an entire census would be simulated, not just one small area.

Figure 6.1: Actual and nominal coverage for different levels of clustering with a fixed small area dataset, under bootstrapping of clusters, *Cluster*, or bootstrapping of households, *SRS*.



Then, each simulated survey dataset would be constructed as a sample from the simulated census. But a standard design-based approach was not practical for this study, being computationally expensive. In addition, the simulation study is based around prediction over one small area, not the whole census. However, using a fixed simulated small area dataset for prediction may have produced artificially good results. The high coverage probabilities seen in Figure 6.1 may be an artifact of this particular simulated small area.

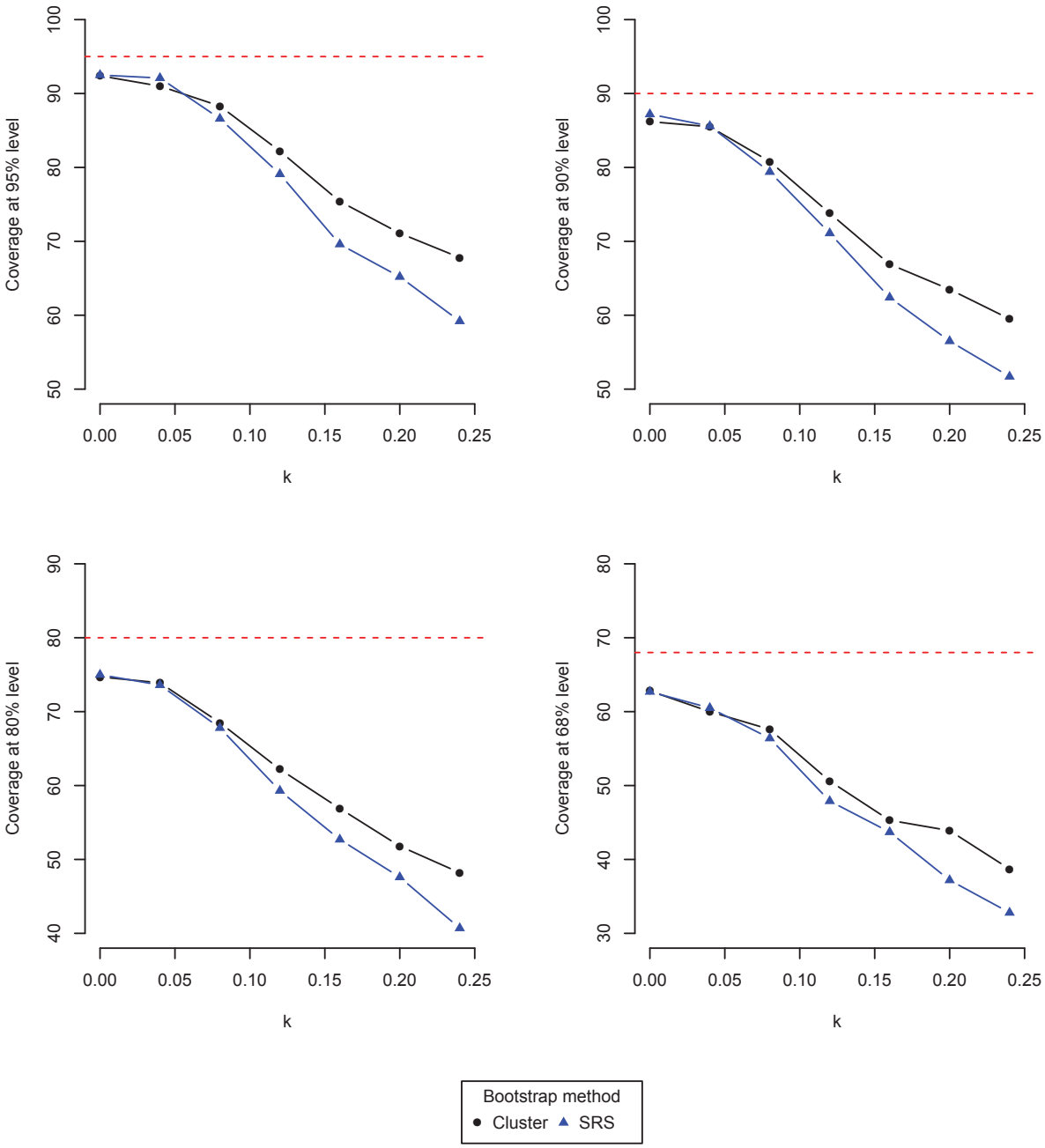
The disadvantage of using a fixed small area dataset in the modelling process is the possibility of bias in predictions, since the fixed small area dataset has its own particular characteristics. An example of this type of bias was seen in Section 5.8. Use of a single small area dataset allowed a graphical illustration of coverage performance for jackknife and bootstrap hard and soft estimation methods with sample sizes of $n = 300$ and 3000, by plotting all the intervals generated under a particular method on the same graph (Figures 5.2 and 5.3). Coverage of bootstrap hard estimates for survey size of 3000 was only 2%, due to the combination of large bias and small standard error in the hard estimates.

Model-based estimation (Lohr 1999), in which survey and census datasets are generated independently from the same model, provides an alternative method of simulation. This approach is comparable with the methodology of ELL, since the survey used for model building in ELL often represents a different time period to the census used for prediction. The Monte Carlo exercise was modified to simulate a new small area dataset with each simulation, and the results are displayed in Figure 6.2. Because a different value of true poverty is provided by each different small area, it is not feasible to plot all the prediction intervals on the same graph. Instead, coverage probabilities for the two bootstrap resampling methods, ordinary bootstrap and cluster bootstrap, are displayed for different levels of clustering in the model.

The first point to note is the severe undercoverage resulting from applying the bootstrap soft estimation methodology to clustered data. For both types of bootstrapping, the cluster bootstrap and simple random bootstrap, undercoverage increases markedly with increasing clustering in the data for all four nominal coverage levels. The effect is greater for the standard, simple random sampling, bootstrap resampling method and suggests a difference between coverage probabilities of the cluster bootstrap and naive bootstrap methods for medium to large intraclass correlation, corresponding to values of k between 0.12 and 0.24. The plots (Figure 6.2) provide graphical evidence to suggest that the cluster bootstrap is indeed necessary for valid estimates when substantial clustering is present in the data.

A statistical test would indicate whether the differences seen in the plot are statistically significant. For each level of clustering in the model, the naive and cluster bootstrap methods are applied to the same datasets, and so are not independent. A Z-test is not feasible in this situation but McNemar's test can be used (Conover 1999).

Figure 6.2: Actual and nominal coverage for different levels of clustering, new small area each simulation, under bootstrapping of clusters, *Cluster*, or bootstrapping of households, *SRS*.



The coverage data was summarised using $\boldsymbol{X}$ and $\boldsymbol{Y}$, where

$$X_i = \begin{cases} 1 & \text{if the } i^{th} \text{ interval contains true P0 for cluster bootstrap} \\ 0 & \text{otherwise} \end{cases}$$

and

$$Y_i = \begin{cases} 1 & \text{if the } i^{th} \text{ interval contains true P0 for ordinary bootstrap} \\ 0 & \text{otherwise} \end{cases}$$

Then McNemar's test was applied to the bivariate pairs $\boldsymbol{X}, \boldsymbol{Y}$. True P0 is the level of poverty of the small area dataset generated for a specific simulation. Table 6.2 displays the p-values of the McNemar tests for different levels of clustering in the data structure, $k = 0$, 0.04, 0.08, 0.12, 0.16, 0.20, 0.24, and 95% nominal coverage level. Results for the three other nominal coverage levels were similar. The p-values shown in Table 6.2 support the evidence of the coverage plots in Figure 6.2, that ordinary bootstrap sampling results in smaller coverage probabilities than the cluster bootstrap, even for a moderate amount of intracluster correlation.

Table 6.2: P-values for McNemar's test of coverage for ordinary bootstrap and cluster bootstrap, 95% nominal level

k	0	0.04	0.08	0.12	0.16	0.20	0.24
p-value	1	0.0633	0.0068	0.0002	0	0	0

Using the cluster bootstrap or naive bootstrap for clustered data has resulted in marked undercoverage, which increased with an increased level of clustering. Table 6.3 displays average standard error of prediction generated by the 1000 Monte Carlo simulations at different levels of clustering. The standard errors are quite small, between 0.0073 and 0.0093, which suggests that the true standard errors of prediction were underestimated, indicating that the undercoverage seen in Figure 6.2 was the result of prediction intervals which were too narrow. For values of the cluster effect $k > 0.1$, corresponding to intracluster correlation of $\rho > 0.05$, the degree of undercoverage was more marked for the simple random sampling bootstrap than for the cluster bootstrap. The cluster bootstrap is capturing some of the clustering in the data structure, since the difference in standard error between naive and cluster bootstrap methods, shown in Table 6.3, increases as the level of clustering increases above 0.1. Although the McNemar's test has confirmed that cluster bootstrap resampling has improved coverage and should be utilised with clustered data, the cluster bootstrap is still producing undercoverage.

The undercoverage seen in Figure 6.2 reflects the lack of a strategy to incorporate cluster effects into the predictions, a necessary step in order that the small area estimates correctly reflect the variability due to clustering in the data. In the ELL methodology, the linear mixed modelling of log expenditure using the survey data generates a set of random effects at both cluster and household levels. When the linear mixed model is applied to census data for small area estimation, bootstrapping is used to incorporate cluster

Table 6.3: Average standard error of predictions for cluster and ordinary bootstrap with small area dataset simulated for each Monte Carlo iteration

Bootstrap method	**Cluster effect**						
	0	0.04	0.08	0.12	0.16	0.20	0.24
Cluster	0.0074	0.0074	0.0076	0.0079	0.0083	0.0087	0.0093
SRS	0.0074	0.0074	0.0074	0.0074	0.0075	0.0076	0.0076

variability and household variability into predictions (Section 2.2.1.12). Each bootstrap estimate of log expenditure for a household in the census, Y_{ij}^b, includes a randomly selected cluster effect, ϵ_{ch}^b from the set of cluster residuals generated at the modelling stage.

$$Y_{ij}^b = \mathbf{x}_{ij}^T \boldsymbol{\beta}^b + \gamma_j^b + \epsilon_{ij}^b \,, \qquad b = 1, ..., B \,, \tag{6.4}$$

The household estimates, Y_{ij}^b, are then aggregated to provide a bootstrap small area estimate. From multiple bootstrap predictions, a bootstrap prediction of poverty and associated standard error at small area level are computed. A similar procedure should be applied when using tree based models for small area estimation of poverty, by introducing cluster effects into the tree predictions. Non-parametric and parametric approaches to incorporating cluster effects into predictions generated from clustered data are discussed in the next sections.

6.4 Introducing cluster effects into predictions

When the data includes cluster effects, the correct bootstrap technique is the cluster bootstrap since it takes the survey design into account. The results of the simulation with cluster effects in the modelling stage, as described in Section 6.3, suggest that the cluster bootstrap provides better coverage than the ordinary bootstrap, but is still under covering with even moderate clustering in the data. Slight undercoverage could be expected, as is seen for $k = 0$, but the severe under performance for typical levels of intracluster correlation is a problem.

Intervals used to examine coverage of small area statistics are perhaps better thought of as prediction rather than confidence intervals, and so should include variability at individual or cluster level, as is done with the ELL methodology (Equation (6.4)), in which each bootstrapped prediction includes a cluster effect randomly selected from a set of model residuals at cluster level. Generally, the bootstrap method is applied to obtain an estimate of a population parameter, which doesn't need to take account of cluster effects. Since the objective here is estimation at small area level, a method is needed to capture cluster variation and include this in the small area estimate: predict at household level and perturb each prediction by some means that encapsulates the cluster variation.

The soft estimate for each household in the k^{th} leaf is the posterior probability for that leaf, $\hat{p}_k$. Let c_j denote the perturbation assigned to each household in the j^{th}

cluster. Then a household in the j^{th} cluster which migrates to the k^{th} leaf would have an adjusted prediction of $\hat{p}^*_{jk} = \hat{p}_k + c_j$. The average of the $\hat{p}^*_{jk}$ across a specific small domain provides a single bootstrap small area estimate of poverty. Averaging these small area estimates over 100 bootstrap iterations would then provide a bootstrap prediction of poverty, and the variability due to clustering effects would be estimated by the standard error of the 100 small area predictions.

In addition to different random effects in each cluster, there are also different fixed effects, the covariates, which are the model predictors. In the process of bootstrapping clusters to obtain different bootstrap samples, some clusters are omitted from the bootstrap sample, so that the influence of the covariates differs for each bootstrap sample. One feature of using multiple predictors in the modelling of poverty incidence is the expectation that a large proportion of the cluster variability is explained by the covariates.

When the small area data was simulated, cluster effects were introduced into the model at Step 2 of the simulation process outlined in Section 5.6, to give adjusted log expenditure values,

$$Y^*_{ij} \;=\; Y_{ij} \;+\; k\, Z_i \,,$$

where Y_{ij} denotes the value of log expenditure for the j^{th} household in the i^{th} cluster, and kZ_i the cluster effect for the i^{th} cluster. The Y^*_{ij} values were then converted to provide a poverty indicator variable, $P0^*_{ij}$,

$$P0^*_{ij} \;=\; \begin{cases} 1 & \text{if } Y^*_{ij} < 8.948423 \\ 0 & \text{otherwise} \end{cases}$$

and $P0^*_{ij}$ used to provide the true level of poverty incidence in the small area data. But the small area dataset used for prediction did not include the poverty response variable, $P0^*_{ij}$, so the influence of the clustering in the data structure was not encapsulated in the predictions. An alternative measure of cluster variability needed to be introduced into the methodology, by incorporating cluster effects into the predicted values. These cluster effects, added to predictions, essentially represent perturbations of the leaves of the tree used for prediction, and are similar to introducing random cluster effects in a linear model, Equation (5.1), to provide a linear mixed model, Equation (6.1). Prediction perturbations for trees can be generated using parametric or nonparametric approaches.

6.5 A non-parametric method for incorporating cluster effects into predictions

To ensure valid estimates of the standard error of prediction, cluster effects need to be incorporated into predictions as well as into the bootstrap procedure. A prediction of $\hat{P}0$ obtained from the classification tree includes the variability explained by the model, and so is satisfactory for providing a point estimate of $P0$, but not for estimating standard error. Actual small areas have small area effects, and the variability due to these small area

effects should be included in predictions. ELL methodology differs from other methods for small area estimation of poverty, in that the model includes random effects at cluster level, rather than small area level (Haslett & Jones 2006). The undercoverage seen the plots in Figure 6.2 occurs because the small area predictions do not include cluster effects.

A non-parametric method used to incorporate clustering effects into predictions involved a modelling approach similar to that used in the ELL methodology, in which a set of cluster effects is created from the cluster levels residuals arising when the model of Equation (6.4) is applied to the survey data (Haslett & Jones 2006). The methodology used to incorporate non-parametric cluster effects into tree predictions developed cluster residuals as the difference between "actual" and estimated poverty for clusters in the survey. The true value of poverty for the j^{th} survey cluster, p_j, was estimated as the proportion of households in the cluster which are classified as being poor. Then a classification tree was built from all the data except the j^{th} survey cluster, and used to provide a prediction, $\hat{p}_j$, for the j^{th} survey cluster. A non-parametric method of deriving a cluster effect for the j^{th} survey cluster is to use $c_j = p_j - \hat{p}_j$. Using this procedure would provide a set of 250 cluster level residuals. At the prediction stage, the small area data provides a soft prediction, $\tilde{p}_k$ for each household which terminates at the k^{th} leaf. Then, for a household from the k^{th} leaf which is also in the j^{th} survey cluster, $\tilde{p}_k$ is augmented by the addition of a randomly chosen cluster residual, c_j from the set of residuals generated from the survey dataset, to provide an adjusted prediction, $\tilde{p}^*_{jk}$ which takes account of clustering in the data, as follows

$$\tilde{p}^*_{jk} = \tilde{p}_k + c_j \,. \tag{6.5}$$

The method outlined above assumes that these cluster level residuals are identically and independently distributed, so that the cluster level residual, $c_j = p_j - \hat{p}_j$, the difference between actual and estimated poverty, is independent of actual poverty. This is reasonable in the ELL context which assumes a normal distribution and constant variance. However, for cluster residuals, c_j derived from the tree, this assumption doesn't hold, since we're dealing with proportions which have a Binomial distribution, so an adjustment is needed. The usual adjustment for heteroscedasticity in a Binomial distribution is to standardise the variability, which results in a cluster residual, c^*_j, such that

$$c^*_j = \frac{c_j}{\sqrt{p_j\,(1-p_j)}} \,,$$

which is similar in form to the Pearson residual. A back transformation can then be applied to obtain a cluster residual, $\tilde{c}_j$, on the same scale as $\tilde{p}_k$, the poverty estimate for each household in the j^{th} cluster which ends up in the k^{th} leaf,

$$\tilde{c}_j = c^*_j \sqrt{\tilde{p}_k\,(1-\tilde{p}_k)} \,.$$

The small area prediction, $\tilde{p}^*_{jk}$, adjusted to incorporate clustering effects is then,

$$\tilde{p}^*_{jk} \;=\; \tilde{p}_k \;+\; \tilde{c}_j \;.$$

However, this modification may not ensure that the adjusted prediction of poverty remains within the interval $(0, 1)$. A solution to this problem is to instead take differences on the logistic scale, as follows

$$c^*_j \;=\; \text{logit}\,(p_j) \;-\; \text{logit}\,(\hat{p}_j) \;. \tag{6.6}$$

Then, at the point in the modelling process when the simulated small area is used to provide predictions, a cluster effect derived on the logistic scale, c^*_j, is randomly selected from the set of cluster residuals generated from the simulated survey data, and added to the logit of the prediction, $\tilde{p}_k$, for a census household which is directed to the k^{th} leaf and which originates from the j^{th} census cluster, to provide an intermediary value, $\tilde{p}^{\#}_{jk}$, such that,

$$\tilde{p}^{\#}_{jk} \;=\; \text{logit}\,(\tilde{p}_k) + c^*_j \;.$$

The inverse logit transformation of $\tilde{p}^{\#}_{jk}$,

$$\tilde{p}^*_{jk} \;=\; \text{logit}^{-1}\left(\tilde{p}^{\#}_{jk}\right) \;, \tag{6.7}$$

is then used as the amended prediction, $\tilde{p}^*_{jk}$, incorporating the clustering present in the data, for the a household in the j^{th} cluster which migrates to the k^{th} leaf. However, another problem arises when the actual cluster proportion of poverty is one of the boundary values, 0 or 1. The small size of the clusters, twelve households each, can accentuate this problem. The logit function has the form,

$$\text{logit}(p) \;=\; \log\left(\frac{p}{1 \;-\; p}\right) \;,$$

and so $\text{logit}(0) = \log(0)$ which is undefined, while $\text{logit}(1) = \log(\infty)$, also undefined, thus for the methodology to work it must eschew logits of 0 or 1. To avoid the boundary problem requires a correction factor to be added to the actual poverty level, p_j, for the j^{th} cluster of the survey data. Consider the probability correction formula,

$$p_j \;=\; \frac{x \;+\; 0.5}{n \;+\; 1} \;, \tag{6.8}$$

where $n = 12$ is the cluster size for the survey, and x the number of households in the cluster designated as being poor. The effect of Equation (6.8) is to shrink the proportion of poor in each survey household towards 0.5. The actual poverty level of all survey clusters is corrected, not just those clusters with the extreme values of 0 and 1. Equation (6.8) is equivalent to using Jeffrey's prior in the Bayesian context for the Binomial

distribution (Lee 2012). The probability correction formula can be generalised to,

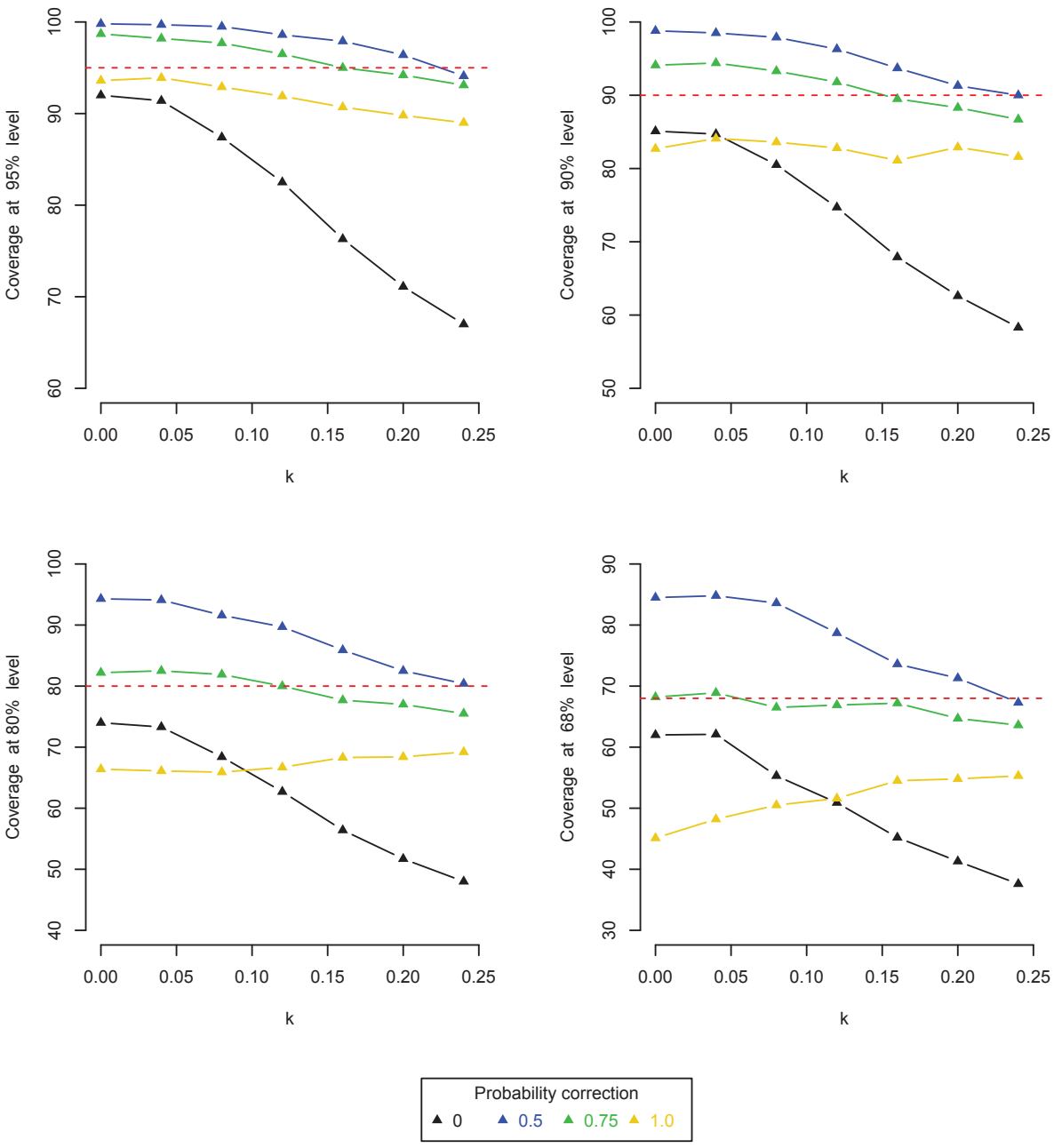
$$p_j = \frac{x + 0.5\,a}{n + a}\,. \tag{6.9}$$

The non-parametric method of perturbing tree predictions, as described above, was incorporated into Step 5 of the Monte Carlo simulation process outlined in Section 5.6. Values of $a = 1.0$, 0.75, and 0.5 were used in the correction formula, Equation (6.9). The value $a = 0.75$ was included as research has suggested that this value is optimal in terms of constant variability for other transformations of a Binomial variable (Anscombe 1948). To examine the effect of appending smaller sized cluster residuals to the predictions, a value of $a = 0.5$ was also used in the probability correction formula. Adjustment of the proportion of poor in a cluster, to avoid logit values of 0 and 1, was only necessary for the actual poverty status of a cluster, p_j. A prediction of poverty for the j^{th} cluster, $\hat{p}_j$, computed as the mean of posterior probabilities for all households in the j^{th} cluster, is unlikely to take the value 0 or 1. This would happen only if all the households in a cluster ended up in pure leaves, having posterior probability of either 0 or 1, which occurs when all observations in the leaf have identical poverty status. Such a situation did not arise, since the degree of pruning in the tree ensured that the none of the terminal nodes of the trees built were pure. The results of the modelling with non-parametric cluster effects incorporated into the predictions are discussed in the next section.

6.5.1 Results of modelling with non-parametric cluster effects in predictions

Monte Carlo simulations were run for values of $k = 0$, 0.04, 0.08, 0.12, 0.16, 0.20, and 0.24. For each of the 1000 Monte Carlo simulation for a particular value of k, a bootstrap point estimate of poverty incidence and bootstrap standard error were computed. Using these statistics, three different types of prediction intervals were constructed for nominal coverage values of $95\%, 90\%, 80\%$ and 68%. Standard parametric intervals having the form *centre* $\pm\, Z\times$ *standard error* were built using the bootstrap soft estimate of poverty as the interval centre, with the bootstrap estimate of standard error contributing to interval width. Bootstrap percentile intervals were also constructed for the four nominal values. For example, the 97.5^{th} percentile and 2.5^{th} percentile provided the upper and lower interval limits for nominal 95% confidence level. A second type of parametric interval was formed which was centred about the bootstrap estimate of poverty from the tree model built from the full simulated survey dataset, rather than the mean of the 100 bootstrap estimates. For both parametric type intervals, the interval width was determined using the bootstrap estimates of standard error and the critical value associated with the nominal confidence level. Figure 6.3 displays the coverage properties of the parametric interval with mean of bootstrap soft estimates as centre. Each solid line on the graphs represents empirical coverage for a specific size of the probability correction factor, a, across different values of k, the amount of clustering built into the data structure. The red dashed line indicates the nominal coverage level.

Figure 6.3: Empirical coverage for non-parametric cluster effects in predictions using bootstrap soft estimate as interval centre, and shrinking all actual survey cluster proportions to 0.5. Probability correction of 0 denotes no cluster effects in predictions.



The first impression gained from the plots in Figure 6.3 is that incorporating cluster residuals into the predictions has tended to increase actual coverage probabilities for all four nominal confidence levels, when compared with empirical coverage when clustering is included only in the model (Figure 6.2). This impression is confirmed by examining the standard errors of prediction produced by the two methods. With cluster effects incorporated only into the model, but not the predictions, standard errors were in the range of 0.007 to 0.009. When cluster effects were incorporated into predictions as well as the data structure, standard errors increased two fold, being in the range 0.013 to 0.018. In general, for intervals with the mean of the bootstrap soft estimates as interval centre, empirical coverage decreases with increasing values of k, representing cluster effects in the data structure, the exception being for probability correction factor $a = 1.0$ at the 80% and 68% nominal levels. Coverage also reduces with increasing values of the probability correction factor a. The pattern revealed is some overcoverage for $a = 0.5$, reasonable coverage for $a = 0.75$ and undercoverage for $a = 1.0$. This suggests that the use of probability correction factor $a = 0.75$ has produced results in accord with the paper by Anscombe (1948). However, examination of bias in the predictions, and whether the non-coverage is due to the interval being above or below the true poverty level tells a different story.

