MICROCOMPUTERS IN POLICY RESEARCH

3

# CLASSIFICATION AND REGRESSION TREES, CART™

## A USER MANUAL FOR IDENTIFYING INDICATORS OF VULNERABILITY TO FAMINE AND CHRONIC FOOD INSECURITY

YISEHAC YOHANNES
PATRICK WEBB

INTERNATIONAL FOOD POLICY RESEARCH INSTITUTE

The International Food Policy Research Institute was established in 1975 to identify and analyze alternative national and international strategies and policies for meeting food needs of the developing world on a sustainable basis, with particular emphasis on low-income countries and on the poorer groups in those countries. While the research effort is geared to the precise objective of contributing to the reduction of hunger and malnutrition, the factors involved are many and wide-ranging, requiring analysis of underlying processes and extending beyond a narrowly defined food sector. The Institute's research program reflects worldwide collaboration with governments and private and public institutions interested in increasing food production and improving the equity of its distribution. Research results are disseminated to policymakers, opinion formers, administrators, policy analysts, researchers, and others concerned with national and international food and agricultural policy.

IFPRI is a research center of the Consultative Group on International Agricultural Research and receives support from the Arab Fund for Economic and Social Development, Argentina, the Asian Development Bank, Australia, Belgium, Canada, China, Denmark, the European Commission, the Ford Foundation, France, the German Agency for Technical Cooperation (GTZ), the German Federal Ministry for Economic Cooperation and Development (BMZ), India, the Inter-American Development Bank, the International Development Research Centre (Canada), the International Fund for Agricultural Development, Ireland, Japan, the Land and Agriculture Policy Centre (South Africa), Mozambique, the Netherlands, Norway, the Philippines, the Rockefeller Foundation, the Rural Industries Research and Development Corporation (Australia), South Africa, the Southern African Development Bank, Spain, Sweden, Switzerland, the United Kingdom, the United Nations Children's Fund, the United States, and the World Bank.

# CLASSIFICATION AND REGRESSION TREES, CART™

## A USER MANUAL FOR IDENTIFYING INDICATORS OF VULNERABILITY TO FAMINE AND CHRONIC FOOD INSECURITY

YISEHAC YOHANNES
PATRICK WEBB

**CART is a registered trademark of California Statistical Software, Inc.**

MICROCOMPUTERS IN POLICY RESEARCH 3

INTERNATIONAL FOOD POLICY RESEARCH INSTITUTE

International Food Policy Research Institute
2033 K Street, N.W.
Washington, D.C. 20006-1002
U.S.A.

Library of Congress Cataloging-in-Publication Data available

Yohannes, Yisehac
Classification and Regression Trees, Cart™: A User Manual for Identifying Indicators of Vulnerability to Famine and Chronic Food Insecurity / Yisehac Yohannes, Patrick Webb.
p. cm.
Includes bibliographical references (p. ).
ISBN 0-89629-337-8

# CONTENTS

# TABLES

# ILLUSTRATIONS

# PREFACE

Over the past decade, the increasing power and reliability of microcomputers and the development of sophisticated software designed specifically for use with them has led to significant changes in the way that socioeconomic data are collected and analyzed. The venue of the computations has shifted from off-site mainframes, dependent on highly trained operators and significant capital investment in supporting equipment, to desktop and laptop computers, dependent only on the occasional availability of electricity. This means that it is now feasible to quickly transfer new statistical techniques between IFPRI and IFPRI's collaborators in developing countries, that data manipulation costs of policy analysis have been substantially reduced, and that a new level of complexity and accuracy is now possible in the collection and analysis of survey data in developing countries.

As with any new technology, however, there are substantial costs in time and money involved in learning the most efficient ways of using this new technology and then transmitting these lessons to others. This series, Microcomputers in Policy Research, represents IFPRI's collective ongoing experience in adapting microcomputer technology for use in food policy analysis in developing countries. The papers in the series are primarily for the purpose of sharing these lessons with potential users in developing countries, although persons and institutions in developed countries may also find them useful. The series is designed to provide hands-on methods for resolving statistical and data-collection problems encountered in food policy research. In our opinion, examples provide the best and clearest form of instruction; therefore, examples—including actual software codes wherever relevant—are used extensively throughout this series.