Bias is measured as the difference between the estimate of poverty obtained through the bootstrap soft estimation procedure, $\hat{P}0$, and the "true" poverty level, $P0$, the measure of poverty incidence in the simulated small area dataset. Incorporating cluster effects into the predictions as well as into the data structure has resulted in positive bias being present in all estimates. This bias increased with increasing values of the probability correction factor a, and reduced as cluster effects in the data structure, k, got larger. In contrast, the standard errors of prediction reduced with increasing values of a and increased for larger values of k.

Coverage is measured as the proportion of intervals which include the true parameter value, $P0$. The great majority of intervals centred about the bootstrap soft estimate which do not include $P0$ lie above the true poverty level. Thus, the distribution of empirical prediction intervals is skewed above the true value of poverty. Consequently, intervals based on probability correction of $a = 0.5$ show overcoverage because the bias in the resulting predictions is offset by larger standard error. With $a = 0.75$, bias and standard error "balance out", resulting in the associated prediction intervals having reasonable coverage. Undercoverage when $a = 1.0$ is mainly due to high standard error values, which, in conjunction with the bias, results in more intervals lying above the true poverty level.

Another type of interval examined was a parametric interval centred around the estimate of poverty extracted from the classification tree built from the complete simulated survey dataset, the "full" tree, instead of a bootstrap subsample. Empirical coverage, produced by this type of interval, for the four nominal coverage levels is displayed in Figure 6.4, which indicates a tendency for actual coverage to decrease with increasing values of k, representing levels of clustering in the data structure. But, there is very little difference in empirical coverage for differing values of the probability correction factor a.

For this particular parametric prediction interval, overcoverage is a problem for all but the very highest levels of clustering in the data structure and predictions, and the problem is worse for the smaller nominal coverage levels. The intervals which do not provide coverage are fairly evenly spread above and below the true poverty value. The reason for overcoverage here is that the standard errors of prediction are too large.

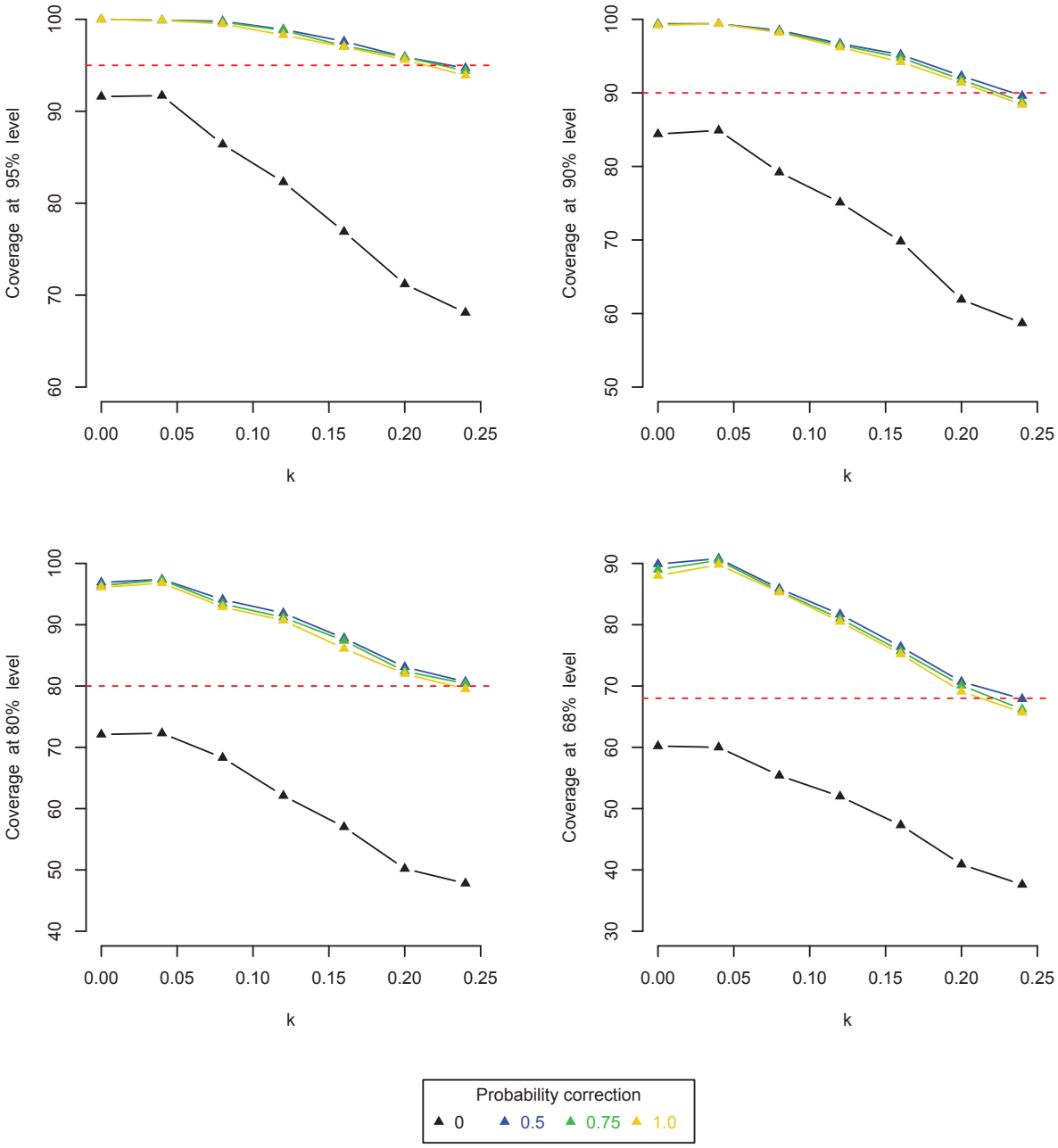
Empirical coverage for percentile bootstrap intervals were also examined, and were found to demonstrate patterns of coverage very similar to those in Figure 6.3, produced by the parametric interval centred about the bootstrap soft estimate of poverty incidence.

The evidence from Figures 6.3 and 6.4 indicates that we do not have satisfactory empirical coverage for parametric intervals having the soft bootstrap estimate or the full tree estimate as the interval centre, or for bootstrap percentile intervals. In an attempt to to reduce the variability in the predictions, the method of deriving prediction perturbations by shrinking all actual cluster proportions to 0.5 was amended. One modification was to apply the probability correction only to extreme cluster probabilities, in which cluster proportions were either 0 or 1. A second adjustment shrank the probabilities to 0.2 rather than 0.5. The rationale for the latter move was to better reflect the actual level of poverty in the simulated data, which was approximately 0.2. The adjusted probability correction formula then became

$$p_j \;=\; \frac{x \;+\; 0.2\,a}{n \;+\; a}\,,$$

with values of the correction factor, $a = 0.5$, 0.75 and 1.0, being retained in the modelling. Neither amendment provided any improvement. Applying the probability correction only to extreme cluster proportions and/or shrinking these probabilities to 0.2 resulted in substantial overcoverage, no matter what type of prediction interval was built. In conclusion, the methodology devised to generate cluster residuals non-parametrically has not proved to be a valid method for incorporating cluster effects into predictions. This technique was chosen because it is analogous to the method used in ELL to incorporate cluster effects into predictions (Equation (6.4)), but it resulted in standard errors of prediction which were too large, producing overcoverage. Even if there were no cluster effects in the data, the non-parametric method would "detect" estimated cluster effects, particularly since the clusters are small, having are only twelve households per cluster. The cluster effect c_j^* defined in Equation (6.6) is unlikely to be zero, since $\hat{p}_j$ is obtained from the data and $\hat{p}_j$ is estimated by the tree. So c_j^* is capturing all unexplained variation, variability at cluster level and household level. A method is required which will separate out the household level variability from the cluster level variability. The next section discusses a parametric approach to the task of incorporating cluster effects into prediction.

Figure 6.4: Empirical coverage for non-parametric prediction perturbations, full tree estimate as interval centre, shrinking only extreme survey cluster proportions to 0.5. Probability correction of 0 denotes no cluster effects in predictions.



6.6 A parametric method for incorporating cluster effects into predictions

Incorporating cluster effects into the prediction stage of the process corresponds to perturbing the leaves of the tree which generates the predictions of poverty. A non-parametric approach to construct these perturbations, as described in Section 6.5, selected a cluster level effect from a set of cluster residuals which was then appended to the prediction for each household. The set of cluster level residuals comprised the difference on the logit scale of actual and estimated poverty levels for each cluster in the simulated survey. This method to incorporate cluster effects into the predictions did not provide a valid technique, since it overestimated the cluster effects and produced overcoverage. An alternate approach is a parametric technique which assumes a normal distribution, $N(0, \sigma_c^2)$, for the logit transformation of the cluster perturbations, c_j as defined in Equation (6.5). Sampling from this distribution then would provide the perturbations to be added to predictions.

The task was to find an appropriate method to estimate σ_c, cluster level variability, utilising the structure of the classification tree. The approach taken was to model the probability of being poor for each household based upon the terminal node, or leaf, for that household, and include a term to capture cluster variability. The binary response variable for the model was $Y_{jk} \sim \text{Bernoulli}\,(\pi_{jk})$, where Y_{jk} is the true designation as "poor" or "not poor" for the household in the j^{th} cluster which has the k^{th} leaf as its final destination, and π_{jk} denotes the probability of being poor. There may be more than one household in a cluster having the same terminal node.

The classical linear mixed model, as defined in Equation (6.1), assumes a normal distribution for the response variable (Chambers & Hastie 1992). A generalised linear mixed model with logistic link function takes account of the Bernoulli distribution of Y_{jk}, to give a model for the π_{jk}'s of,

$$\begin{aligned} \text{logit}\,(\pi_{jk}) &= \eta_{jk} \\ &= \alpha_k + c_j\,, \end{aligned} \tag{6.10}$$

where

$$\text{logit}\,(\pi_{jk}) = \log\left(\frac{\pi_{jk}}{1-\pi_{jk}}\right)\,,$$

and the π_{jk}'s are modelled using the monotonic logit function which links π_{jk}, as the mean of the response variable, to a linear predictor, η_{jk}, comprising the explanatory variables. All coefficients in the linear predictor, η_{jk}, are on the logit scale. Fixed effects are provided by each α_k term, which represents the posterior probability of the k^{th} leaf, p_k, on the logit scale. The c_j's represent the variability in the data due to cluster effects, and contribute to the total variability in the probabilities of being poor. The cluster effects, c_j, are assumed to be normally distributed, $c_j \sim \text{N}\left(0,\, \sigma_c^2\right)$, where σ_c^2 is estimated from the generalised linear model, Equation (6.10). Thus, the cluster factor is random effect since the cluster effects in the survey are assumed to be a random sample from a population of cluster

effects. This type of model is known as a *random intercept model* (Agresti 2013).

Modelling was carried out using the *glmer* function found in the statistical package *lme4* (Bates, D. and Maechler, M. and Bolker, B. and Walker S. 2013), available through the R software environment (R Core Team 2015). The model was fitted using the Laplace approximation for maximum likelihood estimation. The estimated variance of the random cluster effect obtained from fitting the model provided an estimate of σ_c^2. Thus, parametric cluster effects could be incorporated into predictions by selecting a value from the normal distribution, $\mathrm{N}\left(0,\, \sigma_c^2\right)$, to append to the logit of $\tilde{p}_k$, the prediction of poverty generated by the tree model for the a household from the k^{th} leaf. Since c_j is already on the logistic scale it can be added directly to the logit of $\tilde{p}_k$, as follows,

$$\tilde{p}_{jk}^{\#} \;=\; \mathrm{logit}\left(\tilde{p}_k\right) + c_j \;, \tag{6.11}$$

and taking the inverse logit of $\tilde{p}_{jk}^{\#}$ produces,

$$\tilde{p}_{jk}^{*} \;=\; \mathrm{logit}^{-1}\left(\tilde{p}_{jk}^{\#}\right) \;. \tag{6.12}$$

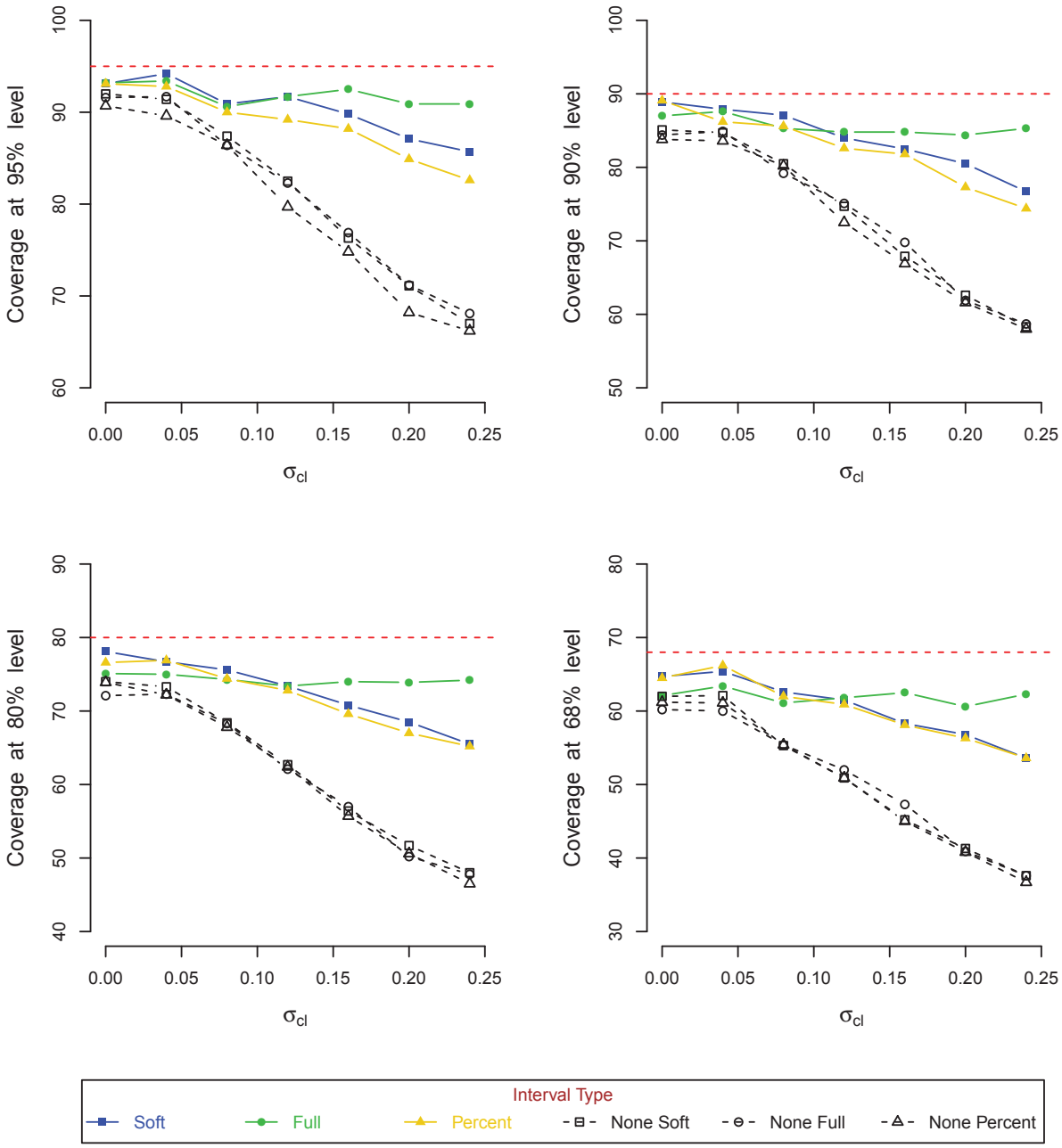
Then, $\tilde{p}_{jk}^{*}$ provides the amended prediction of poverty incidence for a household in the j^{th} census cluster which migrated to the k^{th} leaf of the tree. The results of incorporating parametric cluster effects into the modelling are discussed in the next section.

6.6.1 Results of modelling with parametric clusters effects in predictions

The Monte Carlo exercise used to incorporate non-parametric cluster effects in the Nepal poverty data, as outlined in Section 6.5, was amended for parametric cluster effects, using the code in Appendix C.2. Instead of estimating non-parametric cluster effects as the difference between actual and estimated poverty levels for each cluster in the simulated survey, parametric cluster effects were obtained directly as described above and incorporated into Step 5 of the simulation process outlined in Section 5.6.

Monte Carlo simulations to investigate the coverage properties of the parametric technique for incorporating cluster effects into predictions were run for various levels of clustering in the data structure, k = 0, 0.04, 0.08, 0.12, 0.16, 0.20, 0.24. The plots in Figure 6.5 display empirical coverage for three different types of interval, as described in Section 6.5.1. The blue lines in the plots indicate coverage for parametric prediction intervals with the bootstrap soft estimate as centre; green lines represent the parametric intervals centred about the full tree estimates; the gold lines relate to bootstrap percentile intervals. For completeness, also displayed on the graphs are black lines representing empirical coverage for the same three interval types, based on simulated data which did not include prediction perturbations but only clustering in the model. Clearly, when the data is clustered, then cluster effects need to be incorporated into the predictions to generate valid standard errors of prediction.

Figure 6.5: Empirical coverage for parametric prediction cluster effects, for three types of intervals: Soft - centred about soft bootstrap estimate; Full - centred about full tree estimate; Percent - bootstrap percentile; None - no cluster effects in predictions



The plots in Figure 6.5 indicate no overcoverage, for each type of prediction interval built, when parametric perturbations are incorporated into the predictions. Empirical coverage for the parametric interval centred about the poverty estimate generated using the full tree model is fairly consistent for all levels of clustering, but parametric intervals centred about the mean of bootstrap soft estimates and the non-parametric percentile bootstrap intervals show a decreasing trend in coverage. However, for low to moderate amounts of clustering in the data, values of $k < 0.15$, empirical coverage is very similar for all three interval types. The parametric interval centred about the "full" tree estimate had a symmetric distribution of non-covering intervals, whereas bootstrap percentile intervals and those centred about the mean of the bootstrap estimates were skewed below, i.e. more intervals lying below than above the true poverty level. Coverage error, the difference between empirical and nominal coverage, is acceptably small (Ólafsdóttir & Mudelsee 2014), being within a few points of the nominal value, for all but the highest amounts of clustering.

These results are encouraging, particularly since the two parametric intervals were based upon the normal distribution, whereas the parameter of interest is not likely to have a sampling distribution which is normal (Stangenhaus & Narula 1991). Achieving nominal coverage at every level of clustering would only occur if the sampling distribution of poverty incidence was perfectly normal.

Standard errors of prediction increased with increasing levels of clustering in the data structure, k, ranging from about 0.008 to 0.016, but are of magnitude which is less than the standard errors produced by the non-parametric method (Section 6.5.1). Bias of the full tree estimate was less than 0.0002 for all values of k, but average bias for the bootstrap estimates, although small for $k = 0$, at 0.0007, increased with increasing k, to about 0.01 for maximum clustering. The evidence of the coverage plots and patterns of bias and standard error from the simulation exercise indicate that all three methods are useful to generate small area estimates of poverty incidence for small to moderate amounts of clustering in the data structure, but that the best approach with highly clustered data is to use the full survey tree model to provide the point estimate for poverty incidence and employ bootstrap resampling to estimate the standard errors of prediction.

The Monte Carlo exercise to this point has not addressed the importance of taking account of the complex survey design component of stratification in the data for small area estimation of poverty. Extension of the methodology to include stratification is obvious, and will be illustrated using the actual Nepal data. The rest of the chapter focuses on modifying the methodology developed and tested so far to include stratification, applying it to the Nepal data to obtain small area estimates of poverty incidence for a particular district in Nepal using a classification tree model, and comparing these estimates with published results obtained using the ELL method (Haslett & Jones 2006).

6.7 Classification tree modelling for small area estimation in Nepal

Section 6.6 described a parametric method for incorporating cluster effects into bootstrap soft predictions of poverty extracted from a classification tree model. Results of the Monte Carlo simulations to test the validity of this method, as a means of adapting a classification tree for complex survey data with a clustering component, were outlined in Section 6.6.1. The results indicated that using the bootstrap soft estimation method has provided a valid method for small area estimation of poverty based on a classification tree. Using the Nepal data as an illustration, the next step is to extend the methodology to also take into account stratification in the survey design.

The 2003/4 Nepal Living Standards Survey comprises a two-stage stratified sampling design (Haslett & Jones 2006). At the first stage, clusters, the primary sampling units (psu's), were selected with probability proportional to size independently within six strata. Then, within each chosen cluster twelve households were selected using systematic sampling. The discussion so far in this chapter has covered bootstrap variance estimation for simulated data with cluster effects. Modelling is now extended to complex sampling design which includes stratification, and applied to the Nepal dataset.

Section 2.2.3.6 discussed the development of cluster bootstrap sampling for survey data which also comprises stratification. The usual practice is to select bootstrap samples independently from each stratum, with the size of each bootstrap sample being one less than the number of clusters in the particular stratum from which it is drawn. The bootstrap estimator should then be constructed so as to replicate the original probability sampling design in each stratum (Rao & Wu 1988). However, Shao (2003) asserts that reproducing the complex structure of the parent sample within each stratum is difficult to achieve especially for small stratum size, n_h. He suggests a straightforward application of the bootstrap by drawing independent bootstrap samples from each stratum, and combining these into a single bootstrap replicate, from which the bootstrap estimate, $\hat{\theta}_b^*$ is computed.

In the context of poverty mapping, Elbers et al. (2003) recommend fitting a separate regression model for each stratum in the survey design to provide stratum predictions. This approach, however, can risk overfitting of the model, especially if small strata occur in the sampling scheme (Haslett & Jones 2006). Since the ELL technique models poverty based on household level and area level predictors, a model for each stratum may not be needed, as stratum differences can be captured by the explanatory variables, and interactions with stratum can be included, where required. Using a single model for all strata results in more stable estimates. In the ELL modelling of poverty incidence in Nepal (Haslett & Jones 2006), interaction terms allowed for differing effects of some predictors across some *groups*. These groups were constructed from the survey strata to be fairly homogeneous, for the purpose of building regional price indices (Haslett & Jones 2006). A comparison of the composition of these groups and the original survey strata

is provided in Table 6.4. All the groups are either urban or rural, and the strata, except Urban Kathmandu, are also exclusively rural or urban.

Table 6.4: Comparing the composition of strata and groups in the Nepal modelling

Strata	**Groups**
Urban Kathmandu	Urban Kathmandu
Mountains	Other Urban
Other Urban Hills	Rural Western Mountains and Hills
Rural Hills	Rural Eastern Mountains and Hills
Urban Terai	Rural Western Terai
Rural Terai	Rural Eastern Terai

Trees are more unstable than regression models, so employing separate bootstrap samples to build a tree for each stratum is not useful. The most efficient approach is to fit a single tree for the whole population. Thus, the procedure used for bootstrap variance estimation with trees, under the multistage stratified sampling design of the Nepal data, was to draw independent bootstrap samples from each stratum, and combine these stratum samples into a single bootstrap replicate from which a tree was built. Bootstrap sample sizes were set at $n_h^* = n_h - 1$ for n_h the number of clusters in the h^{th} stratum, and weights for the secondary sampling units, households in clusters, scaled up so that the bootstrap weights for the bootstrap sample drawn from a particular stratum summed to the total weight of that stratum. A categorical predictor representing the six groups specified in Table 6.4 was available as a covariate in building the tree so that differences between groups could be incorporated into the model. This is analogous to the use of stratum interactions in Haslett & Jones's (2006) implementation of ELL in Nepal. The binary partitioning algorithm used to build the tree model automatically fits interactions, conditional on the previous splits. Since the group variable was not a significant splitter, then groups differences appear to be captured by other explanatory variables. The methodology for applying the bootstrap variance estimation to the Nepal dataset is discussed in the next sections.

6.7.1 Setting up the analysis

The modelling process for applying the bootstrap soft estimation method to the Nepal data was,

1. Draw a bootstrap sample independently from each stratum, of size n_h - 1 in the h^{th} stratum, where n_h is the number of clusters
2. Generate bootstrap weights for all households in each bootstrapped cluster
3. Combine the six bootstrap samples to form a single bootstrap replicate, with associated bootstrap weights

4. Build a weighted classification tree model using the bootstrap replicate and weights from Step 3, using $cp = 0.001$ as pruning criterion

5. Estimate the cluster variance, σ_c^2, as described in Section 6.6

6. Apply the model built in Step 4 to generate predictions $\tilde{p}_k$, for each household in the k^{th} leaf

7. For each houshold in the j^{th} cluster in the district of Nepal, add a cluster effect, $c_j \sim \mathrm{N}\left(0,\, \sigma_c^2\right)$, to logit$(\tilde{p}_k)$ in Step 6, to get $\tilde{p}_{jk}^{\#}$ as described in Equation (6.11); each household in the j^{th} cluster is assigned the same cluster effect c_j

8. Take the inverse logit of $\tilde{p}_{jk}^{\#}$ to give an amended poverty prediction $\tilde{p}_{jk}^{*}$, which incorporates cluster effects, as described in Equation (6.12)

9. Aggregate the $\tilde{p}_{jk}^{*}$'s across the 18 ilakas in the district to provide a bootstrap prediction of poverty incidence for each ilaka

10. Repeat steps 1 to 9 for 100 bootstrap replicates to provide 100 bootstrap predictions of poverty incidence for each ilaka

11. The mean and standard deviation of the 100 bootstrap predictions in Step 10 provide a bootstrap point estimate of poverty incidence, $\hat{\theta}_i$ with associated standard error, $\hat{\sigma}_i$, for each ilaka

12. Apply the cluster bootstrap to the full Nepal dataset, build a tree and generate point estimates of poverty which do not include prediction perturbations

A stopping criterion of $cp = 0.001$ was used in model building for the actual Nepal data, Step 4 in the above process, rather than the procedure applied in the Monte Carlo simulations, in which cp was set at zero, and the tree pruned by specifying a maximum tree size of 5. The reason for utilising $cp = 0.001$ as the stopping rule was to provide a comparison between the estimates discussed in Section 3.3.2, which were obtained from a tree which incorporated only the sampling weights, and the estimates generated using the process described above, from a model which took account of all the components in the Nepal survey design: clustering, stratification and design weights.

To incorporate stratification into the estimation process, information related to the stratification structure in the Nepal data needed to be extracted and included in the modelling. A variable identifying each of the strata was constructed, and the size of each stratum, in terms of the number of clusters in the stratum, extracted from this variable. The stratum weight, i.e. total household weight for each stratum, was also computed, to be used in constructing bootstrap weights for the weighted classification tree.

The usual procedure for reweighting observations when the cluster bootstrap is applied within strata is (Rao & Wu 1988)

$$w_{hij}^{*} \;=\; w_{hij}\,\frac{n_h}{n_h - 1} \tag{6.13}$$

where w^*_{hij} denotes the bootstrap weight, w_{hij} is the sampling weight of y_{hij}, the i^{th} household in the j^{th} cluster of the h^{th} stratum and n_h the size of the h^{th} stratum in terms of the number of clusters it contains. Instead of using Equation (6.13) to define the weights to be used in building the tree, bootstrap weights, w^*_{hij}, for a bootstrap sample drawn from a given stratum were constructed so that the sum of the sampling weights for all households in the bootstrap sample equaled the total weight of all units in that stratum, as follows,

$$w^*_{hij} \;=\; w_{hij}\,\frac{\sum\limits_{stratum} w_{hij}}{\sum\limits_{BSsample} w_{hij}} \tag{6.14}$$

Table 6.5 records the size, i.e. number of clusters in the stratum, and total household weight for each of the six strata in the Nepal survey dataset.

Table 6.5: Size and total sampling weights for each stratum in the Nepal analysis

Stratum	Mountains	Rural Hills	Rural Tarai	Urban Hills	Urban Kathmandu	Urban Tarai
Size	32.0	96.0	102.0	28.0	34.0	34.0
Total weight	1568287.4	7429909.1	9807749.9	690268.2	1182755.4	1405184.1

6.7.2 Results for classification tree small area estimation in Nepal

Using the methodology described in Section 6.7.1, small area estimates of poverty incidence, *P0*, and associated standard error of prediction, *se*, for eighteen ilakas in a district of Nepal were obtained from the classification tree model, and compared with the published ELL results from Nepal (Haslett & Jones 2006), as shown in Table 6.6. The column labelled *BS soft P0* in Table 6.6 represents point estimates of poverty incidence for each ilaka computed as the mean of 100 bootstrap estimates. *Full tree P0* indicates point estimates of poverty incidence obtained from the tree model based upon all of the Nepal survey data. The heading *BS se* refers to the bootstrap estimate of the standard error of prediction, calculated as the standard deviation of the 100 bootstrap small area predictions of poverty incidence for the ilaka.

Table 6.6 indicates that standard errors of prediction from the classification tree model are in the same order of magnitude as the ELL standard errors, but are slightly larger than the ELL estimates. This is expected because the estimates of standard error from the tree model incorporate model uncertainty. Each new bootstrap sample produces a different model, a different tree based on a different selection of covariates, and the variability resulting from these different models is captured by the bootstrap estimate of standard error. In contrast, the ELL method uses one fixed model for predictions, based on a fixed set of covariates, and bootstraps all sources of variability to provide an estimate of standard error: variability in the regression coefficients, at cluster level and at household level (Section 2.3.3).

Table 6.6: Comparison of ELL and bootstrap soft tree estimates for an ilaka in one district of Nepal

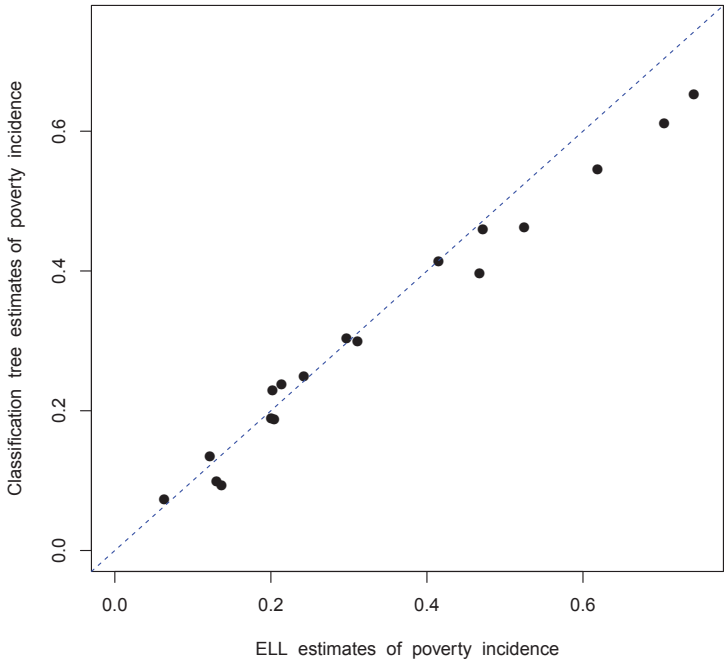
Ilaka	**ELL estimates**		**Tree estimates**		
	P0	se	Full Tree P0	BS soft P0	BS se
1	0.742	0.030	0.655	0.653	0.058
2	0.525	0.033	0.469	0.463	0.045
3	0.619	0.035	0.527	0.545	0.061
4	0.468	0.024	0.411	0.397	0.042
5	0.297	0.028	0.324	0.304	0.046
6	0.472	0.027	0.443	0.460	0.035
7	0.704	0.036	0.601	0.611	0.065
8	0.204	0.031	0.185	0.188	0.047
9	0.201	0.028	0.198	0.190	0.057
10	0.122	0.024	0.105	0.135	0.035
11	0.242	0.020	0.222	0.249	0.034
12	0.202	0.027	0.245	0.229	0.035
13	0.213	0.020	0.233	0.238	0.033
14	0.415	0.031	0.399	0.414	0.038
15	0.311	0.026	0.297	0.300	0.033
16	0.063	0.021	0.066	0.073	0.028
17	0.130	0.040	0.069	0.099	0.046
18	0.137	0.034	0.088	0.094	0.043

The Monte Carlo simulation study in which parametric cluster effects were incorporated into predictions, Section 6.6.1, suggested that point estimates of poverty incidence from the full tree were a better choice than point estimates obtained as the mean of bootstrap estimates of poverty incidence. However, the simulation study used a smaller tree, which was pruned to a maximum depth of five, whereas the process of estimating estimates from the Nepal district data employed $cp = 0.001$ as the stopping rule in Step 4 of the tree building stage of the process outlined in section 6.7.1, resulting in a larger tree. The rationale for choosing this pruning method for modelling of the Nepal data was to enable meaningful comparison with the small area estimates generated in Section 3.3.2 using *cp* values of 0.005 and 0.001 as pruning criteria. A consequence of using a larger tree is that the point estimates of poverty incidence computed as the mean of bootstrap estimates are now very similar to the points estimates from the full tree.

Comparison of the point estimates generated by both types of model, ELL and the classification tree, listed in Table 6.6, reveals that the classification tree point estimates of poverty incidence, *P0*, whether generated from the full tree model or as the mean of bootstrap soft estimates, are consistently lower than the ELL point estimates for the higher poverty rates, $P0 > 0.4$. This pattern is more clearly illustrated in Figure 6.6, which displays point estimates from the classification tree, obtained by taking the mean of a hundred bootstrap estimates, versus the ELL method. The plot in Figure 6.6 shows that, for higher poverty rates, the points representing tree versus ELL values drop below

the diagonal $y = x$, the blue dashed line on the plot, representing equal values for ELL and tree estimates. Since there is no "gold standard" for poverty estimates in Nepal, it is difficult to determine which of the two methodologies provides the most accurate results.

Figure 6.6: Plot of classification tree estimates versus ELL estimates for 18 ilakas in a district of Nepal



The points plotted in Figure 3.14 represent estimates based on a weighted tree model which incorporated elements of survey design only through the household sampling weights. The small area estimates graphed in Figure 6.6 were generated using a model which took into account all elements of the survey design, by incorporating cluster effects into predictions and applying the cluster bootstrap within each stratum, as well as including sampling weights into the tree. Comparison of Figure 6.6 with the bottom right graph in Figure 3.14, suggests that fully incorporating the survey design into the modelling has, as expected, produced classification tree estimates which are more similar to their ELL equivalents. The tree estimates are closer to the corresponding ELL estimates for the higher poverty rates, and more consistent, in that the distance below the diagonal $y = x$ is similar for poverty levels > 0.5.

6.8 Conclusion

The purpose of the Monte Carlo simulation exercise outlined in this chapter was to find a method with which to generate reasonable standard errors of prediction from a classification tree model, when the data structure included clustering and stratification. In Chapter 5, the technique of bootstrap soft estimation for poverty estimates was found to

produce reasonable standard errors of prediction for poverty incidence when the data had been collected through simple random sampling. The bootstrap soft estimation technique developed in Chapter 5 utilised soft tree estimates, the posterior probability of being poor, to generate predictions, and bootstrap resampling for variance estimation. Extension in this chapter of the simulation process outlined in Section 5.6 to complex survey data required incorporating clustering and stratification into the modelling process.

Monte Carlo modelling described in this chapter comprised 1000 simulations, generating 1000 survey and small area datasets. Clustering was introduced into the data structure of both simulated survey and small area datasets. From each simulated survey a classification tree model was built, which was then applied to the small area data to obtain small area estimates of poverty incidence. Standard errors were calculated using the cluster bootstrap and incorporating residual cluster effects into the small area predictions. Non-parametric and parametric methods were employed to generate the cluster residuals.

The purpose of the Monte Carlo study was to gauge the validity of bootstrap soft estimation for complex survey data, by testing whether each method for incorporating cluster effects into predictions, non-parametric or parametric, produced reasonable standard errors of prediction. A thousand prediction intervals based on estimates of poverty incidence and associated standard errors were generated for each method used and their coverage properties examined. A reasonable amount of empirical coverage for a nominal coverage level provided support for the usefulness of a particular technique for adapting classification trees for clustered data.

Monte Carlo simulation indicated that only applying the cluster bootstrap, with no further modification, was not a sufficient adaptation of the classification tree model when clustering is present in the survey data (Section 6.3). The McNemar's test described in Section 6.3.2 demonstrated that the cluster bootstrap is superior to the ordinary bootstrap sampling method when the data comprises clustering. Despite this, applying the cluster bootstrap method without also incorporating cluster effects into predictions resulted in severe undercoverage, especially for large amounts of clustering in the data. Non-parametric modelling of residual cluster effects, generated from the survey and incorporated into predictions, was found to be an unsuitable procedure for classification tree models when clustering is present in the data (Section 6.5). Three types of prediction intervals were investigated to test this methodology: parametric intervals centred about the mean of the soft bootstrap estimates and the estimate from the full tree respectively; a non-parametric percentile bootstrap interval. None of these interval types proved adequate since standard errors of prediction were too large. This was because the method used to incorporate non-parametric perturbations into predictions had the effect of inflating the bootstrap estimates too much. The point estimates based on the full tree were unbiased, but the the mean of the soft bootstrap estimates was positively biased.

The parametric method for incorporating cluster effects into predictions, outlined in Section 6.6, was more successful. Standard errors of prediction were reasonable, since the three interval types demonstrated similar coverage for small to medium amounts of

clustering in the data, only a few points below the nominal values. The fact that actual coverage did not not quite reach nominal coverage is not of great concern, since achieving nominal coverage at every level of clustering would occur only if the sampling distribution was perfectly normal, which is not the case for statistics representing probabilities.

When the methodology developed to deal with clustered data was applied to the actual Nepal data and extended to include stratification, the resulting point estimates generated from the tree model were found to be similar to the published ELL estimates for Nepal Haslett & Jones (2006). Standard errors for the tree were slightly larger than those for the ELL technique, the extra variation representing model variability which is captured by the tree method, but not the ELL model.

Previous research projects using Monte Carlo simulation have shown similar patterns of coverage to the investigation into bootstrap soft estimation for complex data and concluded this as evidence to justify the use of the particular statistical technique being investigated (Francq & Govaerts 2014, Ali et al. 2014, Ólafsdóttir & Mudelsee 2014). Indeed, since the simulations involve a complex model, an approximately normal sampling distribution for poverty incidence is an unreasonable expectation, and so it is unlikely that the empirical coverage will lie within confidence limits of the nominal value. Most of the simulation studies examined were restricted to investigating empirical coverage only for 95% nominal coverage, whereas the thesis has presented a method which provides consistent coverage rates across a wide range of nominal coverage levels.

In this chapter, a valid method to adapt classification tree models for complex survey data to estimate poverty incidence has been developed. However, the classification tree cannot model measures of deprivation such as poverty gap and poverty severity. The next step is to extend the methodology to regression tree models for small area estimation of poverty incidence, poverty gap and poverty severity.

Chapter 7

Regression tree modelling of poverty measures

7.1 Introduction

The analysis in Chapter 6 indicated that the cluster bootstrap resampling method with soft tree estimation provides a valid method of generating standard errors of prediction of poverty incidence using classification tree models. The Monte Carlo study, which used classification tree methodology to model poverty incidence, is extended in this chapter to examine regression tree modelling of three different measures of deprivation: poverty incidence, poverty gap and poverty severity, as represented in the formula devised by Foster, Greer and Thorbecke (1984), described in Section 1.2. Poverty gap and poverty severity are measures based upon a continuous numerical response, so cannot be modelled using a classification tree. As discussed in Section 2.4, a continuous numerical response variable requires a regression tree model, under which the predicted value for each terminal node is the average response value for all observations which end up in that leaf.

7.1.1 FGT formula

The ELL analysis of poverty in Nepal included three measures of poverty which were functions of household per capita expenditure (Haslett & Jones 2006). These measures can be represented in a common mathematical framework known as the FGT equations (Foster et al. 1984):

$$P_a \;=\; \frac{1}{N}\sum_{n=1}^{N}\left(\frac{\mathrm{z}-\mathcal{E}_n}{z}\right)^a \cdot \mathrm{I}\left(\mathcal{E}_n < \mathrm{z}\right)\;, \tag{7.1}$$

where N denotes the population size of the area under investigation, $\mathcal{E}_n$ is the per capita expenditure of the n^{th} individual, and z = 7696 denotes the poverty line in terms of per capita expenditure. The term $\mathrm{I}\left(\mathcal{E}_n < \mathrm{z}\right)$ is an indicator function which take the value 1 when expenditure is below the poverty line and 0 otherwise.

The value of a represents a specific poverty measure. For a given small area, *poverty incidence*, the proportion of individuals in the area living in households with expenditure below the poverty line, corresponds to $a = 0$. *poverty gap*, corresponding to $a = 1$, is the average distance below the poverty line, for poor households only. This particular measure of deprivation has implications for the targeting of aid: two small domains having similar levels of poverty incidence, would not necessarily have the same depth of poverty. The value of $a = 2$ provides a measure of the average squared distance below the poverty line, designated as *poverty severity*, giving more weight to those in extreme poverty.

The ELL modelling of poverty measures involved a prediction of log expenditure per household, which was then converted to a measure of per capita expenditure using the exponential transformation. Then, the non-linear functions described by Equation (7.1) were applied to per capita expenditure values at individual level, and these results aggregated to supply an estimate for the specific area of interest. The linear mixed model approach utilised in ELL provides a single type of prediction, whereas a tree model can produce two different estimate types, referred to as *hard* and *soft*.

7.2 Developing hard and soft regression tree estimates

A classification tree model can provide two types of prediction for poverty incidence, *hard* and *soft*. When a new observation is put into the model, it travels down the tree until it reaches a terminal node, or leaf, of the tree. The predicted value for each leaf in a classification tree is one of the two classes of the binary response variable. A hard estimate for a specific observation is the predicted value of the particular leaf to which it migrates. A soft estimate for the observation is the posterior probability of the class of interest in that particular leaf of the tree. Under the classification tree modelling of poverty incidence, the hard prediction for each household was the designation as "poor" or "not poor". These indicators of poverty status were aggregated to predict the proportion of poor in an area of interest, and so provided a hard small area estimate of poverty incidence. The probability of being poor in a leaf, the proportion of households in the leaf designated as "poor", represented a soft estimate of poverty incidence for each household in that particular leaf. These soft estimates were averaged over the small area to provide a soft small area estimate of poverty incidence.

When modelling the three measures of poverty, incidence, gap and severity, using the regression tree technique, the response variable used, Y, was log expenditure, as compared with a category of "poor" or "not poor" for classification tree models. The predicted value at the k^{th} leaf of a regression tree is the mean, μ_k, of the response variable Y_{ik}, log expenditure, for all observations which comprise that particular leaf. Furthermore, a probability distribution for the values of Y_{ik} can be associated with each leaf of the regression tree (Clark & Pregibon 1992). A normal probability distribution at each leaf of a regression tree can be used to develop a soft estimate for the three poverty measures.

7.2.1 Node distribution for a regression tree

The response variable for the regression tree is $Y = \log$ expenditure, the log transformation being applied so that Y has an approximately normal distribution, $Y \sim N(\mu, \sigma^2)$. The k^{th} terminal node, or leaf, of a regression tree can be considered to have an associated normal distribution (Clark & Pregibon 1992), $Y_{ik} \sim N\left(\mu_k, \sigma_k^2\right)$, for Y_{ik} the value of log expenditure for the i^{th} observation in the k^{th} leaf, with μ_k and σ_k^2 the mean and variance, respectively, of the log expenditure values for those observations which end up in the k^{th} leaf. The criteria for choosing splitting conditions at the k^{th} node is based upon the deviance, the sum of squares for that node.

$$\sum_{i=1}^{n_k} (y_{ik} \; - \; \mu_k)^2 \; ,$$

where n_k denotes the number of observations in the k^{th} node. The best splitting criterion (Breiman et al. 1984) is that which maximises the change in deviance between the parent node and the two daughter nodes (see Section 2.4.2.3).

Hard and soft predictions for each of the three poverty measures can be derived from the distribution parameters μ_k and σ_k^2. The development of hard and soft estimators is discussed in a separate section for each poverty measure, beginning with poverty incidence.

7.2.2 Poverty incidence

The measure of deprivation denoted as *poverty incidence* indicates the proportion of the population in poverty. Being an easy measure to interpret it is commonly used, but it does not quantify the degree of poverty present (World Bank 2005). The mathematical expression for poverty incidence corresponds to $a = 0$ in the FGT expression in Equation (7.1),

$$P_0 \; = \; \frac{1}{N} \sum_{n=1}^{N} \mathrm{I}\left(\mathcal{E}_n < \mathrm{z}\right) \; . \tag{7.2}$$

A hard estimate for poverty incidence from a regression tree is obtained by assigning to each household in the k^{th} leaf the value μ_k, the predicted value for all households in that leaf. All households in the k^{th} terminal node are then designated as being poor if the mean log expenditure value for that node, μ_k, is below the poverty line, otherwise they are classed as not poor. The hard small area estimate of poverty incidence is then the proportion of individuals classed as being poor in the area of interest.

A soft estimate for poverty incidence for a household in census data which ends up in the k^{th} leaf is obtained from the regression tree model by taking the expectation of the indicator function in Equation (7.2), as follows,

$$\begin{aligned}
\mathrm{E}\left[\,\mathrm{I}\left(\mathcal{E}_{ik} < \mathrm{z}\right) \mid k\,\right] &= \int_{-\infty}^{\infty} \left[\,\mathrm{I}\left(\mathcal{E}_{ik} < \mathrm{z}\right) \mid k\,\right] \, f_k(\boldsymbol{\varepsilon})\, d\boldsymbol{\varepsilon} \\
&= \int_{-\infty}^{\mathrm{z}} f_k(\boldsymbol{\varepsilon})\, d\boldsymbol{\varepsilon} \\
&= \mathrm{P}\left[\mathcal{E}_{ik} < \mathrm{z} \mid k\,\right] \\
&= \mathrm{P}\left[Y_{ik} < \log(\mathrm{z}) \mid k\,\right] , \qquad (7.3)
\end{aligned}$$

where $\mathcal{E}_{ik}$ denotes per capita expenditure of the i^{th} household in the census dataset which is assigned to the k^{th} leaf, and $Y_{ik} = \log\left(\mathcal{E}_{ik}\right)$, for $Y_{ik} \sim N(\mu_k, \sigma_k^2)$. The term $f_k(\boldsymbol{\varepsilon})$ indicates that the density function is dependent upon the leaf which is a household's final destination. Thus, the soft estimate of poverty incidence, Equation (7.3), for a household in the k^{th} leaf is $\mathrm{P}\left[Y_{ik} < \log(\mathrm{z}) \mid \mu_k, \sigma_k^2\,\right]$, the probability of that household is poor, given the value of mean μ_k and variance, σ_k^2 at the k^{th} terminal node. A soft estimate of poverty incidence, the probability of being poor, can be generated for each household, then for each individual in a specific household, and these probabilities aggregated across individuals to provide a small area soft estimate of poverty incidence.

7.2.3 Poverty gap

Poverty gap measures the difference between actual household expenditure and the poverty line, as a proportion of the poverty line, and indicates the extent of poverty in a household. The sum of the poverty gap values provides a minimum cost of eliminating poverty (World Bank 2005). A mathematical formula for poverty gap corresponds to the FGT formula, Equation (7.1), with $a = 1$, as follows,

$$P_1 \;=\; \frac{1}{N} \sum_{n=1}^{N} \left(\frac{\mathrm{z} - \mathcal{E}_n}{\mathrm{z}} \right) \;\cdot\; \mathrm{I}\left(\mathcal{E}_n < \mathrm{z}\right) \; .$$

The hard estimate for poverty gap for the i^{th} census observation which ends up in the k^{th} leaf is simply

$$P_{1_{ik}} \;=\; \frac{\mathrm{z} \;-\; \mathcal{E}_{ik}}{\mathrm{z}} \;\cdot\; \mathrm{I}\left(\mathcal{E}_{ik} < \mathrm{z}\right) \; ,$$

where $\mathrm{I}\left(\mathcal{E}_{ik} < \mathrm{z}\right)$ indicates that poverty gap is measured only for households below the poverty line, z. The term $\mathcal{E}_{ik} = \mathrm{e}^{\,\mu_k}$, and μ_k is the predicted value of log expenditure for all households in the k^{th} terminal node. The values of P_{1_i} are then aggregated across the small area of interest only for individuals in households below the poverty line, i.e. having $(\mathcal{E}_{ik} < \mathrm{z})$. To provide a soft estimator of poverty gap we utilise the function $g(\mathcal{E}_{ik})$, where

$$g(\mathcal{E}_{ik}) \;=\; \mathrm{I}\left(\mathcal{E}_{ik} < \mathrm{z}\right) \;\cdot\; \frac{(\mathrm{z} \;-\; \mathcal{E}_{ik})}{\mathrm{z}} \; .$$

A soft tree estimate for poverty gap is developed by taking the expectation of $g(\mathcal{E}_{ik})$,

$$\begin{aligned}
\mathrm{E}\left[g(\mathcal{E}_{ik}) \mid k\right] &= \mathrm{E}\left[\mathrm{I}\left(\mathcal{E}_{ik} < \mathrm{z}\right) \cdot \frac{(\mathrm{z} - \mathcal{E}_{ik})}{\mathrm{z}} \mid k\right] \\
&= \int_{-\infty}^{\mathrm{z}} \frac{(\mathrm{z} - \varepsilon)}{\mathrm{z}} f_k(\varepsilon)\, d\varepsilon \\
&= \int_{-\infty}^{\mathrm{z}} \left(1 - \frac{\varepsilon}{\mathrm{z}}\right) f_k(\varepsilon)\, d\varepsilon \\
&= \mathrm{P}\left[\mathcal{E}_{ik} < \mathrm{z} \mid k\right] \quad - \quad \frac{1}{\mathrm{z}} \int_{-\infty}^{\mathrm{z}} \varepsilon\, f_k(\varepsilon)\, d\varepsilon\,. \qquad (7.4)
\end{aligned}$$

where $f_k(\varepsilon)$ denotes the density function of the expenditure variable $\mathcal{E}$, conditional on the k^{th} leaf.

To facilitate the formulation of a soft estimator we consider a change of variable, $Y = \log(\mathcal{E})$, since Y has a normal distribution, $Y \sim N(\mu, \sigma^2)$. Since $\mathcal{E} = \mathrm{e}^Y$, the expectation, Equation (7.4), becomes,

$$\begin{aligned}
\mathrm{P}\left[\mathcal{E}_{ik} < \mathrm{z}\right] \quad - \quad \frac{1}{z} \int_{-\infty}^{\mathrm{z}} \varepsilon\, f_k(\varepsilon)\, d\varepsilon &= \mathrm{P}\left[Y_{ik} < \log(\mathrm{z}) \mid k\right] - \frac{1}{\mathrm{z}} \int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^{\,y} f_k(y)\, dy \\
&= \mathrm{P}\left[Y_{ik} < \log(\mathrm{z}) \mid k\right] - \frac{1}{\mathrm{z}} \int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^{\,y} \frac{1}{\sqrt{2\pi\sigma_k^2}}\, \mathrm{e}^{-\frac{(y-\mu_k)^2}{2\,\sigma_k^2}}\, dy \\
&= \mathrm{P}\left[Y_{ik} < \log(\mathrm{z}) \mid k\right] - \frac{1}{\mathrm{z}} \frac{1}{\sqrt{2\pi\sigma_k^2}} \int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^{\,y} \cdot \mathrm{e}^{-\frac{(y-\mu_k)^2}{2\,\sigma_k^2}}\, dy\,, \qquad (7.5)
\end{aligned}$$

where $f_k(y)$ is the density function of Y given leaf k. By completing the square, Equation (7.5) can be written as,

$$\mathrm{P}\left[Y_{ik} < \log(z) \mid k\right] - \frac{\mathrm{e}^{\left(\mu_k + \frac{\sigma_k^2}{2}\right)}}{z} \int_{-\infty}^{\log(\mathrm{z})} \frac{1}{\sqrt{2\pi\sigma_k^2}}\, \mathrm{e}^{-\frac{\left[y - \left(\mu_k + \sigma_k^2\right)\right]^2}{2\,\sigma_k^2}}\, dy$$

which can be expressed in terms of two probabilities,

$$\mathrm{P}\left[Y_{ik} < \log(\mathrm{z}) \mid \mu_k, \sigma_k^2\right] \quad - \quad \frac{\mathrm{e}^{\left(\mu_k + \frac{\sigma_k^2}{2}\right)}}{z}\, \mathrm{P}\left[Y_{ik}^* < \log(\mathrm{z}) \mid \mu_k, \sigma_k^2\right] \qquad (7.6)$$

with $Y_{ik} \sim N(\mu_k, \sigma_k^2)$ and $Y_{ik}^* \sim N(\mu_k + \sigma_k^2, \sigma_k^2)$. The full mathematical derivation of Equation (7.6) is provided in Appendix D.1.

The procedure outlined above, completing the square, is similar to the derivation of a Bayesian posterior distribution from a Gaussian likelihood and conjugate Gaussian prior. We see that the soft estimate for poverty gap comprises a fairly straightforward expression which includes the soft estimate for poverty incidence and a second probability term related to a random variable Y^*. The distribution of Y^* represents a shift in mean of σ^2 from the mean, μ, of the distribution of the original response Y, but the variance of Y^* is the same as the variance of Y. We now develop expressions for hard and soft estimates for poverty severity.