This third book in the series, *Classification and Regression Trees, CART™: A User Manual for Identifying Indicators of Vulnerability to Famine and Chronic Food Insecurity,* by Yisehac Yohannes and Patrick Webb, is a manual outlining how to use CART software to conduct classification- and regression-tree analysis. The manual is based on IFPRI's experiences from its famine research program, which was completed in 1998 with the publication of the book by Joachim von Braun, Tesfaye Teklu, and Patrick Webb, *Famine in Africa: Causes, Responses, and Prevention.* The manual shows how to use CART to identify indicators of a number of outcomes, including food insecurity and vulnerability to famine. Examples are provided throughout using programs from CART.™

Howarth Bouis, Lawrence Haddad,
and Stephen Vosti, Editors

# ACKNOWLEDGMENTS

We express our deepest appreciation to Lawrence Haddad, who not only initiated this task, but also provided continuous guidance and support throughout the writing of this manual. We are extremely grateful to Professor Leo Breiman of the University of California at Berkeley for his constructive critical review of the earlier draft and the suggestions he made. We also thank Professor Richard Olshen of Stanford University for his useful remarks and suggestions. Many thanks to Jay Willis for the task of editing and formatting the manual and to Uday Mohan for the editorial work he did. Of course, we are solely responsible for any errors incurred in the manual.

Yisehac Yohannes
International Food Policy Research Institute

Patrick Webb
Tufts University

# 1 INTRODUCTION

The Classification and Regression Tree (CART) approach to classifying statistical data has been used in many fields. One of its first uses involved the identification of ship structures from their radar-range profiles. Hospitals have used it to identify indicators of heart failure. CART also has been used in finance by Frydman, Altman, and Kao (1985) to classify distressed firms, and by Marais, Patell, and Wolfson (1985) to classify commercial loans. The International Food Policy Research Institute (IFPRI) has used CART to identify indicators of vulnerability to famine at regional and household levels (Webb et al. 1994). This manual is a guide for setting up CART-based information systems to identify indicators of vulnerability to famine, chronic food insecurity, and other failures of entitlement.

CART is a nonparametric technique that can select those variables and their interactions that are most important in determining an outcome or dependent variable. If an outcome variable is continuous, CART produces regression trees; if the variable is categorical, CART produces classification trees. The outcome variable used in this manual to approximate vulnerability to famine is the population in need of food over time as estimated by the Ethiopian Relief and Rehabilitation Commission (the commission has been renamed recently). CART produces regression trees with this variable.

This manual is intended to introduce the reader to the basic principles of CART methodology. It provides examples of CART analysis from IFPRI's earlier famine vulnerability studies (Webb et al. 1994; Seyoum et al. 1995) and discusses when and why CART might be useful for data analysis. In addition, the manual provides examples of computer program codes and discusses CART hardware and software requirements.

Earlier IFPRI research on famine in Africa (see Webb, von Braun, and Yohannes 1992; Teklu, von Braun, and Zaki 1991) had concluded that the basis for geographical and socioeconomic targeting of relief and rehabilitation interventions had to be improved. This required a refinement of existing methodologies for selecting and weighting indicators of household distress. The most important challenges entailed addressing existing methodological shortcomings in defining data needs, weighting the relative value of individual variables, and defining the significance of alternate (proxy) variables (Borton and Shoham 1991; Hutchinson et al. 1992; Riely 1993).

Vulnerability to food insecurity and famine cannot be measured by single, discrete variables. Given the close relationship between resources, food production, prices, and consumption, it is crucial that assessments of vulnerability examine the multiple factors that contribute not only to food supply shortfalls, but also to the performance of markets during crises, the failure of purchasing power among the poor, and the decline in consumption and nutritional status. This approach calls for the inclusion of other variables into the analysis, such as household asset base, isolation from major markets, level of farm technology, constraints to human capital, income levels and fluctuation, health and sanitation environments, and availability of nonfarm employment (Downing 1990; Webb, von Braun, and Yohannes 1992).

However, it is no easy matter to determine which additional variables have a role to play in the analysis (either singly or in combination), or what the relative predictive value is of constituent factors. These two problems have provided the starting point for numerous recent endeavors aimed at identifying the best predictive indicators for early warning and targeting purposes.

Progress *has* been made in targeting in recent years. There is now substantial agreement that indicators should reflect the behavior and livelihood conditions of target populations—those that are most often, and more severely, affected by acute food insecurity (Borton and Shoham 1991; FEWS 1993; FAO 1998). Such groups include the rural poor, women-headed households, asset-poor pastoralists, recently resettled households, households constrained by a high dependency ratio, the landless, the urban poor and unemployed, and retrenched civil servants. Not all of the groups are equally or always affected, but each is affected at a higher rate than more favored households inhabiting the same environment. Similarly, there is greater consensus today on the need for a core of indicators in addition to supply-side indexes in order to achieve more complete vulnerability assessments.

It remains unclear, however, which variables are most important, and what the minimum number of variables should be for the purpose of a valid analysis. For example, Borton and Shoham (1991) suggest 20 core indicators; Cutler (1986), Frankenberger (1992), and Seaman, Holt, and Allen (1993) each take between 20 and 30 indicators as the starting point; Riely (1993) and Downing (1993) both suggest more than 50 variables; while Currey (1978), one of the earliest practitioners in the field, started with 60 variables for his analysis of vulnerability in Bangladesh.

Some intervention programs are also constrained by data limitations that are at the root of many highly questionable assumptions (Maxwell 1989). Such limitations can lead to creative (sometimes very successful) approaches to comparing and combining data that are, in fact, incomparable and incompatible (Downing 1990; Hutchinson et al. 1992). Effective vulnerability analysis and early warning need to go beyond the cataloging and arbitrary indexing of available data.

In the past, the problems of determining the relative significance of indicators and sorting out statistical collinearity (data that may influence each other and not act as "independent" indicators of vulnerability) have been dealt with either through "delphic" techniques or through best-guess estimations based on a perceived "convergence of evidence" (Downing 1990; Borton and Shoham 1991). As a result, many analysts have called for refinements in methodology (Downing 1990; Borton and Shoham 1991; Hutchinson et al. 1992). So far, only two other methods have been explored. The first is to work closely with local experts who can help define indicators of local significance, thereby improving the reliability of subjective data manipulations (Currey 1978; Borton and Shoham 1991). The second is to analyze predicted characteristics of vulnerability (key variables and combinations of variables) against observed benchmark measures of crises, such as numbers of people actually affected by famine or receiving external assistance by region (Hutchinson et al. 1992).

IFPRI's research on vulnerability contributes to the latter approach. The analysis seeks to understand which indicators of vulnerability best explain reported numbers of "people in need" in Ethiopia across geographic regions and years. It thereby tries to establish a statistical basis for understanding the relative importance of various indicators (with nonsubjective weighting), while substantially reducing the problem of collinearity.

IFPRI used CART methodology to identify possible indicators of vulnerability in the form of classification and regression trees and thus overcame the problem of arbitrarily selecting indicators. IFPRI conducted a CART analysis of famine at two levels: at the household level and at the regional level. The household-level analysis used data from the 1989/90 IFPRI survey of households in famine-affected areas of Ethiopia. The regional analysis used regional-level, time-series data sets that were acquired and compiled by IFPRI. Breiman et al. (1984) suggest that CART methodology should "not be used to the exclusion of other methods," even though empirical evidence shows that CART performs much better than other procedures, such as discriminant analysis. The results generated by CART should be compared with results obtained by applying other methods to the same data set.

The main reference material on CART is the book *Classification and Regression Trees* by Breiman et al. (1984). It is the only book that offers both theory and methodology of CART and illustrates a number of examples in many disciplines.

This manual is organized as follows. Chapter 2 provides an overview of CART, including a detailed example of a classification tree. CART application areas and the strengths and weaknesses of CART are also discussed. Chapter 3 reviews CART methodologies for classification trees and provides a detailed description of the tree-building process. Chapter 4 provides a brief discussion of regression-tree methodology and an example from a regional vulnerability study. Chapter 5 describes software requirements as well as the basic programming codes needed to run CART software. Chapter 6 discusses refinements to the CART analyses presented earlier. The final chapter assesses the gains achieved by using CART and suggests what remains to be done. Selected results from CART programs are in Appendixes 1 and 2 and complete output examples are provided on diskette.