7.2.4 Poverty severity

A measure of deprivation which gives more weight to households in extreme poverty is obtained by utilising poverty severity, the averaged squared distance below the poverty line (World Bank 2005). Poverty severity, P_2, is calculated by letting $a = 2$ in the FGT formulation, Equation (7.1). The hard estimate for poverty severity in the i^{th} household in the k^{th} leaf is then

$$P_2 \;=\; \left(\frac{\mathrm{z} \,-\, \mathcal{E}_{ik}}{\mathrm{z}}\right)^2 \cdot\; \mathrm{I}\left(\mathcal{E}_{ik} < \mathrm{z}\right) \;,$$

where $\mathcal{E}_{ik} = \mathrm{e}^{\mu_k}$ for an individual in the i^{th} household terminating its route down the regression tree at the j^{th} leaf. The term μ_k, denoting the predicted value of $Y = \log(\mathcal{E})$ at the k^{th} leaf, is the average of $Y_{ik} = \log(\mathcal{E}_{ik})$ values for all households in the leaf. The values of P_{2_i} are then aggregated across the small area of interest only for households with $\mathcal{E}_{ik}$ below the poverty live. A soft estimator of poverty severity is obtained using a process similar to that for poverty gap. To devise a soft estimator for poverty severity we consider the function $h(\mathcal{E}_{ik})$,

$$h(\mathcal{E}_{ik}) \;=\; \left(\frac{\mathrm{z} \,-\, \mathcal{E}_{ik}}{\mathrm{z}}\right)^2 \cdot\; \mathrm{I}\left(\mathcal{E}_{ik} < \mathrm{z}\right) \;.$$

A soft tree estimate for poverty severity is developed by taking the expectation of $h(E)$,

$$\begin{aligned}
\mathrm{E}\left[h(\mathcal{E}_{ik})\right] \;&=\; \mathrm{E}\left[\mathrm{I}\left(\mathcal{E}_{ik} < \mathrm{z}\right) \cdot \left(\frac{\mathrm{z} \,-\, \mathcal{E}_{ik}}{\mathrm{z}}\right)^2\right] \\
&=\; \int_{-\infty}^{\mathrm{z}} \left(\frac{\mathrm{z} \,-\, \varepsilon}{\mathrm{z}}\right)^2 f_k(\varepsilon)\, d\varepsilon \\
&=\; \int_{-\infty}^{\mathrm{z}} \left(\frac{\mathrm{z}^2 - 2\mathrm{z}\varepsilon + \varepsilon^2}{\mathrm{z}^2}\right) f_k(\varepsilon)\, d\varepsilon \;.
\end{aligned}$$

The expression under the integral on the right hand side can be expanded as follows,

$$\begin{aligned} \mathrm{E}\left[h(\mathcal{E}_{ik})\right] &= \int_{-\infty}^{\mathrm{z}} \left(1 - \frac{2\varepsilon}{\mathrm{z}} + \frac{\varepsilon^2}{\mathrm{z}^2}\right) f_k(\varepsilon)\, d\varepsilon \\ &= \mathrm{P}\left[\mathcal{E}_{ik} < \mathrm{z}\right] - \frac{2}{\mathrm{z}} \int_{-\infty}^{\mathrm{z}} \varepsilon\, f_k(\varepsilon)\, d\varepsilon + \frac{1}{\mathrm{z}^2} \int_{-\infty}^{\mathrm{z}} \varepsilon^2\, f_k(\varepsilon)\, d\varepsilon\,, \end{aligned} \tag{7.7}$$

where $f_k(\varepsilon)$ denotes the density function of the expenditure variable $\mathcal{E}_{ik}$, at the k^{th} leaf. As with the soft estimator for poverty gap we we consider a change of variable, to use $Y = \log(\mathcal{E})$, since Y has a normal distribution, $Y \sim N(\mu, \sigma^2)$. Since

$$\begin{aligned} \mathcal{E}_{ik} &= \mathrm{e}^{Y} \\ \therefore \quad \mathcal{E}^2 &= \mathrm{e}^{2Y}\,. \end{aligned}$$

The expectation can then be written in terms of Y instead of $\mathcal{E}$ as follows;

$$\mathrm{P}\left[Y_{ik} < \log(\mathrm{z}) \mid k\,\right] - \frac{2}{\mathrm{z}} \int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^{y}\, f_k(y)\, dy + \frac{1}{\mathrm{z}^2} \int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^{2y}\, f_k(y)\, dy\,. \tag{7.8}$$

As was done in Section 7.2.3 to develop a soft estimator for Poverty Gap, completion of the square can be applied to the second term in Equation (7.8). Comparing Equation (7.8) with Equations (7.5) and (7.6) indicates that Equation (7.8) can be rewritten as,

$$\mathrm{P}\left[Y_{ik} < \log(\mathrm{z}) \mid \mu_k, \sigma_k^2\right] - \frac{2\,\mathrm{e}^{\left(\mu_k + \frac{\sigma_k^2}{2}\right)}}{z}\, \mathrm{P}\left[Y < \log(\mathrm{z}) \mid \mu_k + \sigma_k^2, \sigma_k^2\right] + \frac{1}{\mathrm{z}^2} \int_{-\infty}^{\mathrm{z}} \varepsilon^2\, f(\varepsilon)\, d\varepsilon, \tag{7.9}$$

Using the same procedure in the third term in the Expression 7.9, a soft estimate of poverty severity can be expressed as;

$$\begin{aligned} &\mathrm{P}\left[Y_{ik} < \log(\mathrm{z}) \mid \mu_k, \sigma_k^2\right] - \frac{2\,\mathrm{e}^{\left(\mu_k + \frac{\sigma_k^2}{2}\right)}}{z}\, \mathrm{P}\left[Y_{ik}^{*} < \log(\mathrm{z}) \mid \mu_k, \sigma_k^2\right] \\ &\qquad + \frac{\mathrm{e}^{2\left(\mu_k + \sigma_k^2\right)}}{\mathrm{z}^2}\, \mathrm{P}\left[Y_{ik}^{**} < \log(\mathrm{z}) \mid \mu_k, \sigma_k^2\right]\,, \end{aligned} \tag{7.10}$$

where $Y_{ik} \sim N(\mu_k, \sigma_k^2)$, $Y_{ik}^{*} \sim N(\mu_k + \sigma_k^2, \sigma_k^2)$ and $Y_{ik}^{**} \sim N(\mu_k + 2\,\sigma_k^2, \sigma_k^2)$. Thus the soft estimate of poverty severity, the expectation of the function $h(\mathcal{E})$, can be expressed as a linear combination of three probabilities, Equation (7.10). The full mathematical derivation of Equation (7.10) is provided in Appendix D.2.

Utilising regression tree models allows us to model poverty gap and poverty severity as well as poverty incidence. The results from estimating poverty incidence using regression trees provides an interesting comparison to classification tree models for the proportion of poor in an area. The analysis of each poverty measure is discussed in separate sections, but, firstly we describe the adjustments made to the simulation process described in Chapter 6 for modelling poverty measures using regression trees as opposed to classification trees.

7.3 Monte Carlo simulation with regression tree modelling

The Monte Carlo study for regression trees adapts the simulation process applied to classification trees, as outlined in Chapter 6, by utilising a regression tree rather than classification tree model to generate estimates. The regression tree model for poverty measures utilised a continuous numeric response variable, Y = log expenditure, as compared with the classification tree model which involved a categorical response of "poor" or "not poor". Consequently, the predicted value at the k^{th} terminal node for a regression tree was μ_k, the mean response value of Y_{ik} for all households dispatched to that particular node. To simulate clustering in the data structure, various levels of clustering, k = 0, 0.04, 0.08, 0.12, 0.16, 0.20 and 0.24, were introduced into model used to construct the simulated datasets, Equation (6.1). Since the response variable for the regression tree, Y = log expenditure, is assumed to have a normal distribution, the variance of the cluster residuals, σ_c^2, was estimated using a linear mixed model (LMM) of the form,

$$Y_{jk} \;=\; \mu_k \;+\; c_j \;, \tag{7.11}$$

rather than the generalised linear mixed model, Equation (6.10), which was applied to estimate cluster variance from the classification tree model. The term μ_k in equation (7.11) is the predicted value at the k^{th} leaf, the average of log expenditure values for all households which end up in the k^{th} leaf, and c_j denotes the random cluster effect. Parametric perturbations, c_j, were generated from the normal distribution $N\left(0, \sigma_c^2\right)$ and incorporated into the tree predictions, to provide amended predictions μ_{jk}^*,

$$\mu_{jk}^* \;=\; \mu_k \;+\; c_j \;,$$

for each household in the j^{th} small area cluster which ends up in the k^{th} leaf. Sela & Simonoff (2012) applied the LMM technique to adapt regression tress for clustered data, using an iterative approach similar to the EM algorithm, but found that re-estimating the tree did not really provide better predictions. The thesis utilises LMM for a different purpose, variance estimation, by adding cluster effects to tree predictions. From the amended predictions, μ_{jk}^*, bootstrap hard and soft regression estimates, as presented in Section 7.2, were generated for all three measures: poverty incidence, poverty gap and poverty severity. Thus, the probabilities used in the soft estimators of the three poverty

measures, as described by Equations (7.3), (7.6) and (7.10), took the form.

$$\mathrm{P}\left[Y_{jk} < \log(\mathrm{z}) \mid \mu^*_{jk}, \sigma^2_k\right], \qquad \mathrm{P}\left[Y^*_{jk} < \log(\mathrm{z}) \mid \mu^*_{jk}, \sigma^2_k\right], \qquad \mathrm{P}\left[Y^{**}_{jk} < \log(\mathrm{z}) \mid \mu^*_{jk}, \sigma^2_k\right]$$

where, $Y_{jk} \sim N(\mu^*_{jk}, \sigma^2_k)$, $Y^*_{jk} \sim N(\mu^*_{jk} + \sigma^2_k, \sigma^2_k)$ and $Y^{**}_{jk} \sim N(\mu^*_{jk} + 2\,\sigma^2_k, \sigma^2_k)$

To test the validity of the cluster bootstrap soft method under the regression tree model, coverage patterns of three interval types were examined. Simulation results for each of the three poverty measures are given in separate sections, beginning with poverty incidence.

7.3.1 Results for poverty incidence

A Monte Carlo simulation investigated the application of the cluster bootstrap variance estimation method for predicting poverty incidence using a regression tree model. Results of this study are summarised in Table 7.1, which displays average bias and standard error (s.e.) for bootstrap hard and soft tree estimates for different amounts of clustering in the data structure. The rationale for including hard bootstrap estimation in the simulations was to compare these results with the hard bootstrap estimates for poverty incidence obtained from the classification tree model, as listed in Table 5.1.

Table 7.1: Average bias and s.e. of hard and soft regression tree estimates for poverty incidence

K	**True**	**Hard**		**Soft**	
	P0	Bias	se	Bias	se
0	0.1993	-0.1016	0.0277	-0.0010	0.0061
0.04	0.1997	-0.1015	0.0259	-0.0007	0.0065
0.08	0.2009	-0.1002	0.0210	0.0002	0.0076
0.12	0.2028	-0.0980	0.0182	0.0017	0.0094
0.16	0.2054	-0.0950	0.0175	0.0038	0.0114
0.20	0.2087	-0.0915	0.0182	0.0062	0.0135
0.24	0.2125	-0.0873	0.0197	0.0090	0.0156

The estimates displayed in Table 5.1 were generated using classification tree models built from simulated simple random sample data, which equates to no clustering in the model, $k = 0$, a survey size of 3000 and a fixed small area of 6000 observations having "true" P0 of 0.1962. A comparison of Table 7.1 with Table 5.1 indicates that hard bootstrap estimates of poverty incidence generated by a regression tree model have bias and standard error of the same order as the hard estimates from the classification tree model. Even though hard predictions from the regression tree are based on continuous values rather than one of two classes, the granular nature of the hard type of tree estimate for poverty incidence is still evident. Interestingly, bias of the bootstrap hard estimates from the regression tree model tends to decrease with increasing amounts of clustering in the data structure. However, the hard cluster bootstrap method is not suitable for

estimating poverty incidence using a regression tree model because it generates biases and standard errors which are too large.

The bootstrap soft estimates from the regression tree model also tend to be of the same order as their counterparts from the classification tree model, Table 5.1. When $k = 0$, i.e. for a data structure arising from simple random sampling, the classification tree estimate of bias, - 0.0003, is smaller than the equivalent estimate from the regression tree, -0.001. However, this comparison is not reliable, since the classification tree estimate is based on a single fixed small area dataset, whereas the regression tree modelling employed a new small area dataset for each simulation.

To investigate whether the cluster bootstrap soft method applied to regression trees provides valid estimates of standard error for poverty incidence, coverage properties of the cluster bootstrap method were investigated using three types of intervals; parametric intervals centered the mean bootstrap estimate and the full tree estimate respectively, and a bootstrap percentile interval. Empirical coverage of the three interval types for different amounts of clustering in the data structure and for four nominal coverage levels is displayed in Figure 7.1.

Coverage patterns for the regression tree modelling in Figure 7.1 are similar to the corresponding results from the classification tree model, shown in Figure 6.5. Severe undercoverage is again evident when the data is clustered but cluster effects are not incorporated into predictions, as indicated in the black lines on the plots. Actual coverage from cluster bootstrap soft estimation using the regression tree model for poverty incidence is slightly lower than that for the classification tree modelling, but still within a few points of the nominal value. Empirical coverage using the regression tree is more consistent than that produced by the classification tree model across varying levels of clustering in the data, for all three interval types. The decreasing trend for higher levels of clustering with the percentile bootstrap interval and the parametric interval centred about the mean of bootstrap estimates, seen in the classification tree modelling (Figure 6.5), is not so apparent for simulations utilizing the regression tree (Figure 7.1). As was seen with the classification tree modelling, the cluster bootstrap soft estimation method applied to the regression tree model has produced valid standard errors of prediction for poverty incidence. Results of simulations for poverty gap are given in the next section.

7.3.2 Results for poverty gap and poverty severity

The deprivation measures, poverty gap and poverty severity, quantify the degree of poverty rather than just identifying the proportion of impoverished households. Poverty gap describes the average level of poverty for those households below the poverty line. The measure of deprivation defined by poverty severity is computed as squared poverty gap relative to the poverty line and gives greater weight to the poorest households. Monte Carlo simulations to investigate cluster bootstrap soft estimation of poverty gap and poverty severity based on the regression tree model generated estimates of bias and standard error for hard and soft bootstrap predictions, as displayed in Tables 7.2 and 7.3.

Figure 7.1: Empirical coverage for regression tree estimates of poverty incidence for three types of intervals: Soft - centred about soft bootstrap estimate; Full - centred about full tree estimate; Percent - bootstrap percentile; None - no cluster effects in predictions

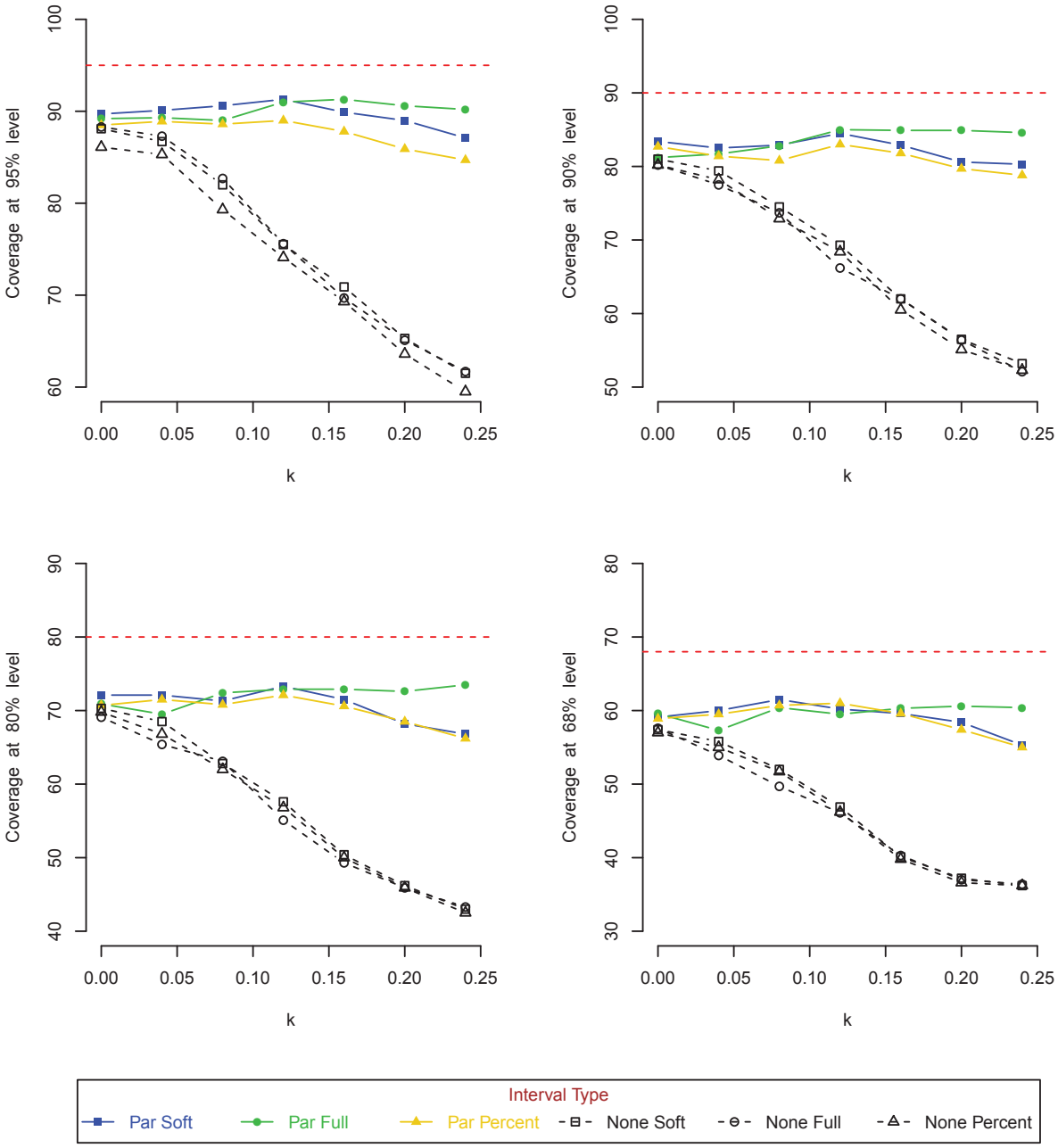


Table 7.2: Average bias and s.e. of hard and soft regression tree estimates for poverty gap

K	**True** P1	**Hard**		**Soft**	
		Bias	se	Bias	se
0	0.0578	-0.0384	0.0037	-0.0004	0.0026
0.04	0.0580	-0.0384	0.0037	-0.0003	0.0026
0.08	0.0586	-0.0384	0.0036	0.0002	0.0030
0.12	0.0597	-0.0383	0.0037	0.0010	0.0037
0.16	0.0611	-0.0381	0.0039	0.0022	0.0044
0.20	0.0629	-0.0378	0.0044	0.0037	0.0053
0.24	0.0650	-0.0373	0.0051	0.0055	0.0063

Table 7.3: Average bias and s.e. of hard and soft regression tree estimates for poverty severity

K	**True** P2	**Hard**		**Soft**	
		Bias	se	Bias	se
0	0.0243	-0.0180	0.0014	-0.0002	0.0014
0.04	0.0244	-0.0181	0.0014	-0.0001	0.0015
0.08	0.0248	-0.0182	0.0014	0.0001	0.0016
0.12	0.0254	-0.0184	0.0015	0.0006	0.0019
0.16	0.0262	-0.0185	0.0016	0.0014	0.0023
0.20	0.0272	-0.0186	0.0018	0.0023	0.0028
0.24	0.0286	-0.0190	0.0021	0.0034	0.0034

It can be seen from Tables 7.2 and 7.3 that the granular nature of the hard tree estimates is still producing considerable bias for poverty gap and poverty severity. However, bias for the soft tree estimates is small for $k = 0$, no clustering in the data structure, but increases with increasing k. Standard errors of prediction are very similar for hard and soft regression tree estimates of poverty gap and poverty severity, in contrast to the modelling for poverty incidence (Table 7.1), in which standard errors of prediction for hard estimates were up to three times the value of standard errors for soft estimates. Despite producing small standard errors, hard estimation is not a suitable method for estimating poverty gap or poverty severity with regression trees, since it results in very large bias.

Coverage properties of the cluster bootstrap soft estimation method under the regression tree model for estimating poverty gap and poverty severity were examined utilising the three types of interval. Empirical coverage for the three interval types is illustrated in Figures 7.2 and 7.3. Black lines on the graph representing severe undercoverage, confirm the need to incorporate cluster effects into regression tree predictions of poverty gap and poverty severity when the data is clustered. As seen with the coverage patterns for poverty incidence using the regression tree, actual coverage is very similar for all three interval types and within a few points of the nominal value.

Figure 7.2: Empirical coverage for regression tree estimates of poverty gap for three types of intervals: Soft - centred about soft bootstrap estimate; Full - centred about full tree estimate; Percent - bootstrap percentile; None - no cluster effects in predictions

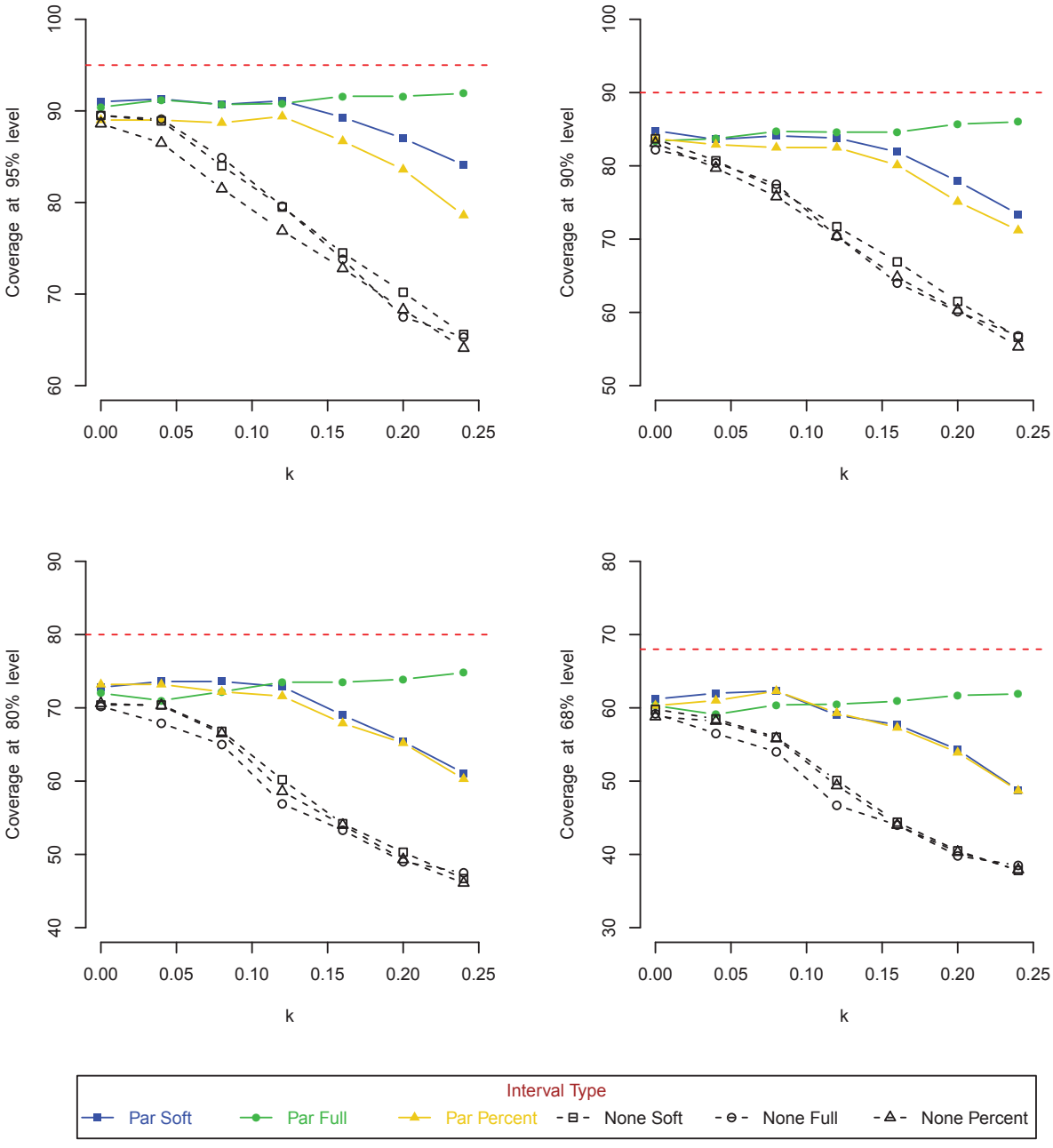
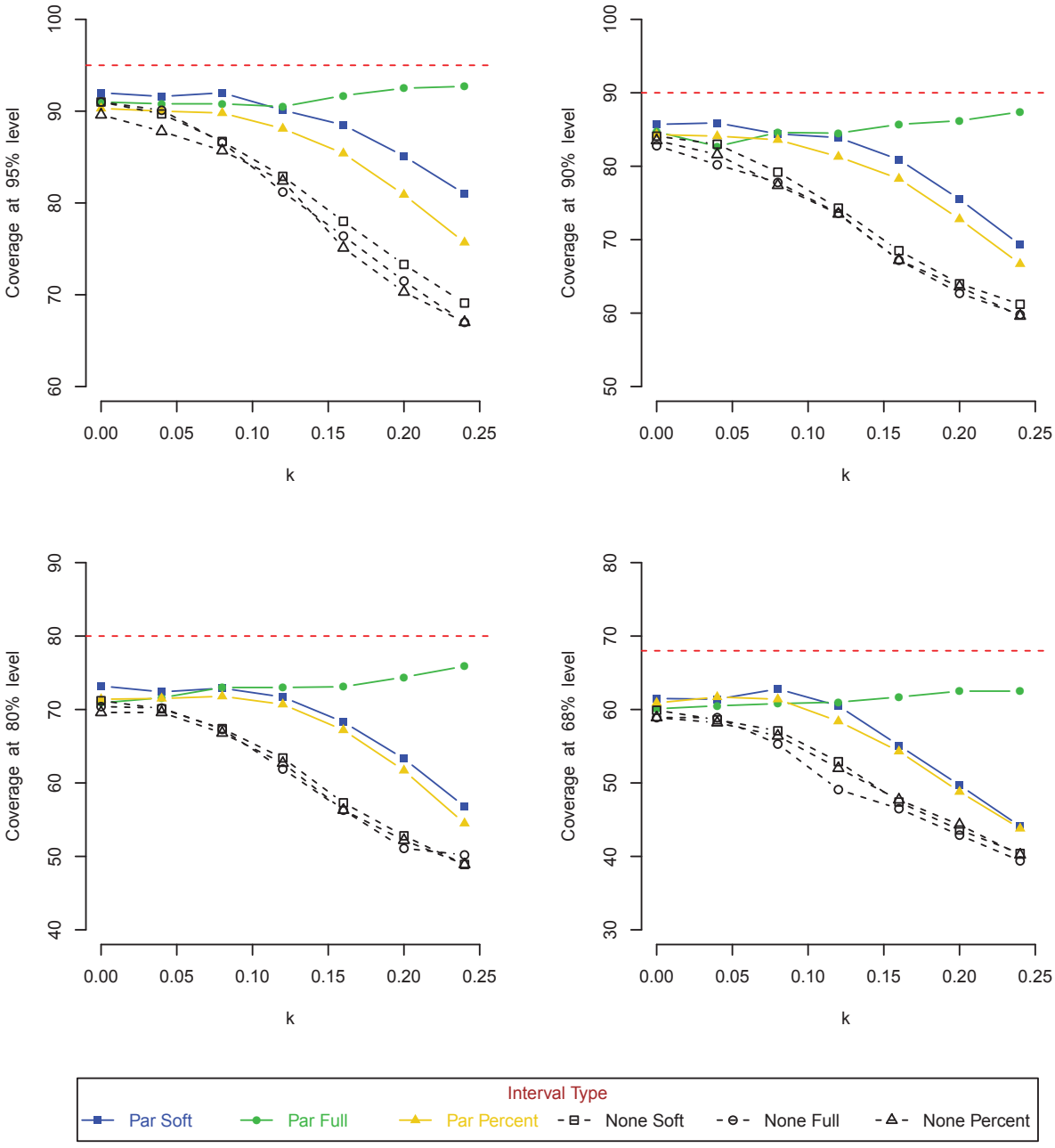


Figure 7.3: Empirical coverage for regression tree estimates of poverty severity for three types of intervals: Soft - centred about soft bootstrap estimate; Full - centred about full tree estimate; Percent - bootstrap percentile; None - no cluster effects in predictions



However, increasing undercover occurs for higher amounts of clustering with the bootstrap percentile intervals and parametric intervals with the mean of bootstrap soft estimates as centre. The method of cluster bootstrap soft estimation has provided valid standard errors of prediction for poverty gap and poverty severity, but for highly clustered data, point estimates generated from the full tree model are preferred above point estimates computed as the mean of bootstrap soft predictions. Based upon the simulation results, the cluster bootstrap soft estimation method using regression tree models is extended to include stratification as well as clustering in the survey design, using the Nepal data as an illustration.