# 2 OVERVIEW OF CART

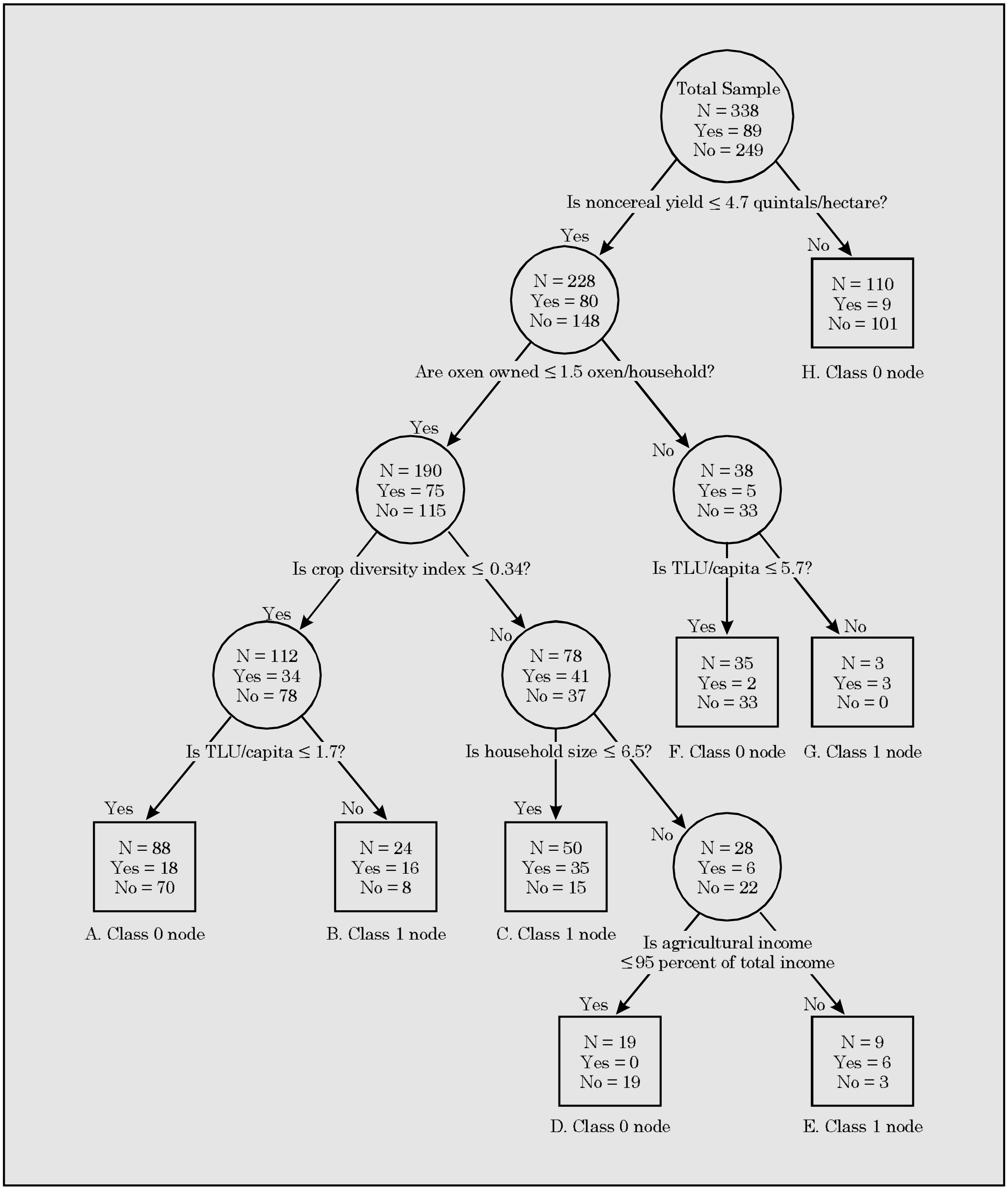
CART is a nonparametric statistical methodology developed for analyzing classification issues either from categorical or continuous dependent variables. If the dependent variable is categorical, CART produces a classification tree. When the dependent variable is continuous, it produces a regression tree. Detailed discussion of a regression tree is provided in Chapter 4. In both classification and regression trees, CART's major goal is to produce an accurate set of data classifiers by uncovering the predictive structure of the problem under consideration (Breiman et al. 1984). That is, CART helps in understanding the variables or interaction of variables that are responsible for a given phenomenon, such as famine, and that best determine one outcome rather than another (Seyoum et al. 1995). The purpose of such classifiers or classification rules is to enable one to predict the class (vulnerable or not vulnerable, in the case of famine households) of any future observation(s) from the profile of characteristics submitted for analysis. That is, given the characteristics of an observation, the goal is to find out whether the observation falls into the vulnerable class or not. The example in Figure 1 illustrates how CART methodology works.