7.4 Cluster bootstrap soft estimation of poverty measures for Nepal

The algorithm, outlined in Section 6.7.1 to generate classification tree small area estimates of poverty incidence in Nepal by applying the cluster bootstrap within each stratum, was adapted for regression tree modelling of poverty incidence, gap and severity. A weighted regression tree model instead of a weighted classification tree was constructed in Step 4 of the estimation process. Cluster variance was estimated in Step 5 using the linear mixed model, as described in Equation (7.11), rather than the linear mixed model, Equation (6.10) used with the classification tree method. Incorporation of cluster effects into predictions was achieved in a single step, which replaced Steps 7 and 8 of the algorithm for the classification tree estimation process described in Section 6.7.1. Soft predictions of poverty gap and severity, as well as poverty incidence, were generated as needed by the algorithm, using the code in Appendix C.3. The small area estimates of poverty incidence, gap and severity generated using the regression tree methodology, for the 18 ilakas in the chosen district of Nepal, are discussed in separate sections.

7.4.1 Regression tree estimates of poverty incidence

Regression tree small area estimates of poverty incidence and associated standard errors of prediction for the district in Nepal, tabulated in Table 7.4, are compared with the results of modelling poverty incidence in Nepal using a classification tree, Table 6.6, and the published results from the ELL method (Haslett & Jones 2006). To provide a meaningful comparison with the classification tree estimates of poverty incidence, the regression tree model used to generate small area estimates was pruned using the criterion of $cp = 0.001$. The resulting tree was then larger then the trees used in the simulation process, for which cp was set at zero and the tree restricted to a maximum depth of five.

Using the larger tree, the point estimate of poverty incidence for each ilaka produced from the full tree model was very close to the mean of the bootstrap soft predictions, as was seen with the classification tree estimation of poverty incidence in Nepal, Section 6.7.2. Thus, the point estimate of poverty incidence listed in Table 7.4, $P0$ in

Table 7.4: ELL and cluster bootstrap soft tree estimates of poverty incidence for a district in Nepal

Ilaka	**ELL estimates**		**Classification tree estimates**		**Regression tree estimates**	
	P0	se	P0	se	P0	se
1	0.742	0.030	0.653	0.058	0.603	0.050
2	0.525	0.033	0.463	0.045	0.448	0.036
3	0.619	0.035	0.545	0.061	0.516	0.046
4	0.468	0.024	0.397	0.042	0.393	0.034
5	0.297	0.028	0.304	0.046	0.254	0.031
6	0.472	0.027	0.460	0.035	0.397	0.036
7	0.704	0.036	0.611	0.065	0.558	0.070
8	0.204	0.031	0.188	0.047	0.176	0.034
9	0.201	0.028	0.190	0.057	0.182	0.035
10	0.122	0.024	0.135	0.035	0.116	0.028
11	0.242	0.020	0.249	0.034	0.239	0.026
12	0.202	0.027	0.229	0.035	0.227	0.029
13	0.213	0.020	0.238	0.033	0.236	0.026
14	0.415	0.031	0.414	0.038	0.412	0.032
15	0.311	0.026	0.300	0.033	0.283	0.029
16	0.063	0.021	0.073	0.028	0.056	0.025
17	0.130	0.040	0.099	0.046	0.106	0.034
18	0.137	0.034	0.094	0.043	0.078	0.029

Columns 4 and 6, is the mean of the bootstrap soft estimates of poverty incidence from the classification and regression tree models respectively, rather then the full tree point estimates, as suggested by the Monte Carlo simulations. From Table 7.4, it can be seen that standard errors of prediction of poverty incidence generated by the regression tree model are of the same order as the the ELL measures of standard error, but slightly larger.

The extra variation is due to the model variability being captured by the tree based method, whereas the ELL estimates are derived from a single model. Regression trees estimates of standard error are slightly smaller than those obtained from the classification tree model, which reflects the slight difference in coverage rates between Figures 6.5 and 7.1, the results of Monte Carlo simulations utilising the classification tree and regression tree models respectively. Point estimates of poverty incidence derived from the regression tree model are slightly smaller than their classification tree model counterparts. The estimates of poverty gap obtained from the regression tree model are examined in the next section.

7.4.2 Regression tree estimates of poverty gap

Table 7.5 presents cluster bootstrap soft small area estimates of poverty gap for a district in Nepal generated by regression tree modelling process as described earlier, and compares these with the published estimates obtained using the ELL methodology (Haslett & Jones 2006).

Table 7.5: ELL and cluster bootstrap soft regression tree estimates of poverty gap for a district in Nepal

Ilaka	**ELL estimates**		**Tree estimates**	
	P1	se	P1	se
1	0.282	0.022	0.200	0.027
2	0.172	0.016	0.137	0.017
3	0.217	0.021	0.162	0.023
4	0.144	0.012	0.115	0.014
5	0.079	0.010	0.069	0.012
6	0.154	0.014	0.117	0.017
7	0.263	0.025	0.183	0.037
8	0.050	0.010	0.044	0.012
9	0.048	0.008	0.045	0.011
10	0.030	0.007	0.028	0.008
11	0.067	0.007	0.066	0.009
12	0.051	0.009	0.063	0.011
13	0.055	0.007	0.064	0.010
14	0.122	0.013	0.117	0.014
15	0.087	0.010	0.079	0.011
16	0.014	0.006	0.012	0.007
17	0.033	0.014	0.024	0.010
18	0.033	0.011	0.016	0.007

As with the estimates for poverty incidence, Table 7.4, standard errors of prediction of poverty gap from the regression tree model are of the same order, but slightly larger, than those produced by the ELL method, again reflecting model variability being incorporated into the regression tree measures of variance. The regression tree point estimate of poverty gap, $P1$, is the average of the bootstrap soft estimates of poverty gap from a hundred bootstrap iterations. Values of $P1$ from the regression tree model are consistently lower than those from the ELL method, which is the same pattern seen with the estimates of poverty incidence, Table 7.4. The regression tree modelling was also applied to provide cluster bootstrap soft estimates of poverty severity for a district in Nepal. The results are presented in the following section.

7.4.3 Regression tree estimates of poverty severity

Table 7.6 displays small area estimates of poverty severity and associated standard errors for eighteen ilakas in a district of Nepal, generated by applying the cluster bootstrap soft estimation method to a regression tree model, and compares these estimates with the published results from the ELL methodology (Haslett & Jones 2006). The patterns seen in the estimates for poverty incidence and poverty gap are evident here also. Standard errors of prediction for poverty severity from the regression tree model are of the same order as those from the ELL model, but slightly larger, again indicating that the estimated variance from the regression tree incorporates model variability. Regression tree points

estimates of poverty severity, $P2$, obtained by taking the mean of a hundred bootstrap soft estimates of $P2$, are consistently smaller than their ELL equivalents.

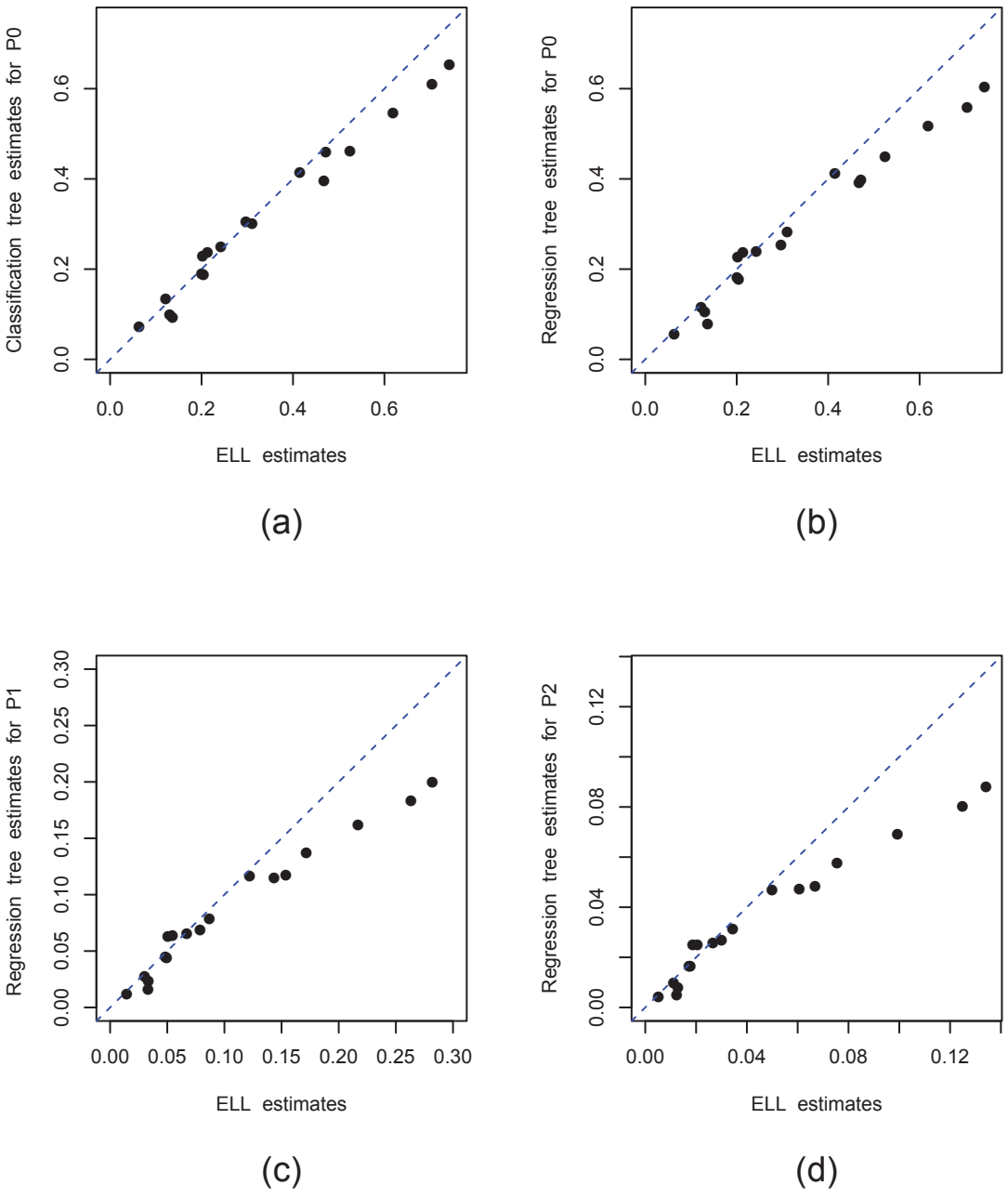
Table 7.6: ELL and cluster bootstrap soft regression tree estimates of poverty severity for a district in Nepal

Ilaka	**ELL estimates**		**Tree estimates**	
	P2	se	P2	se
1	0.134	0.015	0.088	0.015
2	0.075	0.009	0.058	0.009
3	0.099	0.013	0.069	0.012
4	0.061	0.006	0.047	0.007
5	0.030	0.005	0.027	0.006
6	0.067	0.008	0.048	0.009
7	0.125	0.016	0.080	0.021
8	0.018	0.005	0.016	0.005
9	0.017	0.004	0.017	0.005
10	0.011	0.003	0.010	0.003
11	0.027	0.004	0.026	0.004
12	0.019	0.004	0.025	0.005
13	0.021	0.003	0.025	0.005
14	0.050	0.007	0.047	0.007
15	0.034	0.005	0.031	0.006
16	0.005	0.003	0.004	0.003
17	0.013	0.007	0.008	0.004
18	0.012	0.005	0.005	0.003

Patterns of the differences between small areas point estimates of poverty incidence, gap and severity generated by classification and regression tree models, and those obtained from the ELL model are better illustrated using scatterplots, as provided in Figure 7.4. Plot (a) in Figure 7.4 compares point estimates of poverty incidence from the classification tree with ELL estimates; Plot (b) consists of regression tree point estimates of poverty incidence versus ELL equivalents; in Plots (c) and (d), regression tree point estimates of poverty gap and poverty severity respectively are plotted against their ELL counterparts.

The dashed blue lines, denoting $y = x$, represent equal values for ELL and tree estimates, and their inclusion in the plots confirms that tree based estimates are consistently lower than ELL estimates for the higher poverty rates, and consequently the higher values of poverty gap and poverty severity. Comparing the graphs representing estimates of poverty incidence, Plots (a) and (b), we note that the classification tree estimates of poverty incidence are closer to the ELL values than the regression tree estimates, as demonstrated in Table 7.4.

Figure 7.4: Plot of classification and regression tree point estimates versus ELL point estimates for a district of Nepal



Differences between ELL estimates and regression tree estimates are greater for poverty gap and poverty severity than for poverty incidence. The pattern of points in Plots (c) and (d), representing poverty gap and poverty severity, are similar , due to the fact that these measures are functions of the same quantity,

$$\frac{\mathrm{z} - \mathcal{E}}{\mathrm{z}}$$

7.5 Conclusion

Monte Carlo simulations described in Section 7.3, examined the behaviour of the cluster bootstrap soft estimation method when applied to regression trees to estimate poverty incidence, gap and severity. A linear mixed model estimated cluster variability based on the regression tree, and the estimated cluster variance was then used to generate parametric cluster residuals, which were incorporated into predictions. Results of the Monte Carlo study indicated that the cluster bootstrap soft method produced reasonable standard errors for all three poverty measures when used with a regression tree. Consequently, the cluster bootstrap soft method was applied to a particular district in Nepal to generate small area estimates for eighteen ilakas.

Implementation of the cluster bootstrap method for the Nepal data involve building regression tree models from bootstrap replicates which were constructed by combining cluster bootstrap samples from individual strata. Small area estimates of standard error for poverty incidence, gap and severity produced by the regression tree model were of the same order as their ELL counterparts, but slightly larger, which indicated that the estimates of variance included model variability. The cluster bootstrap soft estimation method has proved a valid technique for small area estimation of poverty measures using both the classification and regression tree models.

However, point estimates of poverty incidence, gap and severity from the regression tree model were consistently lower than the corresponding ELL point estimates, for the higher poverty rates. These differences between ELL and tree estimates may be related to model complexity. In the Monte Carlo simulations, tree complexity was determined by setting a maximum tree depth of five. Regression trees used in the estimation of poverty measures for a district in Nepal were pruned by specifying a complexity parameter value $cp = 0.001$. Using a larger tree, i.e. with more terminal nodes, produced larger estimates of all three poverty measures, with values closer to the ELL estimates. There is no obvious way to compare the complexity of the ELL model with the complexity of tree-based models.

Since there is no gold standard for poverty measures in Nepal, it cannot be determined whether the differences in point estimates indicates that the ELL method is overfitting or the tree method is underfitting. If the latter scenario is true, then underestimation at the higher levels of the poverty measures probably occurs because poor households are being misclassified as not being poor. In Section 3.2.7, the model fit of

a classification tree model was tested using the replicate subsamples, and the misclassification rate of poor households was found to be 59%. A similar analysis for a regression tree model, used to estimate poverty incidence, produced a misclassification rate for poor households of 57%. These results suggest that the systematic pattern of lower poverty estimates at higher poverty rates for the tree model, as compared to the ELL model, may be due to a class imbalance problem.

Another factor which may be contributing to the differences seen between ELL and tree estimates is departures from assumptions of normality. In the ELL modelling, normality is assumed for inference purposes but is not explicitly required for estimating poverty gap or severity. However, the soft estimators for poverty gap and poverty severity, derived in Sections 7.2.3 and 7.2.4, may be sensitive to normality assumptions. The procedure used in deriving these poverty measures, completing the square, requires a Gaussian density function for the response variable, Y = log expenditure, in order to combine it with exponential functions of Y under the integrals which define the expectations of poverty gap and severity. Point estimates generated by the tree-based models were (approximately) unbiased in the simulations, and it is difficult to determine for a specific, real dataset which method, ELL or tree-based, is, or may be, biased. This issue is discussed further in the last chapter.

Chapter 8

Discussion

8.1 Review of the thesis

The scope of the thesis was the adaptation of tree-based models for complex survey data, in the context of small area estimation of poverty measures. This required incorporating into the modelling the features of small area estimation: complex survey design, auxiliary information and a variance estimation procedure. A methodology was devised to predict poverty incidence using a classification tree model, and amended to generate predictions of poverty incidence, gap and severity using a regression tree model.

To take account of the complex survey design, a weighted tree model was used, to ensure that the model built from the survey dataset was unbiased. The aspects of stratification and clustering in the data were also incorporated into the variance estimation procedure, as discussed later. Increased precision in estimates is achieved by "borrowing strength" through the inclusion of auxiliary information. The procedure used in the thesis mirrored the approach taken in the ELL methodology: survey data was used to build a model, which was then applied to census data to generate small area estimates.

In developing a suitable variance estimation procedure, the first approach taken was to apply inverse sampling to replicates subsamples of the survey data. This proved unsuccessful, so the next step was to investigate the behaviour of jackknife and bootstrap resampling, with hard and soft types of tree estimate, for data simulated to have a simple random sampling structure. Applying the jackknife method resulted in large standard errors, which can be explained by the well known property of the jackknife technique of inconsistency with non-smooth estimators, such as a tree model. In addition, the discrete nature of the hard type of tree estimate produced large bias for both jackknife and bootstrap procedures.

Monte Carlo simulations indicated that the variance estimation method of ordinary bootstrap resampling with soft tree estimation provided valid standard errors of prediction for data with a simple random sampling structure, since empirical coverage was only a few points below nominal levels. The methodology was amended for the presence of stratification and clustering in the data by utilising the cluster bootstrap independently

within each stratum, and augmenting the soft tree prediction with a cluster residual, derived from a parametric estimate of cluster variability. Adding cluster effects was equivalent to perturbing the posterior probabilities of being poor at the terminal nodes.

The cluster bootstrap procedure with soft tree estimation was shown through Monte Carlo simulations to be a valid method of generating standard errors of prediction for both classification and regression tree models, and for all three poverty measures, with empirical coverage lying only a few points below the nominal level. In addition, the variance estimates obtained by applying the technique to actual Nepal data were of the same order as the corresponding published estimates generated using the ELL methodology. The tree-based estimates of standard error were slightly larger than those from the ELL regression model, since the tree estimates incorporated model variability through different selections of predictor variables. However, tree based estimates of poverty incidence, gap and severity were consistently lower than their ELL counterparts for the higher levels of all three poverty measures. The disparities between ELL and the tree models, with respect to point estimates and standard errors of prediction, are discussed further in the next section.

8.2 Weighing tree-based models against ELL

Tree-based models have advantages over the ELL method, the standard procedure for small area estimation of poverty measures. Structurally the two methodologies are similar, in that a model is fitted to a survey dataset and then applied to census data, but the classification tree model provides direct categorisation of households as poor or not poor, without first estimating income levels. In addition, a tree provides a more automatic method of variable selection, with the more important variables placed higher up in the tree structure. Predictor variables can be reselected at different splits, and variable interactions are readily incorporated into the tree structure. Generally, tree models are independent of distributional assumptions; however, modelling in the thesis was based on the log of per capita expenditure rather than the original income variable, since this transformation corrected to some degree the high level of skewness in the raw data.

The patterns seen in Chapter 7, of tree-based point estimates of poverty measures being consistently lower than the corresponding ELL estimates for higher poverty levels, could possibly be due to the different modelling approaches. ELL methodology builds a single model on the entire dataset, based upon the assumptions of linearity and additive effects. In contrast, the tree constructs a model, a constant, in each partitioning of the dataset, essentially producing a step function across the data space. Alternatively, these differences may be an idiosyncrasy of the particular district selected for predictions, which was chosen because it has a wide range of ELL poverty estimates across its constituent ilakas. Selection bias might appear to be an issue here, but tree-based predictions generated for a second district in Nepal, also having disparate poverty rates across ilakas, showed similar patterns of consistently lower estimates than the ELL equivalents for the

higher poverty levels.

Differences in the type of estimate applied in each technique may also contribute to consistently lower point estimates at higher poverty levels for tree models. The ELL method predicts log expenditure for each household, which is then converted to a poverty measure using the FGT identities (Foster et al. 1984), as outlined in Equation (7.1). This type of estimate from ELL could be described as "hard", since it is similar in form to the hard estimate developed for the regression tree model, as discussed in section 3.3.1. In contrast, the classification and regression tree modelling employed a soft type of estimate: the posterior probability of being poor in the classification tree model, and the regression tree approach utilised the expectation of a function based on the form of the hard estimate, to provide a soft prediction for a particular poverty measure. The "soft" estimators of poverty measures developed for the regression tree modelling of poverty incidence, gap and severity, depended upon the assumption of a normal distribution for the response variable, log expenditure. If the assumption of normality does not hold, this could explain the greater disparity between ELL and tree estimates seen for the regression tree model, as compared with the classification tree method.

A further reason for the discrepancies between ELL point estimates and those generated by the tree models is possible misclassification of poor households as being not poor, which would have the affect of lowering tree estimates for the higher poverty rates. Support for this explanation is provided by modelling with the larger tree, which had more leaves and thus a better misclassification rate, since the larger tree produced estimates that were closer to their ELL counterparts. A classical class imbalance problem, this issue is discussed further in the next section.

Standard errors of prediction obtained from the tree-based models were slightly larger than their ELL counterparts, which can also be explained by the different approaches to variance modelling. The ELL method builds a single, fixed model for predictions, having a fixed set of covariates, so that estimates of standard error are conditional on the structure of the model being correct. Variability is estimating by bootstrapping the sources of error: the regression coefficients ($\boldsymbol{\beta}^b$), cluster level effects (γ_j^b), and household level effects (ϵ_{ij}^b), as represented in Equation (8.1), where the superscript b indicates a bootstrap estimate.

$$Y_{ij}^b = \mathbf{x}_{ij}^T \boldsymbol{\beta}^b + \gamma_j^b + \epsilon_{ij}^b \,, \qquad b = 1, ..., B \,. \tag{8.1}$$

The tree model, however, employs unconditional variance estimation, since a different bootstrap sample at each iteration allows for a different tree structure. Variability is estimated by incorporating a residual cluster effect, c_j, into each tree prediction. This cluster effect, which was on the logit scale for the classification tree, and the scale of the original data for the regression tree, was drawn from a normal distribution, $N\left(0, \sigma_c^2\right)$, where σ_c^2 was estimated from the structure of the tree model used for prediction. For a household in the j^{th} cluster which ended up in the k^{th} leaf, the adjusted prediction for the classification tree was $\tilde{p}_{jk}^*$, such that

$$\tilde{p}^*_{jk} \;=\; \text{logit}^{-1}\left(logit\left(\tilde{p}_k\right) + c_j\right) \;,$$

where $\tilde{p}_k$ represents the posterior probability of being poor for the k^{th} leaf. The regression tree model provided an amended prediction, μ^*_{jk} with the form,

$$\mu^*_{jk} \;=\; \mu_k \;+\; c_j \;,$$

where μ_k denotes the predicted value, the mean of log expenditure for all observations in the k^{th} leaf. Thus, the tree-based variance estimation process incorporated cluster effects and model variability. The unconditional variance estimation approach used with tree models represents another advantage of the tree over the ELL method, since conditioning on the choice of predictors is not the preferred option; the set of explanatory variables used to build the model should be allowed to vary between bootstrap samples. Random Forest methodology facilitates variable choice by forcing a different predictor set for each bootstrap iteration, as discussed in the next section.

As an alternative to the unconditional variance estimation based on tree model, as outlined in the thesis, unconditional approaches could be applied and compared with the ELL unconditional method. One simple type of conditional tree model would utilise the information contained in the leaves of the tree to provide a new estimate, rather than refitting a new tree for each bootstrap replicate. A tree would be built from the full survey dataset, and the poverty measure adjusted to take account of re sampled observations as well as households omitted from the bootstrap sample. Estimates would then be conditional on the tree structure based upon the full dataset,a procedure which is similar to that applied with ELL. A second conditional model could be achieved by forcing the tree built from each bootstrap sample to have the same structure as that arising from the full survey dataset. The tree building algorithm would partition the data using the same splitting variables, in the same order, as the original tree, but allowing for different cut-off points at each split. This process, however, would involve a more complicated algorithm and be computationally more expensive. The purpose of applying conditional variance estimation to tree-based models would be to investigate whether conditioning on the structure of the tree model produces estimates of variance and point estimates of poverty which were closer to the corresponding ELL estimates than those resulting from the unconditional variance model for the tree.

For the methodology to be applied in a different context, with a new dataset, the first step is identification of auxiliary variables which are common to both survey and census, as is done with ELL. Consistency in definition and measurement of these variables is important, to ensure good matching between survey and census, so that the auxiliary information is valid. Careful matching is required when survey and census are taken at different time periods. However, when a new survey is conducted some time after the census, intercensal updating of small area estimates can be carried out (Isidro et al. 2016). The ELL method requires some preparatory work to predictors before modelling can begin, such as transformation of numerical variables, combining categories for factor

variables, creating indicator variables for each level of a factor, and constructing interaction variables. In contrast, once the sets of common auxiliary variables have been decided upon, the tree methodology can be applied without any further adjustment to the predictors, since variable selection occurs automatically with the tree, which readily incorporates nonlinearities, variable interactions and multiple categories.

To summarise, the tree based method is much simpler to use than the regression technique of ELL. The associated tree diagram is easily interpreted, and clearly illustrates the main determinants for poverty in a particular context. The tree algorithm automatically selects the model which best fits the survey data, whereas model selection with the ELL method involves examining different combinations of main effects and interactions, which can be a very time consuming process when the set of predictors is large. Tree based models are particularly useful as an automatic method of estimation, requiring minimal user input.