In brief, the construction of a CART classification rule centers on the definition of three major elements discussed in Chapter 3. These are (1) the sample-splitting rule, (2) the goodness-of-split criteria, and (3) the criteria for choosing an optimal or final tree for analysis. CART builds trees by applying predefined splitting rules and goodness-of-split criteria at every step in the node-splitting process. In a highly condensed form, the steps in the tree-building process involve (1) growing a large tree (a tree with a large number of nodes), (2) combining some of the branches of this large tree to generate a series of subtrees of different sizes (varying numbers of nodes), and (3) selecting an optimal tree via the application of "measures of accuracy of the tree." These will be described in full in Chapter 3.

In Figure 1, the results of a CART analysis based on research on the vulnerability to famine (Webb et al. 1994) is summarized graphically in the form of an inverted tree. The CART analysis has two major objectives: (1) to get a better understanding of the characteristics of households that were vulnerable to famine, and (2) to generate tree-structured classifiers or indicators of vulnerability and assess the potential of these indicators for accurately predicting the prevalence of vulnerability to famine in the future.

The analysis is based on a sample survey of 338 households that was conducted in 1989/90 in Ethiopia. The list of variables used in the analysis is given in Table 1. The dependent variable is *CUTDUM*2. It is an indicator of vulnerability defined as a 0/1 binary variable. A household is vulnerable to famine if *CUTDUM*2=1 and not vulnerable if it

**Figure 1—Classification tree of a famine vulnerability study**



Notes: N stands for number of households at each node. TLU is tropical livestock unit, which converts big and small animals into a common unit.

**Table 1—Household variables**

| Name | Definition |
|---|---|
| *PCAST*80 | Per capita value of household assets (farm and nonfarm) |
| *PCNFRAST* | Per capita value of nonfarm assets (excluding livestock) |
| *PCLIVINC* | Household income per capita from livestock and products |
| *PCFRMAST* | Value of farm assets per capita |
| *PCINC* | Household income per capita |
| *PCAGINC* | Household income from crops and livestock per capita |
| *PCLSU*80 | Tropical livestock units owned per capita |
| *PCFRMINC* | Crop income per capita |
| *PCNNFINC* | Nonfarm income per capita |
| *LVSLSU*80 | Total tropical livestock units owned per household |
| *FRMASRAT* | Value of farm assets in total value of assets held |
| *NFRMASRA* | Value of nonfarm assets in total value of assets held |
| *CERLAR*80 | Cereal area cultivated (hectares) |
| *CERYLD*80 | Cereal yields (wheat equivalents in kilograms per hectare) |
| *HHEADSEX* | Gender of household head |
| *GINI* | Index of crop diversity (larger number = lower diversity) |
| *OXQ*80 | Number of oxen owned per household |
| *NCERYL*80 | Noncereal yields (wheat equivalents in kilograms per hectare) |
| *NCERAR*80 | Noncereal area cultivated (hectares) |
| *AGINCRAT* | Share of crop and livestock income in total income |
| *LIVSYRAT* | Share of income from livestock and livestock products in total income |
| *FARMYRAT* | Share of crop income in total income |
| *NFRMYRAT* | Share of nonfarm income in total income |
| *PCDCALS* | Calorie consumption per day per capita |
| *HHSIZE* | Household size |
| *CUTDUM*2 | Dummy variable (1 = vulnerable household; 0 = not vulnerable) |
| *CALDUM* | Per capita daily calorie consumption group |

Source: International Food Policy Research Institute/Office of National Committee for Central Planning (Ethiopia)/International Livestock Center for Africa (now the International Livestock Research Institute) survey, 1989/90, reported in Webb, von Braun, and Yohannes 1992.

equals 0. These two categories of vulnerability are referred to as class 1 and class 0, respectively. During the Ethiopian famine in the 1980s, 89 of the sample households were classified as vulnerable to famine, while 249 were not. The top circle in Figure 1 contains this basic information (N=338, yes=89, and no=249).

Without going into technical details of the tree-building process (see Chapter 3), it should simply be noted here that CART splits a sample into binary subsamples based on the response to a very simple question requiring only a yes/no answer. The question used to create splits is given at the bottom of each circle (Figure 1). Each question is based only on a single variable chosen from the list of variables in Table 1. Depending on the response (yes/no) to the question, the sample is partitioned into left and right binary subsamples. The issue of how CART chooses a variable and its split point is discussed in Chapter 3. When a split occurs, the subsamples, also called nodes, end up either in a circle or in a rectangular box. The rectangular boxes are referred to as terminal nodes and the circles are nonterminal nodes. Terminal nodes do not split further, while nonterminal nodes do. From here on, node will be used instead of subsample.

The noncereal yield variable produces the first split in the sample (Figure 1). Noncereals are composed mostly of pulses and are given in terms of wheat equivalents. Noncereals, especially pulses, constitute a major component in the diet of the poor in Ethiopia. The average noncereal yield across the sample is 247 quintals per hectare. The cutoff point is 4.7 quintals per hectare. Households with low noncereal yield go to the left node and the remaining to the right node. The right node is in a rectangle and cannot be split any further. Underneath this node are the labels "H" and "class 0 node." These labels identify, respectively, the node and the class to which the node is assigned. This terminal node is classified as class 0 because it contains nonvulnerable households. The left node is nonterminal and is subject to a further split.

The second split is based on whether a household owns less than two oxen. Because farmers can cultivate only with a pair of oxen, households with one ox or none go to the left node and the remaining to the right node. For households with no more than one ox, the next split Is based on a crop diversity index. This index measures the mix of crops planted by a household. The higher the diversity index, the more mixed or diversified are the planted crops. Households with a crop diversity index of less than or equal to 0.34 are sent to the left node while those with a higher diversity index are sent to the right node.

Continuing with the split, households with a crop diversity index of at most 0.34 are further split based on the tropical livestock unit (*TLU*) per capita variable. *TLU* is an index that converts big and small animals into a common unit. Households with *TLU* less than or equal to 1.7 per capita are sent to the left terminal node while the others go to the right terminal node. The two terminal nodes are labeled A and B. Terminal node A is classified as class 0 (nonvulnerable households), while terminal node B is classified as class 1 (vulnerable households). The other terminal nodes, labeled C through G, are generated in a similar manner.

Each terminal node is the endpoint of a separate path or structure, and yet a group of them end up in the same class. This indicates that paths to vulnerability or nonvulnerability to famine depend on the amount of resources with which households are endowed. Households in terminal nodes A, D, F, and H, are classified as nonvulnerable to famine, while households in terminal nodes B, C, E, and G are classified as vulnerable.

The sequential structure leading to terminal node B indicates that this set of vulnerable households has extremely low noncereal yield per hectare, one ox or none, low crop diversity, and high *TLU* per capita. These are typically extremely poor households whose livelihoods appear to depend mostly on livestock holdings. Indeed, examination of the data set shows that 87.5 percent of the vulnerable households at this terminal node come from a survey site where 70 percent of the households reported reduction in the number of meals consumed during the Ethiopian famine of the 1980s. Furthermore, it is a pastoral site (Beke Pond) located in a lowland area where livestock rather than farming sustain well-being. Most of the characteristics of households in this terminal are captured by the four variables used to arrive at the node.

Households in terminal node C are identified by extremely low noncereal yield per hectare ownership of one ox or none, at least average crop diversity, and a household size of at most 6.5. Examination of the data set shows that 71 percent of the vulnerable households at this terminal node come from the Dinki area, which was the survey site most affected by the famine of the 1980s (Webb et al. 1992). Nearly 71 percent of the households at this survey site reported reducing the number of meals consumed during the famine. Clearly, the four variables that lead to this terminal node along with their cutoff points form the best indicators of vulnerability to famine for households at this location.

Terminal node E characterizes vulnerable households as those with extremely low noncereal yield per hectare, less than two oxen, at least average crop diversity, large household size, and almost all income derived from agriculture. Fifty percent of the vulnerable households at this terminal node come from the Dinki survey site.