8.3 Further research

The research outlined in the thesis could be extended to investigate whether a possible class imbalance contributes to consistently lower poverty estimates from the tree model for higher poverty rates when compared with the equivalent ELL estimates. Traditional methods for dealing with a class imbalance problem include over-sampling of the minority class, or under-sampling the dominant class (Japkowicz & Stephen 2002). Another approach involves using a loss matrix in the modelling process, which incorporates a cost for misclassification of the minority class. If addressing the class imbalance issue increases point estimates from the tree models, then the tree-based methodology should be adapted to reflect this. However, due to the lack of a "gold standard" for poverty estimates in Nepal, there is no information available as to which of the methods, ELL or trees, is unbiased.

An alternative method of unconditional variance estimation using tree-based models is the Random Forest methodology (Breiman 2001). In the tree-based methods described in the thesis, all predictors are available as splitting variables for each internal node of the tree structure, whereas, for the Random Forest method, a subset of the predictors is randomly selected to provide the choices of splitting variables at each partitioning of the tree. The algorithm constructs multiple trees to provide an estimate which is the average of values from each tree built. A measure of variance could also be generated from these multiple tree estimates. Thus, an estimate of variance obtained using the Random Forest methodology is conditional on the predictor subset selected at each node, and would incorporate variability arising from forcing a different choice of predictors at each split. The challenge with utilising the Random Forest model would be how to incorporate the complex design into the modelling, in particular how to apply the cluster bootstrap within the existing algorithm.

Model diagnostics is an additional area of possible research. In particular, stability of estimates could be affected by whether nor not the allocation of predicted households is evenly spread across the terminal nodes of the tree. Instability in the estimate would be expected if all households in a specific cluster migrated to one or two leaves, and a more stable estimate should result if the households were spread across a number of leaves. A simple indication of tree purity, representing an even spread across the leaves for observations from the same cluster, could involve devising an appropriate distance measure between actual allocation across the leaves and what would be expected with an even allocation. This value would be subtracted from unity, so that a completely pure tree has purity measure of 1. Such an approach would need to take into account the situation when a cluster size is less than the total number of leaves, so that expected values are less than one.

Molina & Rao (2010) suggest that efficiency of the ELL method is reduced because ELL does not include information related to the direct estimator, based on the survey sample. A topic for further research could be how to amend the tree-based methodology for the situation in which some of the survey data can be identified within the census dataset. Generating a prediction for a survey household, for which the poverty level is already known, would not be useful, but including this "in sample" information could affect how cluster effects are assigned to predictions.

Application of small area estimation methodology to other non-linear situations and data mining techniques, such as neural networks, linear discriminant analysis and support vector machines (Hastie et al. 2001) is another possible research extension. As with the tree-base modelling, the issues to be addressed include incorporating into these methodologies the aspects of a complex data structure and auxiliary information to increase precision in the estimates, as well as devising a variance estimation method.

Bibliography

Agresti, A. (2013), *Categorical data analysis*, 3rd edn, New York: John Wiley & Sons.

Alderman, H., Babita, M., Demombynes, G., Makhatha, N. & Özler, B. (2002), 'How low can you go? combining census and survey data for mapping poverty in South Africa', *Journal of African Economies* **11**(2), 169–200.

Ali, R. A., Ali, M. A. & Wei, Z. (2014), 'On computing standard errors for marginal structural cox models', *Lifetime Data Analysis* **20**(1), 106–131.

Anscombe, F. J. (1948), 'The Transformation of Poisson, Binomial and Negative-Binomial Data', *Biometrika* **35**(3/4), 246–254.

Antal, E. & Tillé, Y. (2011), 'A direct bootstrap method for complex sampling designs from a finite population', *Journal of the American Statistical Association* **106**(494), 534–543.

Azzalini, A. & Scarpa, B. (2012), *Data Analysis and Data Mining : An Introduction*, Oxford University Press.

Baker, J. L. & Grosh, M. E. (1994), 'Poverty reduction through geographic targeting: How well does it work?', *World Development* **22**(7), 983 – 995.

Barnett, V. (2002), *Sample Survey Principles and Methods*, Arnold.

Barton, R. R., Nelson, B. L. & Xie, W. (2014), 'Quantifying input uncertainty via simulation confidence intervals', *INFORMS Journal on Computing* **26**(1), 74–87.

Bates, D. and Maechler, M. and Bolker, B. and Walker S. (2013), *lme4: Linear mixed-effects models using Eigen and S4*. R package version 1.1-7.

Berry, M. J. A. & Linoff, G. S. (2004), *Data Mining Techniques*, Wiley Publishing, Inc.

Betti, G. & Ballini, F. (2008), 'Variance estimates of poverty and inequality measures in Albania', *Eastern European Economics* **46**(6), 84–98.

Bickel, P. J., Götze, F. & van Zwet, W. R. (1997), 'Resampling fewer than n observations: Gains, losses, and remedies for losses', *Statistica Sinica* **7**(1), 1–31.

Breiman, L. (1996*a*), 'Bagging predictors', *Machine learning* **24**(2), 123 – 140.

Breiman, L. (1996*b*), 'Heuristics of instability and stabilization in model selection', *The Annals of Statistics* **24**(6), 2350–2383.

Breiman, L. (2001), 'Random forests', *Machine Learning* **45**(1), 5–32.

Breiman, L., Freidman, J. H., Olshen, R. A. & Stone, C. J. (1984), *Classification and Regression Trees*, Chapman and Hall.

Breslow, N. E. & Clayton, D. G. (1993), 'Approximate inference in generalized linear mixed models', *Journal of the American Statistical Association* **88**(421), 9–25.

Cameron, A. C., Gelbach, J. B. & Miller, D. L. (2008), 'Bootstrap-based improvements for inference with clustered errors', *The Review of Economics and Statistics* **90**(3), 414–427.

Central Bureau of Statistics, Nepal (2002), Population census 2001: National report, Technical report, Central Bureau of Statistics, Nepal.

Central Bureau of Statistics, Nepal (2004a), Nepal livings standards survey 2003/04: Statistical report volume one, Technical report, Central Bureau of Statistics, Nepal.

Central Bureau of Statistics, Nepal (2004b), Nepal livings standards survey 2003/04: Statistical report volume two, Technical report, Central Bureau of Statistics, Nepal.

Chambers, J. M. & Hastie, T. J. (1992), *Statistical Models in S*, Wadsworth & Brookes/Cole.

Ciampi, A., Chang, C.H., Hogg, S. and McKinney, S. (1987), Recursive partition: A versatile method for exploratory-data analysis in biostatistics, *in* I. B. MacNeill, G. J. Umphrey, A. Donner & V. K. Jandhyala, eds, 'Biostatistics', Vol. 38 of *The University of Western Ontario Series in Philosophy of Science*, Springer Netherlands, pp. 23–50.

Clark, L. A. & Pregibon, D. (1992), Tree-based models, *in* J. M. Chambers & T. J. Hastie, eds, 'Statistical Models in S', Wadsworths & Brookes/Cole, chapter 9.

Cochran, W. G. (1977), *Sampling Techniques*, 3rd edn, Wiley.

Conover, W. J. (1999), *Practical Nonparametric Statistics*, Wiley.

Coondoo, D., Majumder, A. & Chattopadhyay, S. (2011), 'District-level poverty estimation: a proposed method', *Journal of Applied Statistics* **38**(10), 2327–2343.

Cornfield, J. (1944), 'On samples from finite populations', *Journal of The American Statistical Association* **39**(226), 236–239.

Darlington, R. B. (1968), 'Multiple regression in psychological research and practice', *Psychological Bulletin* **69**(3), 161–182.

Deming, W. E. (1956), 'On simplifications of sampling design through replication with equal probabilities and without stages', *Journal of The American Statistical Association* **51**(273), 24–53.

Demombynes, G., Elbers, C., Lanjouw, J. O. & Lanjouw, P. F. (2007), 'How good a map? Putting small area estimation to the test', Technical report. World Bank Policy Research Working Paper 4155.

DiCiccio, T. J. & Efron, B. (1996), 'Bootstrap confidence intervals', *Statistical Science* **11**(3), 189–212.

Dictionary.com (2011), 'Bootstrap'.
URL: *http://dictionary.reference.com/browse/bootstrap*

Draper, N. R. & Smith, H. (1998), *Applied Regression Analysis*, John Wiley & Sons.

Dwyer, K. & Holte, R. (2007), Decision tree instability and active learning, *in* 'European Conference on Machine Learning', Springer, pp. 128–139.

Ebers, C., Lanjouw, P. F. & Leite, P. G. (2008), "Brazil within Brazil: testing the poverty map methodology in Mias Gerais". Policy Research Working Paper Number 4513.

Efron, B. (1979), '1977 Rietz lecture - Bootstrap methods - Another look at the jackknife', *Annals of Statistics* **7**(1), 1–26.

Efron, B. (1980), The jackknife, the bootstrap and other resampling plans, Technical report, Stanford University.

Efron, B. (1982), *The Jackknife, the Bootstrap and other Resampling Plans*, Society for Industrial and Applied Mathematics.

Efron, B. (1983), 'Estimating the error rate of a prediction rule - improvement on cross-validation', *Journal of The American Statistical Association* **78**(382), 316–331.

Efron, B. & Stein, C. (1981), 'The jackknife estimate of variance', *Annals of Statistics* **9**(3), 586–596.

Efron, B. & Tibshirani, R. J. (1993), *An Introduction to the Bootstrap*, Chapman and Hall.

Elbers, C., Fujii, T., Lanjouw, P., Özler, B. & Yin, W. (2007), 'Poverty alleviation through geographic targeting: How much does disaggregation help?', *Journal of Development Economics* **83**(1), 198–213.

Elbers, C., Lanjouw, J. O. & Lanjouw, P. (2003), 'Micro-level estimation of poverty and inequality', *Econometrica* **71**(1), 355–364.

Elkan, C. (2001), The foundations of cost-sensitive learning, *in* 'Proceedings of the Seventennth International Joint Conference on Artificial Intelligence'.

Ferré, C., Ferreira, F. H. & Lanjouw, P. (2012), 'Is there a metropolitan bias? the relationship between poverty and city size in a selection of developing countries', *The World Bank Economic Review* .

Field, C. A. & Welsh, A. H. (2007), 'Bootstrapping clustered data', *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **69**(3), 369–390.

Folsom, R E and Shah, B V and Vaish, A K (1999), Substance abuse in states: A metholodogical report on model-based estimates from the 1994-1996 National Household Surveys on Drug Abuse, *in* 'Proceedings of the Section on Survey Research Methods, American Statistical Association, 371-375'.

Foster, J., Greer, J. & Thorbecke, E. (1984), 'A class of decomposable poverty measures', *Econometrica* **52**(3), 761–766.

Francq, B. G. & Govaerts, B. B. (2014), 'Measurement methods comparison with errors-in-variables regressions. from horizontal to vertical {OLS} regression, review and new perspectives', *Chemometrics and Intelligent Laboratory Systems* **134**, 123–139.

Freund, Y. & Schapire, R. E. (1997), 'A decision-theoretic generalization of on-line learning and an application to boosting', *Journal of Computer and System Sciences* **55**(1), 119 – 139.

Freund, Y. & Schapire, R. E. (1999), 'A short introduction to boosting', *Journal of Japanese Society for Artificial Intelligence* **14**(5), 771–780.

Fujii, T. (2008), 'How well can we target aid with rapidly collected data? empirical results for poverty mapping from cambodia', *World Development* **36**(10), 1830–1842.

Fujii, T. (2010), 'Micro-level estimation of child undernutrition indicators in cambodia', *The World Bank Economic Review* **24**(3), 520–553.

Ghosh, M., Natarajan, K., Stroud, T. & Carlin, B. P. (1998), 'Generalized linear models for small-area estimation', *Journal of the American Statistical Association* **93**(441), 273–282.

Ghosh, M. & Rao, J. (1994), 'Small-area estimation - an appraisal', *Statistical Science* **9**(1), 55–76.

Gonzalez, M. E. (1973), 'Use and evaluation of synthetic estimates', *American Statistical Society, Proceedings of the Social Statistics Section* pp. 33–36.

Goodmin, P. & Wright, G. (2014), *Decision Analysis for Management Judgement*, John Wiley & Sons.

Green, P. J. & Silverman, B. W. (1994), *Nonparametric Regression and Generalised Linear Models*, Chapman and Hall.

Hagenaars, A. & De Vos, K. (1988), 'The definition and measurement of poverty', *The Journal of Human Resources* **23**(2), pp. 211–221.

Han, J., Kamber, M. & Pei, J. (2012), *Data Mining: Concepts and Techniques*, Elseveier Inc.

Hansen, M. H. & Hurwitz, W. N. (1942), 'Relative efficiencies of various sampling units in population inquiries', *Journal of The American Statistical Association* **37**(217), 89–94.

Hartley, H. O. (1966), 'Systematic sampling with unequal probability and without replacement', *Journal of The American Statistical Association* **61**(315), 739–748.

Haslett, S., Isidro, M. & Jones, G. (2010), 'Comparison of survey regression techniques in the context of small area estimation of poverty', *Survey Methodology* **36**(2), 157–170.

Haslett, S. J. & Jones, G. (2006), 'Small area estimation of poverty, caloric intake and malnutrition in Nepal', Technical report, Nepal Central Bureau of Statistics/World Food Programme, United Nations/World Bank, Katmandu, Nepal.
URL: *https://www.wfp.org/content/nepal-small-area-estimation-poverty-caloric-intake-and-malnutrition-september-2006*

Haslett, S. J. & Jones, G. (2008*a*), "Potential for small area estimation and poverty mapping at constituency and at gewog/town level in Bhutan". Feasibility Report, Phases 1 and 2.
URL: *http://documents.wfp.org/stellent/groups/public/documents/ena/wfp207351.pdf*

Haslett, S. J. & Jones, G. (2008*b*), "Potential for small area estimation and poverty mapping in Timor-Leste". Feasibility Report, Phases 1 and 2.
URL: *http://home.wfp.org/stellent/groups/public/documents/ena/wfp207678.pdf*

Haslett, S. J. & Jones, G. (2010), 'Small-area estimation of poverty: The aid industry standard and its alternatives', *Australian & New Zealand Journal of Statistics* **52**(4), 341–362.

Hastie, T., Tibshirani, R. & Friedman, J. (2001), *The Elements of Statistical Learning*, Springer.

Haughton, J. & Khandker, S. R. (2009), 'Handbook on poverty and inequality'.
URL: *http://issuu.com/world.bank.publications/docs/9780821376133*

Healy, A. J., Hitsuchon, S. & Vajaragupta, Y. (2003), 'Spatially disaggregated estimates of poverty and inequality in Thailand', Technical report.
URL: *http://siteresources.worldbank.org/INTPGI/Resources/342674-1092157888460/Healy.DisaggregatedThailand.pdf*

Henderson, C. R. (1975), 'Best linear unbiased estimation and prediction under a selection model', *Biometrics* **31**(2), 423–447.

Hentschel, J., Lanjouw, J. O., Lanjouw, P. & Poggi, J. (2000), 'Combining census and survey data to trace the spatial dimensions of poverty: A case study of ecuador', *World Bank Economic Review* **14**(1), 147–165.

Hinkins, S., Oh, H. L. & Scheuren, F. (1997), 'Inverse sampling design algorithms', *Survey Methodology* **23**, 11–22.

Ho, T. K. (1998), 'The random subspace method for constructing decision forests', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **20**(8), 832–844.

Hoffman, E. B. & Weinberg, E. B. (1998), 'Within cluster sampling'. Paper presented at the American Statistical Association Meetings.

Isidro, M., Haslett, S., Jones, G. et al. (2016), 'Extended structure preserving estimation (espree) for updating small area estimates of poverty', *The Annals of Applied Statistics* **10**(1), 451 – 476.

Jaeckel, L. A. (1972), The infinitesimal jackknife, Technical report, Bell Laboratories, Murray Hill, NJ. Memorandum MM 72-1215-11.

Jamal, H. (2005), 'In search of poverty predictors: The case of urban and rural Pakistan', *The Pakistan Development Review* **44**(1), 37–55.

Japkowicz, N. & Stephen, S. (2002), 'The class imbalance problem: A systematic study', *Intelligent data analysis* **6**(5), 429–449.

Jiang, J. & Lahiri, P. (2006), 'Mixed model prediction and small area estimation', *Test* **15**(1), 1–59.

Jones, J. A. & Waller, N. G. (2013), 'Computing confidence intervals for standardized regression coefficients', *Psychological Methods* **18**(4), 435 – 453.

Kalton, G. (1983), *Introduction to survey sampling*, Sage.

Kam, S.-P., Hossain, M., Bose, M. L. & Villano, L. S. (2005), 'Spatial patterns of rural poverty and their relationship with welfare-influencing factors in bangladesh', *Food Policy* **30**(5-6), 551–567.

Kang, L., Xiong, C. & Tian, L. (2013), 'Estimating confidence intervals for the difference in diagnostic accuracy with three ordinal diagnostic categories without a gold standard', *Computational Statistics & Data Analysis* **68**, 326 – 338.

Keyfitz, N. (1957), 'Estimates of sampling variance where 2 units are selected from each stratum', *Journal of The American Statistical Association* **52**(280), 503–510.

Kish, L. (1990), "Weighting: Why, when and how?". American Statistical Association, Proceedings of the Survey Research Methods Section.
URL: *http://www.amstat.org/sections/srms/Proceedings*

Kish, L. & Frankel, M. R. (1974), 'Inference from complex samples', *Journal of the Royal Statistical Society Series B - Methodological* **36**(1), 1–22.

Kotsiantis, S. B. (2013), 'Decision trees: a recent overview', *Artificial Intelligence Review* **39**(4), 261–283.

Kotsiantis, S., Kanellopoulos, D. & Pintelas, P. (2006), 'Handling imbalanced datasets : A review', *GESTS International Transcations on Computer Science and Engineering* **30**(1), 25–36.

Kovar, J.G., Rao, J.N.K and Wu, C.F.J (1988), 'Bootstrap and other methods to measure errors in survey estimates', *Canadian Journal of Statistics - Revue Canadienne de Statistique* **16**(S), 25–45.

Krewski, D. & Rao, J. (1981), 'Inference from stratified samples - properties of the linearization, jackknife and balanced repeated replication methods', *Annals of Statistics* **9**(5), 1010–1019.

Lanjouw, P., Marra, M. & Nguyen Viet, C. (2013), 'Vietnam's evolving poverty map: Patterns and implications for policy', *World Bank Policy Research Working Paper* (6355).

Last, M., Maimon, O. & Minkov, E. (2002), 'Improving stability of decision trees', *International Journal of Pattern Recognition and Artificial Intelligence* **16**(02), 145–159.

Lee, E. S. & Forthofer, R. N. (2006), *Analyzing Complex Survey Data*, 2nd edn, Sage Publications.

Lee, P. M. (2012), *Bayesian Statistics*, John Wiley & Sons.

Lehtonen, R. & Pahkinen, E. (2004), *Practical Methods for Design and Analysis of Complex Surveys*, John Wiley and Sons.

Lepage, R. & Billard, L. (1992), *Exploring the Limits of the Bootstrap*, John Wiley & Sons.

Levy, P. S. (1979), Small area estimation - synthetic and other procedures, *in* J. Steinberg, ed., 'Synthetic Estimation for Small Areas - Statistical Workshop Papers and Discussion', National Institute on Drug Research, pp. 4 – 19.

Li, R.-H. & Belford, G. G. (2002), Instability of decision tree classification algorithms, *in* 'Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 570–575.

Lohr, S. L. (1999), *Sampling : Design and Analysis*, Duxberry Press.

Madow, W. G. & Madow, L. H. (1944), 'On the theory of systematic sampling, I', *Annals of Mathematical Statistics* **15**, 1–24.

Mahalanobis, P. (1946), 'Recent experiments in statistical sampling in the Indian Statistical Institute', *The Indian Journal of Statistics* **109**, 325 –378.

Maindonald, J. & Braun, W. J. (2010), *Data Analysis and Graphics Using R: an example based approach*, Cambridge University Press.

Malec, D., Davis, W. W. & Cao, X. (1999), 'Model-based small area estimates of overweight prevalence using sample selection adjustment', *Statistics in Medicine* **18**(23), 3189–3200.

Malec, D., Sedransk, J., Moriarity, C. L. & LeClere, F. B. (1997), 'Small area inference for binary variables in the national health interview survey', *Journal of the American Statistical Association* **92**(439), 815–826.

Marker, D. A. (1999), 'Organization of small area estimators using a generalized linear regression framework', *Journal of Official Statistics* **15**(1), 1–24.

McCarthy, P. J. (1969), 'Pseudo-replication: Half samples', *Revue de L'Institute International de Statistique - Review of the International Statistical Institute* **37**(3), 239–264.

McCarthy, P. J. & Snowden, C. B. (1985), 'The bootstrap and finite population sampling', *Hyattsville Md US National Center for Health Statistics [NCHS] 1985.* .

McCullagh, P. & Nelder, J. A. (1989), *Generalized Linear Models*, Chapman and Hall.

McCulloch, C. E., Searle, S. R. & Neuhaus, J. M. (2008), *Generalized, Linear, and Mixed Models*, 2nd edn, John Wiley & Sons.

Milborrow, S. (2015), *rpart.plot: Plot rpart Models. An Enhanced Version of plot.rpart.* R package version 1.5.2.
URL: *http://CRAN.R-project.org/package=rpart.plot*

Miller, R. G. (1964), 'A trustworthy jackknife', *The Annals of Mathematical Statistics* **35**(4), pp. 1594–1605.

Miller, R. G. (1968), 'Jackknifing variances', *Annals of Mathematical Statistics* **39**(2), 567–&.

Miller, R. G. (1974), 'The jackknife - a review', *Biometrika* **61**(1), 1–15.

Minot, N. & Baulch, B. (2005), 'Poverty mapping with aggregate census data: What is the loss in precision?', *Review of Development Economics* **9**(1), 5–24.

Molina, I. & Rao, J. (2010), 'Small area estimation of poverty indicators', *Canadian Journal of Statistics* **38**(3), 369–385.

Morgan, J. N. & Sonquist, J. A. (1963), 'Problems in the analysis of survey data, and a proposal', *Journal of the American Statistical Association* **58**(302), 415–434.

Nathan, G. & Holt, D. (1980), 'The effect of survey design on regression analysis', *Journal of the Royal Statistical Society Series B - Methodological* **42**(3), 377–386.

Nepal Demographic and Health Survey (2001), 'Nepal demographic and health survey 2001'. Ministry of Health, Nepal; New ERA; and ORC Macro.

Neyman, J. (1934), 'On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection.', *Journal of the Royal Statistical Society* **97**(Part 4), 558–625.

Noble, A., Haslett, S. & Arnold, G. (2002), 'Small area estimation via generalized linear models', *Journal of Official Statistics* **18**(1), 45–60.

Ólafsdóttir, K. & Mudelsee, M. (2014), 'More accurate, calibrated bootstrap confidence intervals for estimating the correlation between two time series', *Mathematical Geosciences* **46**(4), 411–427.

Paul, S. & Zhang, X. (2014), 'Small sample GEE estimation of regression parameters for longitudinal data', *Statistics in Medicine* **33**(22), 3869–3881.

Pérez, J. M., Muguerza, J., Arbelaitz, O., Gurrutxaga, I. & Martín, J. I. (2004), Behavior of consolidated trees when using resampling techniques, *in* 'PRIS', pp. 139–148.

Pfeffermann, D. (1993), 'The role of sampling weights when modeling survey data', *International Statistical Review* **61**(2), 317–337.

Pfeffermann, D. (2002), 'Small area estimation - new developments and directions', *International Statistical Review* **70**(1), 125–143.

Plackett, R. L. & Burman, J. P. (1946), 'The design of optimum multifactorial experiments', *Biometrika* **33**(Part 4), 305–325.

Prasad, N. & Rao, J. (1990), 'The estimation of the mean square error of small-area estimators', *Journal of the American Statistical Society* **85**(409), 163–171.

Quenouille, M. H. (1949), 'Problems in plane sampling', *Annals of Mathematical Statistics* **20**(3), 355–375.

Quenouille, M. H. (1956), 'Notes on bias in estimation', *Biometrika* **43**(3-4), 353–360.

Quinlan, J. R. (1986), 'Induction of decision trees', *Machine learning* **1**, 81–106.

Quinlan, J. R. (1990), 'Decision trees and decision-making', *Systems, Man and Cybernetics, IEEE Transactions on Systems* **20**(2), 339–346.

Quintano, C., Castellano, R. & Punzo, G. (2007), 'Estimating poverty in the italian provinces using small area estimation models', *Metodološki zvezki* **4**(1), 37–70.

R Core Team (2015), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *http://www.R-project.org/*

Rao, J. (2003), 'Some new developments in small area estimation', *Journal of the Iranian Statistical Society* **2**(2), 145–169.

Rao, J. (2007), 'Jackknife and bootstrap methods for small area estimation', American Statistical Association. Survey Research Methods Section.

Rao, J. (2011), 'Impact of Frequentist and Bayesian methods on survey sampling practice: A selective appraisal', *Statistical Science* **26**(2, SI), 240–256.

Rao, J. & Choudhry, G. (1995), 'Small area estimation: Overview and empirical study', *in* 'Business Survey Methods', Wiley, pp. 527–542.

Rao, J. N. & Molina, I. (2015), *Small Area Estimation*, 2nd edn, John Wiley & Sons.

Rao, J. & Scott, A. (2000), 'Undoing complex survey data structures: some theory of inverse sampling'. American Statistical Association, Proceedings on the Survey methods Section.

Rao, J., Scott, A. & Benhin, E. (2003), 'Undoing complex survey data structures: some theory and applications of inverse sampling', *Survey Methodology* **29**(2), 107–128.

Rao, J. & Wu, C. (1988), 'Resampling inference with complex survey data', *Journal of the American Statistical Association* **83**(401), 231–241.

Ravallion, M. (1998), 'Poverty lines in theory and in practice'. LSMS Working Paper Number 133.

Ravallion, M. & Bidani, B. (1994), 'How robust is a poverty profile?', *The World Bank Economic Review* **8**(1), 75–102.

Rubin, D. B. (1981), 'The Bayesian bootstrap', *Annals of Statistics* **9**(1), 130–134.

Schall, R. (1991), 'Estimation in generalized linear models with random effects', *Biometrika* **78**(4), 719–727.

Sela, R. J. & Simonoff, J. S. (2012), 'RE-EM trees: a data mining approach for longitudinal and clustered data', *Machine Learning* **86**(2), 169 – 207.
URL: *http://dx.doi.org/10.1007/s10994-011-5258-3*

Sen, A. K. (1985), 'Commodities and capabilities. Lectures in economics: Theory, institutions', *Policy* **7**.

Serlin, R. C. (2000), 'Testing for robustness in Monte Carlo studies', *Psychological Methods* **5**(2), 230.

Shao, J. (2003), 'Impact of the bootstrap on sample surveys', *Statistical Science* **18**(2), 191–198.

Shao, J. & Tu, D. (1995), *The Jackknife and the Bootstrap*, Springer-Verlag.

Singh, A. C., Stukel, D. M. & Pfeffermann, D. (1998), 'Bayesian versus frequentist measures of error in small area estimation', *Journal of the Royal Statistical Society Series B - Statistical Methodology* **60**(2).

Sitter, R. R. (1992), 'Comparing three bootstrap methods for survey data', *The Canadian Journal of Statistics / La Revue Canadienne de Statistique* **20**(2), pp. 135–154.

Skinner, C. J., Holt, D. & Smith, T. F. (1989), *Analysis of Complex Surveys*, Wiley.

Stangenhaus, G. & Narula, S. C. (1991), 'Inference procedures for the {L1} regression', *Computational Statistics & Data Analysis* **12**(1), 79 – 85.

Steel, R. G. D., Torrie, J. H. & Dickey, D. A. (1997), *Principles and Procedures of Statistics : A Biometrical Approach*, McGraw-Hill.

Stein, S. K. (1987), *Calculus and Analytical Geometry*, 4th edn, McGraw-Hill.

Student (1908), 'Probable error of a correlation coefficient', *Biometrika* **6**, 302–310.

Sutradhar, B. C. & Rao, R. P. (2001), 'On marginal quasi-likelihood inference in generalized linear mixed models', *Journal of Multivariate Analysis* **76**(1), 1–34.

Therneau, T., Atkinson, B. & Ripley, B. (2013), *rpart: Recursive Partitioning*. R package version 4.1-3.
URL: *http://CRAN.R-project.org/package=rpart*

Therneau, T. M. (2011). Nabble R forum.
URL: *http://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf*

Therneau, T. M. & Atkinson, E. J. (2000), An introduction to recursive partitioning using the rpart routines, Technical report.
URL: *http://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf*

Therneau, T. M. & Atkinson, E. J. (2013), An introduction to recursive partitioning using the rpart routines, Technical report.
URL: *http://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf*

Thompson, M. E. (1997), *Theory of Sample Surveys*, Chapman and Hall.

Toth, D. & Eltinge, J. L. (2011), 'Building consistent regression trees from complex sample data', *Journal of the American Statistical Association* **106**(496), 1626–1636.

Tukey, J. W. (1958), 'Bias and confidence in not quite large samples', *Annals of Mathematical Statistics* **29**(2), 614.

Turney, P. (1995), 'Bias and the quantification of stability', *Machine Learning* **20**(1-2), 23–33.

United Nations (2016), 'Sustainable development goals'.
URL: *http://www.un.org/sustainabledevelopment/sustainable-development-goals/*

Venables, W. N. & Ripley, B. D. (2002), *Modern Applied Statistics with S*, Springer.

Verma, V. & Betti, G. (2011), 'Taylor linearization sampling errors and design effects for poverty measures and other complex statistics', *Journal of Applied Statistics* **38**(8), 1549–1576.

Villa Juan-Albacea, Z. (2009), Small area estimation of poverty statistics, Technical report. Discussion Paper Series 2009-16.
URL: *http://dirp4.pids.gov.ph/ris/dps/pidsdps0916.pdf*

WFP (2015). World Food Programme.
URL: *http://wfp.org*

Wodon, Q. T. (1997), 'Food energy intake and cost of basic needs: Measuring poverty in Bangladesh', *The Journal of Development Studies* **34**(2), 66–101.

Wolter, K. M. (2007), *Introduction to Variance Estimation*, 2 edn, Springer.

World Bank (2000), 'World development report 2000/2001 : Attacking poverty'.
URL: *https://openknowledge.worldbank.org/handle/10986/11856*

World Bank (2005), *Introduction to Poverty Analysis*. World Bank Institute.
URL: *http://siteresources.worldbank.org/PGLP/Resources/PovertyManual.pdf*

World Bank (2015), 'Mapping poverty'.
URL: *http://go.worldbank.org/9CYUFEUQ30*

Zeger, S. L., Liang, K.-Y. & Albert, P. S. (1988), 'Models for longitudinal data: A generalized estimating equation approach', *Biometrics* **44**(4), pp. 1049–1060.

Appendices

Appendix A

Auxiliary variables

A.1 Household predictors

hh denotes *household*

Name	Type	Description
Poverty	categorical response	notpoor poor
hhsize	integer	household size
hhszsq	integer	$(\text{hhsize} - 6)^2$
skids6	numeric	% kids 0 - 6
skids714	numeric	% kids 7 - 14
samen	numeric	% adult men
entprs	categorical	a. hh head with no ss enterprise b. hh head with trade ss enterprise c. hh head with service/manu ss enterprise
group	categorical	1. Urban Kathmandu 2. Urban other 3. Rural Western mountain & hills 4. Rural Eastern mountain & hills 5. Rural Western terai 6. Rural Eastern terai
hage	categorical	a. hh head aged 18 - 29 b. hh head aged 30 - 44 c. hh head aged 45 - 59 d. hh head aged 60+

Name	Type	Description
hethn	categorical	a. hh head Brahmin/Chhetri b. hh head Terai Middel Caste c. hh head Dalit d. hh head Newar e. hh head Hill Janajatis f. hh head Hill Jajajatis g. hh head Other castes
hfem	binary	no : hh head not female yes : hh head female
hrelig	categorical	a. hh head Hindu b. hh head Buddhist c. hh head Muslim d. hh head Other
huown	binary	house rented or free house owned
hutype	categorical	a. house permanent b. house semi-permanent c. house temporary
nagar	categorical	1. Urban and agri area 0 - 0.1 Ha 2. Urban and agri area 0.1 + Ha 3. Rural and agri area 0 - 0.013 Ha 4. Rural and agri area 0.013 - 0.1 Ha 5. Rural and agri area 0.1 - 0.25 Ha 6. Rural and agri area 0.25 - 0.5 Ha 7. Rural and agri area 0.5 - 1.0 Ha 8. Rural and agri area 1.0 - 2.0 Ha 9. Rural and agri area 2.0+ Ha
numlvst	categorical	1. : Urban and no livestock 2. : Urban and 1+ livestock 3. : Rural and no livestock 4. : Rural and 1 - 2 livestock 5. : Rural and 3 - 5 livestock 6. : Rural and 6+ livestock
numpltry	categorical	1. Urban and no poultry 2. Urban and 1+ poultry 3. Rural and no poultry 4. Rural and 1 - 10 poultry 5. Rural and 11 - 20 poultry 6. Rural and 21+ poultry

Name	Type	Description
remtab	binary	labour abroad none
urbrural	binary	rural urban

A.2 Ward level census means

Name	Type	Description
ckfuel3w	numeric	% cooking fuel LP/gas, ward
ckfuel4w	numeric	% cooking fuel kerosene, ward
edulv3w	numeric	% 15+ pop 5 - 7 yr completed, ward
edulv4w	numeric	% 15+ pop 8 - 10 yr completed, ward
edulv5w	numeric	% 15+ pop 11+ yr completed, ward
elecw	numeric	% lighting fuel electricity, ward
entprs3w	numeric	% hh with service/manu ss enterprise, ward
ftoiletw	numeric	% with flush toilet, ward
hage2w	numeric	% hh head age 30 - 44, ward
hage3w	numeric	% hh head age 45 - 59, ward
hethn2w	numeric	% hh head Terai middle caste, ward
hethn6w	numeric	% hh head Terai Jajajatis, ward
hhsizew	numeric	% average hh size, ward
huown2w	numeric	% house rented or free, ward
hutype2w	numeric	% semi-permanent house, ward
ltfuel2w	numeric	% lighting fuel kerosene, ward
ltfuel3w	numeric	% lighting fuel other, ward
motvehw	numeric	% own a motor vehicle/motor bike
outlfw	numeric	% 15+ pop employed inactive/unemployed, ward
radiow	numeric	% own radio, ward
samenw	numeric	% adult men, ward
skids6w	numeric	% kids 0-6,ward
toilet3w	numeric	% with no toilet, ward
tvw	numeric	% own tv, ward
we_nagw	numeric	% 15+ pop employed in wage - non agri, ward

A.3 VDC level census means

Name	Type	Description
cmortv	numeric	Mortality rate for under 5's, VDC
dmortv	numeric	Mortality rate due to infectious disease, VDC
pch16bpv	numeric	% children (<16) living with both parents, VDC
pch16opv	numeric	% children (<16) living with one parent'relative, VDC
pfhousev	numeric	% hholds with house-owning females, VDC
pflandv	numeric	% hholds with land-owning females, VDC
pflvstv	numeric	% hholds with livestock-owning females, VDC
pschv	numeric	% attending school (6 - 16), VDC

A.4 GIS variables

Name	Type	Description
dhq	numeric	Distance (km) to district headquarters, VDC
meanht	numeric	Mean elevation ('000m) above sea level, VDC
meanslp	numeric	Mean slope (as %), VDC
popdens	numeric	Population density in persons/km2 , VDC
riverpa	numeric	Total length in km of rivers & streams per km2, VDC
roadpai	numeric	Total length in km of motorable road/1000 persons, ilaka
stdht	numeric	Standard deviation of height within VDC in km

Appendix B

Rpart summary output

Refer to Tables 4.3, 4.4 and 4.5, on pages 86 to 87.

B.1 Summary for weighted classification tree model on Replicate 1

```
> summary(Rep1.tree)
Call:
rpart(formula = Poverty ~ ., data = Rep.1, weights = Rep.wt.1,
    method = "class", control = rpart.control(cp = 0, minsplit = 3,
        maxsurrogate = 0, maxdepth = 4))

  n= 326


Variable importance
 skids6w   skids6   bratev avanuwsv hrelig2w  pflvstv ltfuel2w agarea3w
      22       18       12       11        8        8        6        6
  meanht   avmwhv
       4        3


Node number 1: 326 observations,    complexity param=0.07697987
  predicted class=notpoor  expected loss=0.2273548  P(node) =1
    class counts: 251.882 74.1177
   probabilities: 0.773 0.227
  left son=2 (153 obs) right son=3 (173 obs)
  Primary splits:
      skids6w  < 0.1539799  to the left,  improve=14.57771, (0 missing)
      skids6   < 0.1339285  to the left,  improve=13.86400, (0 missing)
      edulv4w  < 0.032498   to the right, improve=13.56240, (0 missing)
      bratev   < 0.0492612  to the left,  improve=12.57349, (0 missing)
      toilet3w < 0.4083555  to the left,  improve=11.70362, (0 missing)


Node number 2: 153 observations,    complexity param=0.03080664
  predicted class=notpoor  expected loss=0.0708985  P(node) =0.4773671
    class counts: 144.588 11.0333
   probabilities: 0.929 0.071
```

```
  left son=4 (151 obs) right son=5 (2 obs)
  Primary splits:
      skids6   < 0.55       to the left,  improve=4.000752, (0 missing)
      agarea3w < 0.364433   to the left,  improve=3.781164, (0 missing)
      cmortv   < 0.2756026  to the left,  improve=2.860790, (0 missing)
      avmwhv   < 8.52177    to the left,  improve=2.001564, (0 missing)
      extendww < 0.1502331  to the left,  improve=2.001564, (0 missing)


Node number 3: 173 observations,    complexity param=0.07697987
  predicted class=notpoor  expected loss=0.3702602  P(node) =0.5226329
    class counts: 107.294 63.0843
   probabilities: 0.630 0.370
  left son=6 (57 obs) right son=7 (116 obs)
  Primary splits:
      skids6  < 0.1339285  to the left,  improve=8.331066, (0 missing)
      edulv4w < 0.0344905  to the right, improve=6.837424, (0 missing)
      bratev  < 0.0492612  to the left,  improve=6.337775, (0 missing)
      hethn5w < 0.3383938  to the left,  improve=6.230215, (0 missing)
      hethn2w < 0.3072236  to the right, improve=5.944799, (0 missing)
```

B.2 Summary of weighted classification tree model on jackknife sample # 25

```
> summary(Model.25)
Call:
rpart(formula = Poverty ~ ., data = JK.25, weights = JK.wt.25,
    method = "class", control = rpart.control(cp = 0, minsplit = 3,
        maxsurrogate = 0, maxdepth = 4))

  n= 324


Variable importance
edulv4w  cyclew  skids6  samenw  dmortv skids6w  hage4w  radiow  hage2w
     24      11      11       8       8       8       7       7       6
avmwhv  remtabw  se_nagw
 5     4       2


Node number 1: 324 observations,    complexity param=0.0968018
  predicted class=notpoor  expected loss=0.2273796  P(node) =1
    class counts: 249.524 73.434
   probabilities: 0.773 0.227
  left son=2 (254 obs) right son=3 (70 obs)
  Primary splits:
      edulv4w < 0.032498   to the right, improve=14.73419, (0 missing)
      skids6w < 0.1539799  to the left,  improve=14.70889, (0 missing)
      skids6  < 0.1339285  to the left,  improve=13.96244, (0 missing)
      bratev  < 0.0492612  to the left,  improve=12.77508, (0 missing)
      popdens < 496.5419   to the right, improve=12.10873, (0 missing)
```

```
Node number 2: 254 observations,    complexity param=0.02746826
  predicted class=notpoor  expected loss=0.1502717  P(node) =0.7932454
    class counts: 217.688 38.4973
   probabilities: 0.850 0.150
  left son=4 (125 obs) right son=5 (129 obs)
  Primary splits:
      skids6   < 0.1339285  to the left,  improve=6.518747, (0 missing)
      bratev   < 0.04115245 to the left,  improve=5.765076, (0 missing)
      skids6w  < 0.1539799  to the left,  improve=5.367890, (0 missing)
      agarea2w < 0.2679917  to the left,  improve=4.525339, (0 missing)
      numpltry splits as  LRRRRL, improve=4.371850, (0 missing)


Node number 3: 70 observations,    complexity param=0.0968018
  predicted class=poor     expected loss=0.4767841  P(node) =0.2067546
    class counts: 31.8363 34.9367
   probabilities: 0.477 0.523
  left son=6 (10 obs) right son=7 (60 obs)
  Primary splits:
      cyclew   < 0.4525998  to the right, improve=6.753387, (0 missing)
      pch16hsv < 0.0080973  to the right, improve=5.504062, (0 missing)
      pflvstv  < 0.1681436  to the left,  improve=5.409905, (0 missing)
      remtabw  < 0.03652935 to the left,  improve=5.235222, (0 missing)
      samen    < 0.1339285  to the right, improve=5.225628, (0 missing)
```

Appendix C

R code

C.1 Code for improve function

Refer to Section 4.9

```
Improve <- function(left.poor,left.notpoor,right.poor,right.notpoor) {

 left.total <- left.poor + left.notpoor
 right.total <- right.poor + right.notpoor

 root.poor <- left.poor + right.poor
 root.notpoor <- left.notpoor + right.notpoor
 root.total <- root.poor + root.notpoor

 IG <- 2*(root.poor/root.total)*(root.notpoor/root.total)
     - 2*( (left.total/root.total)*(left.poor/left.total)*
        (left.notpoor/left.total)
     + (right.total/root.total)*(right.poor/right.total)*
        (right.notpoor/right.total) )
 Improve <- root.total * IG

 return(Improve)
}
```

C.2 Code for simulations using a classification tree

```
## Code to set up data for simulations with clustering, use large K as
##              function argument because want to use k in naming the dataset
## 21/03/2014  extend analysis to 1000 CI's
##              restrict to n = 3000 only  (not a designed experiment now)
##  29/03/2014   Amend to create new census dataset for each survey dataset,
                                                        i.e. 1000 runs
##   05/05/2014   Adjust code to remove loops where ever possible
##   06/06/2014  included code to extract BS percentiles for percentile CI's
##                also  record how many survey clusters have value 0, 1
##   09/06/2014   add loops to run through k = 0.08 to 0.24
```

```
##   12/07/2014   changed order of code : put code for creating bag of
          residuals  AFTER code to put cluster effects into census log(exp)
##    so that true P0's will be the same for BS and Full tree estimation

###  02/09/2014   change code to create parametric census cluster effects
##    (perturbations on predictions) use glmm to find variance of clusters

#######################################################

###  1.   Set up factor level for split and survey size, cluster effect and
            perturbation values are function arguments
          ###  specify cluster size for survey size 3000

library(boot)
library(rpart)
library(lme4)

SPLIT <- 20
n <- 3000
C <- 250

###  residual se  for  simulated  datasets  approx  0.5

################################################

# 2.  Bring in actual survey data for simulation

NLSS.ELL <-  read.csv(file = "NLSS_ELL.csv")

#################################

#  3.  bring in square root of covariance matrix for NLSS survey data

Half.sigma <- read.csv(file="HalfSigma.csv")

Root.sigma <- as.matrix(Half.sigma)

#################################

###  4.  Create vector of means of NLSS survey variables  (Removed loop)

NLSS.mean <- as.vector(apply(NLSS.ELL,2,mean))

#####################################################

###   5.  Set seeds -
##        21/03/2014  for  1000 CI's need 1000 simulated survey datasets,
                         so 1000 seeds
##        28/03/2014  need to simulate survey data before census, to
                maintain consistency with previous analysis

set.seed(1837)
rand.seed = round(10000*runif(1000))    ## doing 1000 CI's

#####################################

##  6.  Set up function to simulate 1000 survey and census datasets
```

```
Sim.Clust <- function(K,Survey,RootSigma,Means,RSeed){

###  add cluster effects to survey & census log(exp) variables,
###  model soft BS estimates, survey size 3000, census size 6000, split = 20

TrueP0 <- NULL     ##  P0 of census dataset
P0 <- NULL         ##  P0 of survey dataset
Soft <- NULL       ##  mean of 100 BS soft predictions of P0
Soft.se <- NULL    ##  BS se of 100 BS soft predictions of P0
S.size <- NULL
Split <- NULL
Cl.no <- NULL
Cl.se <- NULL
QuantBS <- NULL  ## percentiles of BS distribution to build percentile CI's
PSU.se <- NULL
Leaf.propn <- NULL

pvec <- c(0.025,0.05,0.1,0.16,0.84,0.9,0.95,0.975)

##  Set up variables to create cluster index for all hh's in survey
                                                (250 clusters)

Cluster.ID <- rep(1:250,each=12)

##  creates variable to indicate which cluster a hh belongs to
##  Cluster.ID  :  12  1's, followed by 12  2's, then 12  3's, ..etc


######  Set up loop for 1000 survey & census datasets

for (j in RSeed) {

set.seed(j)

#################

#####   7.  Set up simulated survey dataset

# set up simulated survey data each time, size n = 3000 observations,
                                                    26 variables

X <- matrix(rnorm(3000*26), ncol=26)

S.size.j <- n
S.size <- c(S.size, S.size.j)

X.temp <-  as.data.frame(X %*% RootSigma)

X.sim <- X.temp + matrix(Means, nrow=3000, ncol=length(Means),byrow=T)

colnames(X.sim) <- dimnames(Survey)[[2]]

######################

#  8.  Simulate prediction data, size = 6000

Z <- matrix(rnorm(6000*26), ncol=26)
```

```
Z.temp <-  as.data.frame(Z %*% RootSigma)

Z.sim <- Z.temp + matrix(Means, nrow=6000, ncol=length(Means), byrow=T)
colnames(Z.sim) <- dimnames(Survey)[[2]]

####     9.   add cluster effects to survey response,
                      250 clusters of size 12 = 3000 hh's

B <- rep(rnorm(250), each=12)   ## use seeds here

ClustSEff <-  K * B       ### cluster survey effect
                                    -  S in name means survey

 log_exp_c <- X.sim$log_exp + ClustSEff

##  create poverty variable

Pov <- log_exp_c

Pov.line <- 8.948423

temp.survey <- (Pov < Pov.line)  ##  vector of logical values

P0.j <- mean(temp.survey)      ## treats true = 1  and false = 0

P0 <- c(P0, P0.j)

Poor <- (temp.survey + 0)     #  converts logical to integer

###  set up survey dataset (with clustering) to build model

Poverty <- as.factor(Poor)

Survey.sim <- data.frame(Poverty,X.sim[,-1])

#####################################################

###   10.   Add cluster effect to census data, size 6000
##         average Nepal PSU size is 150,  so  6000 / 150  =  40 clusters

A <- rep(rnorm(40), each=150)     ## use seeds here

ClustEff <-  K * A

log_exp_cl <- Z.sim$log_exp + ClustEff

########################################

###   11.  find true  P0 for census

PovCensus <- log_exp_cl

Pov.line <- 8.948423

TrueP0.j <- mean(PovCensus < Pov.line)  ## true value of census poverty
                           incidence for each iteration
TrueP0 <- c(TrueP0,TrueP0.j)
```

```
##  don't need poverty variable for prediction, only for comparison

Predict.sim <- Z.sim[,-1]

################################################

######   12.  Set up bag of cluster residuals

Model.Sim <- rpart(Poverty ~. , data= Survey.sim, method = "class",
 control=rpart.control(cp=0, minsplit=20, maxcompete=0, maxsurrogate=0,
   maxdepth=5))

Leaf <- as.factor(as.vector(Model.Sim$where))

L1 <- length(unique(Leaf))

L2 <- length(which(Model.Sim$frame$var=="<leaf>"))

Leaf.propn.j <- L1 / L2
Leaf.propn <- c(Leaf.propn,Leaf.propn.j)

####   find PSU variance

PSU <- as.factor(Cluster.ID)

Sim.glmm <- glmer(Poverty ~ Leaf + (1|PSU), family = binomial, nAGQ = 1)

PSU.se.j <- Sim.glmm@theta
PSU.se <- c(PSU.se,PSU.se.j)

################################################

###  13.    model  soft cluster BS estimates on n = 3000,
                              (check BS index), split = 20

SoftEst.j <- NULL

## loop over 100 BS samples

for (i in 1:100) {

     Cstar.clust <- rnorm(40, mean = 0, sd = PSU.se.j)
       ##  take sample of cluster residuals to add to predictions,
                                       one residual for each census PSU
     Cstar.hh <-  rep(Cstar.clust,each=150)
      ## vector of cluster effects, same value for each hh in same cluster

     Index <- sample(1:250,250, replace=TRUE)
             ## to take BS sample of survey clusters to build a tree

##  Index is then a BS sample (with replacement)
            from values of variable for number of Clusters (1 to 250),
##  sample size is the same as number of clusters
                          (C = 250 for  survey n = 3000)
##  So index represents a BS sample of the clusters

     BS.index <- NULL
```

```
      for (h in 1:length(Index)){
         ##  find hh's which are in the bootstrapped clusters
         ## values of BS.index are the positions of all hh's in clusters
                                        selected (with replacement)
           g <- Index[h]
           HH.ID <- which(Cluster.ID == g)
           BS.index <- c(BS.index,HH.ID)
       }

     BS.i <- Survey.sim[BS.index,]     # takes a BS sample
     Model.i <- rpart(Poverty ~. , BS.i, method = "class",
         control=rpart.control(cp=0, minsplit=SPLIT, maxcompete=0,
         maxsurrogate=0, maxdepth=5))

##  compute soft predictions
    Soft.i <- as.vector(predict(Model.i, Predict.sim)[,2])

##  add  cluster residuals to hh's on logit scale then do inverse logit

    L.i <- logit(Soft.i) + Cstar.hh

    SoftEst.i   <- mean(inv.logit(L.i))

    SoftEst.j   <- c(SoftEst.j, SoftEst.i)
                          #  only one estimate per iteration
  }

SoftPov.j <- as.vector(SoftEst.j)

Soft.j <- mean(SoftPov.j)
Soft.se.j <- sd(SoftPov.j) * sqrt(99/100)


QuantBS.j <- quantile(SoftPov.j,pvec)   ##  creates a row of 8
                  percentiles from BS distribution of P0 predictions

## save minsplit and other values and BS statistics

Split.j <- Model.i$control$minsplit
Split   <- c(Split,Split.j)

Soft <- c(Soft, Soft.j)   ## vector of soft estimates from each
                                                 survey sample

Soft.se <- c(Soft.se, Soft.se.j) ## vector of soft s.e.'s from each
                                                  survey sample

QuantBS <- rbind(QuantBS,QuantBS.j)     ## creates matrix of percentiles,
                                                      8 columns wide

Cl.no <- c(Cl.no,C)  ## number of clusters in survey data
Cl.se <- c(Cl.se,K)   ##  record se of cluster effects

}

rownames(QuantBS) <- NULL

BSPreds <- cbind(TrueP0,P0,Soft,Soft.se,S.size,Split,Cl.no,Cl.se,
```

```
                                          QuantBS,PSU.se,Leaf.propn)

dateToday <- Sys.Date()

write.csv(BSPreds, file=paste(dateToday,"_BSclust","_n3000","_1000CI's_"
       ,"k",K,"_simCensus","_ParametricPerturb","_nAGQ1.csv", sep=""),
        row.names=FALSE)

}

################################################

k.vals <- c(0.16,0.20,0.24)

for (Kval in k.vals)  {

    Sim.Clust(Kval,NLSS.ELL,Root.sigma,NLSS.mean,rand.seed)

}
```

C.3 Code for regression tree estimates for a district in Nepal

```
###  H:\R\Regn tree Px with stratification in Nepal\07-03-2016_BS soft ests
###  of Px for Nepal,  par perturbs, stratification, ELL vars,
                                          full tree ests,  cp 0.001.r


## 01/08/2014  code to find bootstrap soft estimates for Nepal data,
        ##   restricted to ELL model variables only

## 16/09/2014  amend for parametric perturbations

##  03/02/2016  amend for stratification

###  07/03/2016   amend for full tree providing point estimates as well as
                  BS estimates,  compare BS with full


##  29/02/2016  amend for cp = 0.001 as pruning parameter

##  07/03/2016   amend for Px from regression tree  -  uses unweighted
                                       full tree for sigma^2_c estimation

####################

library(rpart)
library(boot)
library(lme4)


###  1.  bring in datasets

NLSS.data <- read.csv(file="NLSS_ELL_regn.csv")
                                       ## contains log-exp & Poverty
```

```
str(NLSS.data[1:5])

### remove poverty

NLSS <- NLSS.data[,-2]

str(NLSS)

NLSS$nagar      <- as.factor(NLSS$nagar)
NLSS$numpltry   <- as.factor(NLSS$numpltry)
NLSS$group      <- as.factor(NLSS$group)

District <- read.csv(file="Kavre_ELL_BS.csv")

str(District)

District$group     <- as.factor(District$group)
District$numpltry  <- as.factor(District$numpltry )
District$nagar     <- as.factor(District$nagar)


##################################

#####  2.   set up weights and stratum, cluster and hh variables
                                              for building trees

NLSS.f <- read.csv(file="NLSS_f.csv")

str(NLSS.f[1:10])

indwght <- NLSS.f$indwght

Tree.wt <- indwght * 3912 / sum(indwght)  ## standardise weights
                                             to equal sample size

sum(Tree.wt)



##  for strata: stratum size, bootstrap sample size, stratum ID,
                                           total stratum weights


StratumID <- NLSS.f$stratum

StratumSize <- as.vector(table(StratumID))/12   ## number of psu's
                                          in each stratum

BSsize <- StratumSize - 1

StratumName <- c("Mountains", "Rural Hills", "Rural Tarai", "Urban Hills",
                                           "Urban Kath.", "Urban Tarai")


StratumWt <- as.vector(tapply(Tree.wt,StratumID,sum))
 ## total stratum weight - use weights scaled to equal sample size

ClusterID <- NLSS.f$psu    ## NLSS data has 326 ilaka, each with 12 hh's
```

```
HH.ID <- NLSS.f$WWWHH

##################################

####   3.  set up variables for predictions

Kavre    <- read.csv(file="Pred_Kavre.csv")
                  ## need this to get batch & psu variables for Kavre

batchid  <- Kavre$batchid  ## each psu is a ward/subward within an ilaka
PSU.ID   <- batchid %% 10000    # unique PSU/cluster ID number for each hh,
        ## needed because hh's in same cluster get same cluster residual
                                  added to their prediction

hh.num   <-  as.vector(table(PSU.ID))  ## vector of number of hh's
                                         in a ward/PSU/cluster


HHsize   <- District$hhsize  ## District is Kavre for only ELL variables,
                                                  and ilaka variable
ilaka    <- District$ilaka


Ilaka.Wt <- rep(0,length(HHsize))
N <- length(unique(ilaka))
N


for (i in 1:N) {
Ilaka.Wt[ilaka==i] <- sum(HHsize[ilaka==i])
}


Census <- District[,-1]
                    ## remove ilaka variable from prediction dataset
str(Census)

###################################

##  define poverty line for soft predictions

Pov.line <- 8.948423
ExpPov.line <- exp(Pov.line)    ##  Page xiii  in Nepal report


#########################################

####  4.  Set up bag of cluster residuals


##  find the terminal node (leaf) for each hh in NLSS survey data

## use unweighted model - have done this for simulations

Model.NLSS <- rpart(log_exp ~. , data= NLSS, method = "anova",
   control=rpart.control(cp=0.001, minsplit=20, maxcompete=0,
     maxsurrogate=0))
```

```
Leaf <- as.factor(as.vector(Model.NLSS$where))

L1 <- length(unique(Leaf))

L2 <- length(which(Model.NLSS$frame$var=="<leaf>"))

Leaf.propn <- L1 / L2
Leaf.propn


####   find PSU variance

Cluster.ID <- rep(1:326,each=12)
  ## use survey data to estimate cluster variance and generate
        cluster residuals

PSU <- as.factor(Cluster.ID)

NLSS.lmm <- lmer(NLSS$log_exp ~ Leaf + (1|PSU))

PSU.se <- sqrt(unlist(VarCorr(NLSS.lmm)))
PSU.se
## this value depends on Model.NLSS because the model determines
                               the terminal node for each hh

#########################

##   set up cluster residuals to append to census predictions


Cstar.clust <- rnorm(816, mean = 0, sd = PSU.se)
                            ## 816 PSU's (clusters - wards) in census data
length(Cstar.clust)

Cstar.hh  <- rep(Cstar.clust,hh.num)
            ##  same cluster residual for each hh in a cluster(PSU/ward)
length(Cstar.hh)

length(ilaka)

################################


########   5. run code for BS soft estimates    (rpart called above)


set.seed(2562)   ##  need seed for BS sampling & cluster residual sampling


P0SoftEst <- NULL
P0FullEst <- NULL
P0Soft.se <- NULL

P1SoftEst <- NULL
P1FullEst <- NULL
P1Soft.se <- NULL
```

```
P2SoftEst <- NULL
P2FullEst <- NULL
P2Soft.se <- NULL


for (i in 1:100) {    ## loop over 100 BS samples  -  i denotes BS loop

       BSindex <- NULL     ## draw separate BS sample in each stratum
       ModelWt.i <- NULL    ## model weights have to be adjusted
                                          within each stratum

       for (s in 1:6)  {   ##  loop over 6 stratum
                                  - s denotes stratum loop

       Stratum <- StratumName[s]

       ClusterID.s <-  ClusterID[StratumID == Stratum]
                ## identifies the psu of all hh's in Stratum s

       Cluster.s <- unique(ClusterID.s)
                ## identifies just the psu's themselves in Stratum s

       ###  create index variable for bootstrapped clusters (psu's)

       BSsize.s <- BSsize[s]


       Index.s <- sample(Cluster.s, BSsize.s, replace = TRUE)
        ## BS sample of clusters in Stratum s, sample size 1 less than
                     stratum size (i.e. number of psu's in stratum)

##  Index.s is then a BS sample (with replacement)  from values of
          the variable identifying the clusters in stratum  s,
##  sample size is one less than the number of clusters in the stratum
##  So index.s represents a BS sample of the clusters in stratum s


     BSindex.s <- NULL

      for (h in 1:length(Index.s)){
       ## to find hh's which are in the bootstrapped clusters for stratum s
       ## values of BS.index are the positions of all hh's in clusters
                    selected (with replacement)
       ##  h  denotes loop finding hh's within clusters

           g <- Index.s[h]
           HH.index  <- which(ClusterID == g)
           BSindex.s <- c(BSindex.s,HH.index)
             ## BSindex.s indentifies the hh's in the stratum BS sample

      }  ## end of h loop

     BSwt.s <- Tree.wt[BSindex.s]
     ##  vector of sampling weights for all hh's in BS sample for stratum s

     ModelWt.s <- BSwt.s * StratumWt[s] / sum(BSwt.s)
     ## rescale so weights of hh's in BS sample sum to stratum total weight
```

```
     ModelWt.i <- c(ModelWt.i, ModelWt.s)
         ## combine weights from each stratum BS sample

     BSindex <- c(BSindex,BSindex.s)
        ###  combine BS samples from each stratum to form a single
              BS replicate for the ith iteration

}    ###  end of stratum loop


###  now use i th BS replicate and associated weights to build model

     BS.i <- NLSS[BSindex,]
     # creates a BS replicate which combines BS samples from each stratum

     Model.i <- rpart(log_exp ~. , BS.i, weights = ModelWt.i,
        method = "anova", control=rpart.control(cp=0.001, minsplit=20,
        maxcompete=0, maxsurrogate=0))

##   compute hard tree predictions to be used for soft predictions

    Pred.i <- as.vector(predict(Model.i, Census))
        ## prediction for each hh in census
        ##  default prediction is mean of log(exp) at node


##   add  cluster residuals to predictions at hh level

     Est.i <- Pred.i + Cstar.hh
         ## this is mu* in write up
         ## cluster residuals are normally distributed
                           - don't need logit function here

##   find sigma at leaf for each predicted hh,
                  for soft estimates which are probabilities

    SS <- Model.i$frame$dev
    N  <- Model.i$frame$n
    sigma <- sqrt(SS/(N-1))

    Model.i$frame$yval <- sigma
         ## makes leaf s.d. the predicted quantity
    sigma.hh <- as.vector(predict(Model.i,Census))
         ## vector of leaf s.d.'s for each hh given its leaf

##   compute i th bootstrap soft predictions for Px

    P0Prob.i  <- pnorm(Pov.line, Est.i, sigma.hh)
      ## Pov.line is quantile for pnorm function, Est.i is hh mu*
            & sigma.hh from leaf that hh is in
    P0SoftWtd.i  <- (P0Prob.i * HHsize) / Ilaka.Wt
     ##  ( =  (y_i  x  weight_i) / sum over ilaka(weight.i)  )
    P0SoftEst.i  <- tapply(P0SoftWtd.i,ilaka,sum)
     ##  creates a small area prediction for each ilaka for one BS iteration,
                a vector of 12 values per BS iteration
    P0SoftEst    <- rbind(P0SoftEst, P0SoftEst.i)
     ##  creates a matrix of  100 rows (BS iterations) & 18 columns (ilakas)
```

```
    P1Prob.i  <- pnorm(Pov.line, Est.i, sigma.hh)
       - (exp(Est.i + (sigma.hh^2)/2) / ExpPov.line)
           * pnorm(Pov.line, Est.i + sigma.hh^2, sigma.hh)
    P1SoftWtd.i  <- (P1Prob.i * HHsize) / Ilaka.Wt
       ##  ( =  (y_i  x  weight_i) / sum over ilaka(weight.i)  )
    P1SoftEst.i  <- tapply(P1SoftWtd.i,ilaka,sum)
    ##  creates a small area prediction for each ilaka for one BS iteration,
            a vector of 12 values per BS iteration
    P1SoftEst    <- rbind(P1SoftEst, P1SoftEst.i)
    ##  creates a matrix of  100 rows (BS iterations) & 18 columns (ilakas)


    P2Prob.i  <- pnorm(Pov.line, Est.i, sigma.hh)
        - 2*(exp(Est.i + (sigma.hh^2)/2) / ExpPov.line)
          * pnorm(Pov.line, Est.i + sigma.hh^2, sigma.hh)
             + ( exp( 2*(Est.i + sigma.hh^2) ) / ExpPov.line^2 )
               * pnorm(Pov.line, Est.i + 2*sigma.hh^2, sigma.hh)
    P2SoftWtd.i  <- (P2Prob.i * HHsize) / Ilaka.Wt
      ##  ( =  (y_i  x  weight_i) / sum over ilaka(weight.i)  )
    P2SoftEst.i  <- tapply(P2SoftWtd.i,ilaka,sum)
    ##  creates a small area prediction for each ilaka for one BS iteration,
          a vector of 12 values per BS iteration
    P2SoftEst    <- rbind(P2SoftEst, P2SoftEst.i)
    ##  creates a matrix of  100 rows (BS iterations) & 18 columns (ilakas)


}   ## end of loop for 100 Bootstrap replicates



#######    BS soft point estimates and standard error


P0Soft <- apply(P0SoftEst,2,mean)
            ## 2 means compute average down columns, for each ilaka
P0Soft

P0Soft.se <- apply(P0SoftEst,2,sd)
            ## 2 means compute s.d. down columns
P0Soft.se


P1Soft <- apply(P1SoftEst,2,mean)
P1Soft

P1Soft.se <- apply(P1SoftEst,2,sd)
P1Soft.se


P2Soft <- apply(P2SoftEst,2,mean)
P2Soft

P2Soft.se <- apply(P2SoftEst,2,sd)
P2Soft.se
```

```
###  obtain full tree point estimates  -  don't need cluster BS,
             cluster residuals or stratification
## these include weights in model, so should have weighted model
               for estimation of sigma^2_c

Model.Full <- rpart(log_exp ~. , NLSS, weights = Tree.wt,
    method = "anova", control=rpart.control(cp=0.001,
       minsplit=20, maxcompete=0, maxsurrogate=0))


## functions for soft estimates from full tree
  - one estimate per HH, then aggregate across ilaka,
        do weighted sum
## functions in separate file to save space

## Pov.line is in log(exp) scale

P0FullEst  <- Soft.regn.P0(Model.Full,Census,Pov.line)
P0FullWtd  <- (P0FullEst * HHsize) / Ilaka.Wt
       ##  ( =  (y_i  x  weight_i) / sum over ilaka(weight.i)  )
P0Full     <- tapply(P0FullWtd,ilaka,sum)
              ##  creates a small area prediction for each ilaka,
                    a vector of 18 values
P0Full

P1FullEst  <- Soft.regn.P1(Model.Full,Census,Pov.line)
P1FullWtd  <- (P1FullEst * HHsize) / Ilaka.Wt
P1Full     <- tapply(P1FullWtd,ilaka,sum)
P1Full

P2FullEst  <- Soft.regn.P2(Model.Full,Census,Pov.line)
P2FullWtd  <- (P2FullEst * HHsize) / Ilaka.Wt
P2Full     <- tapply(P2FullWtd,ilaka,sum)
P2Full
```

Appendix D

Mathematical derivations of soft estimators for poverty gap and poverty severity

D.1 Derivation of a soft estimator for poverty gap

Poverty gap describes the average level of poverty for those households below the poverty line. The formula for poverty gap corresponds to the FGT equation with $a = 1$, as follows,

$$P_1 \;=\; \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\mathrm{z}-\mathcal{E}_i}{\mathrm{z}}\right)\;\cdot\;\mathrm{I}\left(\mathcal{E}_i<\mathrm{z}\right)\;. \tag{D.1}$$

The response variable for the regression tree model is $Y = \log(\mathcal{E})$, where $\mathcal{E}$ is per capita expenditure. But the formula for poverty gap, Equation (D.1), is a function of $\mathcal{E}$. Consider the function $g(\mathcal{E})$, where

$$g(\mathcal{E}) \;=\; \mathrm{I}\left(\mathcal{E}<\mathrm{z}\right)\;\cdot\;\frac{(\mathrm{z}\;-\;\mathcal{E})}{\mathrm{z}}\,,$$

with z denoting the poverty line. We develop a "soft" tree estimate for poverty gap by taking the expectation of $g(\mathcal{E})$,

$$\begin{aligned}
\mathrm{E}\left[g(\mathcal{E})\right] \;&=\; \mathrm{E}\left[\mathrm{I}\left(\mathcal{E}<\mathrm{z}\right)\;\cdot\;\frac{(\mathrm{z}\;-\;\mathcal{E})}{\mathrm{z}}\right]\\
&=\; \int_{-\infty}^{\mathrm{z}}\frac{(\mathrm{z}\;-\;\varepsilon)}{\mathrm{z}}\,f(\varepsilon)\,d\varepsilon\\
&=\; \int_{-\infty}^{\mathrm{z}}\left(1\;-\;\frac{\varepsilon}{\mathrm{z}}\right)\,f(\varepsilon)\,d\varepsilon\\
&=\; \mathrm{P}\left[\mathcal{E}<\mathrm{z}\right]\quad-\quad\frac{1}{\mathrm{z}}\int_{-\infty}^{\mathrm{z}}\varepsilon\,f(\varepsilon)\,d\varepsilon\,,
\end{aligned} \tag{D.2}$$

where $f(\varepsilon)$ denotes the density function of the expenditure variable $\mathcal{E}$. To facilitate the formulation of a soft estimator we consider a change of variable, to use Y=log($\mathcal{E}$), since Y has a normal distribution, $Y \sim N(\mu, \sigma^2)$. Since $\mathcal{E} = \mathrm{e}^Y$, the expectation, Equation (D.2), becomes,

$$\begin{aligned} \mathrm{P}\left[\mathcal{E} < \mathrm{z}\right] \;-\; \frac{1}{z}\int_{-\infty}^{\mathrm{z}} \varepsilon\, f(\varepsilon)\, d\varepsilon \;&=\; \mathrm{P}\left[Y < \log(\mathrm{z})\right] \;-\; \frac{1}{\mathrm{z}}\int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^{y}\, f(y)\, dy \\ &=\; \mathrm{P}\left[Y < \log(\mathrm{z})\right] \;-\; \frac{1}{\mathrm{z}}\int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^{y}\, \frac{1}{\sqrt{2\pi\sigma^2}}\; \mathrm{e}^{-\frac{(y-\mu)^2}{2\,\sigma^2}}\, dy \\ &=\; \mathrm{P}\big[Y < \log(\mathrm{z})\big] \;-\; \frac{1}{\mathrm{z}}\,\frac{1}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^{y}\,\cdot\,\mathrm{e}^{-\frac{(y-\mu)^2}{2\,\sigma^2}}\, dy\,. \end{aligned} \tag{D.3}$$

The exponential terms inside the integral in Equation D.3 can be simplified;

$$\begin{aligned} \mathrm{e}^{\,y}\,\mathrm{e}^{-\dfrac{(y-\mu)^2}{2\sigma^2}} \;&=\; \mathrm{e}^{-\dfrac{-\,2\sigma^2 y + (y-\mu)^2}{2\sigma^2}} \\ &=\; \mathrm{e}^{-\dfrac{-\,2\sigma^2 y + \left(y^2 - 2\mu y + \mu^2\right)}{2\sigma^2}} \\ &=\; \mathrm{e}^{-\dfrac{y^2 - 2\mu y - 2\sigma^2 y + \mu^2}{2\sigma^2}} \\ &=\; \mathrm{e}^{-\dfrac{y^2 - 2y(\mu+\sigma^2) + \mu^2}{2\sigma^2}}\,. \end{aligned} \tag{D.4}$$

The process continues by completing the square in the numerator of the exponential exponent in Equation D.4

$$\begin{aligned} \left[y \;-\; \left(\mu+\sigma^2\right)\right]^2 \;&=\; y^2 - 2y\left(\mu+\sigma^2\right) + \left(\mu+\sigma^2\right)^2 \\ &=\; y^2 - 2y\left(\mu+\sigma^2\right) + \mu^2 + 2\mu\sigma^2 + \sigma^4\,. \end{aligned}$$

Therefore, after completing the square the numerator of the exponential term can be expressed as,

$$y^2 \;-\; 2y\left(\mu+\sigma^2\right) \;+\; \mu^2 \;=\; \left[y \;-\; \left(\mu \;+\; \sigma^2\right)\right]^2 \;-\; \sigma^2\left(2\mu \;+\; \sigma^2\right)\,.$$

So, the exponential exponent in Equation D.4 becomes,

$$\begin{aligned} \mathrm{e}^{-\frac{y^2 - 2y(\mu+\sigma^2) + \mu^2}{2\sigma^2}} &= \mathrm{e}^{-\frac{\left[y - \left(\mu+\sigma^2\right)\right]^2 \; - \; \sigma^2\left(2\mu+\sigma^2\right)}{2\sigma^2}} \\ &= \mathrm{e}^{-\frac{\left[y - \left(\mu+\sigma^2\right)\right]^2}{2\sigma^2}} \; \cdot \; \mathrm{e}^{\frac{\sigma^2\left(2\mu+\sigma^2\right)}{2\sigma^2}} , \end{aligned}$$

which gives,

$$\mathrm{e}^{-\frac{y^2 - 2y(\mu+\sigma^2) + \mu^2}{2\sigma^2}} = \mathrm{e}^{-\frac{\left[y - \left(\mu+\sigma^2\right)\right]^2}{2\sigma^2}} \; \cdot \; \mathrm{e}^{\left(\mu + \frac{\sigma^2}{2}\right)} .$$

Hence, we can rewrite Equation D.3 as,

$$\begin{aligned} & \mathrm{P}\left[Y < \log(\mathrm{z})\right] \; - \; \frac{1}{\mathrm{z}} \int_{-\infty}^{\log(\mathrm{z})} \frac{1}{\sqrt{2\pi\sigma^2}} \, \mathrm{e}^{-\frac{\left[y-\left(\mu+\sigma^2\right)\right]^2}{2\,\sigma^2}} \; \cdot \; \mathrm{e}^{\left(\mu+\frac{\sigma^2}{2}\right)} \, dy \\ = \quad & \mathrm{P}\left[Y < \log(z)\right] \; - \; \frac{\mathrm{e}^{\left(\mu+\frac{\sigma^2}{2}\right)}}{z} \int_{-\infty}^{\log(\mathrm{z})} \frac{1}{\sqrt{2\pi\sigma^2}} \, \mathrm{e}^{-\frac{\left[y-\left(\mu+\sigma^2\right)\right]^2}{2\,\sigma^2}} \, dy \, . \end{aligned}$$

Thus Equation (D.3), representing a soft estimate for poverty gap, can be expressed in terms of two probabilities,

$$\mathrm{P}\left[Y < \log(\mathrm{z})\right] \quad - \quad \frac{\mathrm{e}^{\left(\mu+\frac{\sigma^2}{2}\right)}}{z} \, \mathrm{P}\left[Y^* < \log(\mathrm{z})\right] \, , \tag{D.5}$$

where $Y \sim N(\mu, \sigma^2)$ and $Y^* \sim N(\mu + \sigma^2, \sigma^2)$.

D.2 Derivation of a soft estimator for poverty severity

The response variable for the regression tree model is $Y = \log(\mathcal{E})$, where $\mathcal{E}$ is per capita expenditure. But my poverty measure, poverty severity (P_2), is a function of $\mathcal{E}$.

$$P_2\left(\mathcal{E}\right) \; = \; \mathrm{I}\left(\mathcal{E} < \mathrm{z}\right) \; \cdot \; \left(\frac{\mathrm{z} \; - \; \mathcal{E}}{\mathrm{z}}\right)^2 .$$

We develop a "soft" tree estimate for poverty severity by taking the expectation of $h(E)$,

$$\begin{aligned}
\mathrm{E}\left[P_2\left(\mathcal{E}\right)\right] &= \mathrm{E}\left[\mathrm{I}\left(\mathcal{E} < \mathrm{z}\right) \cdot \left(\frac{\mathrm{z} - \mathcal{E}}{\mathrm{z}}\right)^2\right] \\
&= \int_{-\infty}^{\mathrm{z}} \left(\frac{\mathrm{z} - \varepsilon}{\mathrm{z}}\right)^2 f(\varepsilon)\, d\varepsilon \\
&= \int_{-\infty}^{\mathrm{z}} \left(\frac{\mathrm{z}^2 - 2\mathrm{z}\varepsilon + \varepsilon^2}{\mathrm{z}^2}\right) f(\varepsilon)\, d\varepsilon \\
&= \int_{-\infty}^{\mathrm{z}} \left(1 - \frac{2\varepsilon}{\mathrm{z}} + \frac{\varepsilon^2}{\mathrm{z}^2}\right) f(\varepsilon)\, d\varepsilon \\
&= \mathrm{P}\left[\mathcal{E} < \mathrm{z}\right] - \frac{2}{\mathrm{z}} \int_{-\infty}^{\mathrm{z}} \varepsilon\, f(\varepsilon)\, d\varepsilon + \frac{1}{\mathrm{z}^2} \int_{-\infty}^{\mathrm{z}} \varepsilon^2\, f(\varepsilon)\, d\varepsilon\,, \qquad \text{(D.6)}
\end{aligned}$$

where $f(\varepsilon)$ denotes the density function of the expenditure variable $\mathcal{E}$. As with the soft estimator for poverty gap we we consider a change of variable, to use $Y = \log(\mathcal{E})$, since Y has a normal distribution, $Y \sim N(\mu, \sigma^2)$. Since $\mathcal{E} = \mathrm{e}^Y$, then $\mathcal{E}^2 = \mathrm{e}^{2Y}$. Thus Equation (D.6) can be written in terms of the variable Y instead of $\mathcal{E}$, as follows,

$$\mathrm{E}\left[P_2\left(Y\right)\right] = \mathrm{P}\left[Y < \log(\mathrm{z})\right] - \frac{2}{\mathrm{z}} \int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^y\, f(y)\, dy + \frac{1}{\mathrm{z}^2} \int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^{2y}\, f(y)\, dy\,. \qquad \text{(D.7)}$$

The soft estimator for poverty gap, P_1, as derived in Section D.1, is defined to be,

$$\begin{aligned}
\mathrm{E}\left[P_1\left(Y\right)\right] &= \mathrm{P}\left[Y < \log(\mathrm{z})\right] - \frac{1}{\mathrm{z}} \int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^y\, f(y)\, dy \\
&= \mathrm{P}\left[Y < \log(\mathrm{z})\right] - \frac{\mathrm{e}^{\left(\mu + \frac{\sigma^2}{2}\right)}}{z}\, \mathrm{P}\left[Y^* < \log(\mathrm{z})\right]\,, \qquad \text{(D.8)}
\end{aligned}$$

where $Y^* \sim N(\mu + \sigma^2, \sigma^2)$. After comparison with Equation (D.8) we can rewrite Equation (D.7) as,

$$\mathrm{P}\left[Y < \log(\mathrm{z})\right] - \frac{2\,\mathrm{e}^{\left(\mu + \frac{\sigma^2}{2}\right)}}{z}\, \mathrm{P}\left[Y^* < \log(\mathrm{z})\right] + \frac{1}{\mathrm{z}^2} \int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^{2y}\, f(y)\, dy\,. \qquad \text{(D.9)}$$

Completing the square, similar to that used to develop the expression for the soft estimate for poverty gap, was applied to the third term in Equation (D.9). Firstly we substitute in

the expression for $f(y)$, the density function of Y,

$$\frac{1}{\mathrm{z}^2}\int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^{2y}\, f(y)\, dy \;=\; \frac{1}{\mathrm{z}^2}\int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^{2y}\;\cdot\;\frac{1}{\sqrt{2\pi\sigma^2}}\,\mathrm{e}^{-\frac{(y-\mu)^2}{2\,\sigma^2}}\, dy\,. \tag{D.10}$$

Simplification of the exponential term under the integral in Equation D.10 proceeds as follows:

$$\begin{aligned}\mathrm{e}^{2y}\,\mathrm{e}^{-\frac{(y\,-\,\mu)^2}{2\,\sigma^2}} \;&=\; \mathrm{e}^{-\frac{-4y\sigma^2\,+\,y^2\,-\,2\mu y\,+\,\mu^2}{2\sigma^2}}\\ &=\; \mathrm{e}^{-\frac{y^2\,-\,2y\left(\mu\,+\,2\sigma^2\right)\,+\,\mu^2}{2\sigma^2}}\,.\end{aligned}$$

Completing the square in the exponent gives;

$$\begin{aligned}\left[y\;-\;\left(\mu\;+\;2\sigma^2\right)\right]^2 \;&=\; y^2\;-\;2y\left(\mu\;+\;2\sigma^2\right)\;+\;\left(\mu\;+\;2\sigma^2\right)^2\\ &=\; y^2\;-\;2y\left(\mu\;+\;2\sigma^2\right)\;+\;\mu^2\;+\;4\mu\sigma^2\;+\;4\sigma^4\,.\end{aligned}$$

This implies that,

$$\begin{aligned}y^2\;-\;2y\left(\mu\;+\;2\sigma^2\right)\;+\;\mu^2 \;&=\; \left[y\;-\;\left(\mu\;+\;2\sigma^2\right)\right]^2\;-\;\left(4\mu\sigma^2\;+\;4\sigma^4\right)\\ &=\; \left[y\;-\;\left(\mu\;+\;2\sigma^2\right)\right]^2\;-\;4\sigma^2\left(\mu\;+\;\sigma^2\right)\,.\end{aligned}$$

Thus the third term of the soft estimate for poverty severity is expressed as,

$$\frac{1}{\mathrm{z}^2}\int_{-\infty}^{\log(\mathrm{z})} \mathrm{e}^{2y}\, f(y)\, dy \;=\; \frac{1}{\mathrm{z}^2}\int_{-\infty}^{\log(\mathrm{z})} \frac{1}{\sqrt{2\pi\sigma^2}}\,\mathrm{e}^{-\frac{\left[y-\left(\mu+2\sigma^2\right)\right]^2}{2\,\sigma^2}}\cdot\,\mathrm{e}^{\frac{4\sigma^2\left(\mu\,+\,\sigma^2\right)}{2\sigma^2}}\, dy$$

$$=\; \frac{\mathrm{e}^{\,2\left(\mu\,+\,\sigma^2\right)}}{\mathrm{z}^2}\int_{-\infty}^{\log(\mathrm{z})} \frac{1}{\sqrt{2\pi\sigma^2}}\,\mathrm{e}^{-\frac{\left[y-\left(\mu+2\sigma^2\right)\right]^2}{2\,\sigma^2}}\, dy$$

$$=\; \frac{\mathrm{e}^{\,2\left(\mu\,+\,\sigma^2\right)}}{\mathrm{z}^2}\quad \mathrm{P}\left[Y^{**} < \log(\mathrm{z})\right]\,,$$

where $Y^* \sim N(\mu + 2\sigma^2, \sigma^2)$. Thus the soft estimate of poverty severity, the expectation of the function P_2, can be expressed as a linear combination of three probabilities;

$$\mathrm{E}\left[P_2\left(Y\right)\right] \;=\; \mathrm{P}\left[Y < \log(\mathrm{z})\right] \;-\; \frac{2\,\mathrm{e}^{\left(\mu+\frac{\sigma^2}{2}\right)}}{z}\,\mathrm{P}\left[Y^* < \log(\mathrm{z})\right] \;+\; \frac{\mathrm{e}^{2(\mu\,+\,\sigma^2)}}{\mathrm{z}^2}\,\mathrm{P}\left[Y^{**} < \log(\mathrm{z})\right]\,.$$