Terminal node G is a pure node. It contains only households that are vulnerable to famine. These are households with extremely low noncereal yield per hectare, at least two oxen, and a large per capita livestock holding. The vulnerable households at this terminal node come from Beke Pond (a pastoral site).

The most interesting aspect of this exercise is that the CART procedure identified the characteristics of households most affected by the famine of the 1980s by using only 6 of the 27 variables. These 6 variables along with their cutoff points carry most of the information required for establishing tree-structured classification rules that could identify vulnerable households in the future. Vulnerable households at Dinki and Beke Pond account for 67 percent of all vulnerable households in the 7 survey sites. CART has successfully untangled the complexities of a data set and identified the indicators of households vulnerable to famine.

## HIGHLIGHTS OF OTHER CLASSIFICATION METHODS AND PROCEDURES

Besides CART, a number of other methods and procedures for classifying data exist. These methods fall into two groups.

| Group 1 | Group 2 |
|---|---|
| AID | Discriminant analysis |
| THAID | Kernel density estimation |
| CHAID | Kth nearest neighbor |
| | Logistic regression |
| | Probit models |

The methods in Group 1 generate classification trees. AID is an acronym for Automatic Interaction Detection. It is a classification algorithm developed by J. N. Morgan and J. A. Sonquist in 1963 at the University of Michigan. The AID algorithm led to the development of

THAID (a sequential search program for analysis of nominal scale dependent variables) by Morgan and Messenger at the University of Michigan in 1973, and Chi-squared Automatic Interaction Detection (CHAID) by Kass in 1980. These three procedures generate multilevel splits in producing classification trees. Unlike CART, they are not distribution-free and they all employ significance tests on predictor variables to generate node splits and determine the size of a tree. These two methods differ from CART in the process of tree growing and pruning and estimation of predictive error results.

The methods in Group 2 do not produce classification trees. They all assume functional relationships between dependent and predictor variables. Discriminant analysis, Kernel density estimation, and K*th* nearest neighbor are the most widely used classification methods. Breiman et al. (1984, 15–17) provide details on these methods and their weaknesses. Since discriminant analysis or its variation, linear discriminant function, has been widely used as a classification method, especially in education and in psychology, business, and marketing research (for example, in product targeting and market segmentation), a brief review of the methodology follows.

In order to use the linear discriminant function method, the following distributional assumptions must hold (Maddala 1983):

1. All of the predictor variables should follow multivariate normal distribution for each class of dependent variable, and
2. The variance-covariance matrixes of each class should be equal.

The procedure first forms a linear combination of predictor variables and then the coefficients of the variables in the linear combination are estimated. This is followed by computation of a discriminant score for each case or observation using the estimated coefficients and the corresponding values of the predictor variables. A classification rule is formed by applying Baye's Rule to the discriminant scores.

The distributional assumption of normality is strong and the methodology is used regardless of whether the assumptions hold for every variable used in the analysis. The method is designed to handle only continuous predictor variables. Categorical predictor variables need to be transformed into a series of dummy variables. This additional task leads to the problem of dimensionality (having too many variables). Furthermore, all variables that enter into linear combination have to be complete. That is, no case with missing values for a variable can be used in the analysis. Observations with missing values for a variable have to be dropped. This may result in bias due to reduced sample size. Also, the procedure is known to yield poor results if the predictor variables are all binary or a mixture of continuous and binary.

Logistic regression and probit models are other parametric methods used in classification studies. The final outcome of these methods yields the proportion of predicted cases that falls into different categories of the dependent variable. As in linear discriminant analysis, these methods are not distribution-free, do not have any provision for analyzing cases with missing values for a variable, and deal only with categorical dependent variables. As in all parametric models, the variables used in the analysis are entirely determined by the analyst.

CART methodology further develops and enhances the work on classification methodology of AID and THAID (Breiman et al. 1984). But CART overcomes the problems associated with these algorithms and some of the drawbacks associated with the classification methods in Group 2.

Breiman et al. (1984) made several comparative analyses of CART and discriminant analysis results and found that CART performed better than discriminant analysis. Marais, Patell, and Wolfson (1985) also noted similar findings in their classification study of commercial loans, as did Srinivasan and Kim (1987) in their credit-granting study. But in models where linear structure and the assumption of normality hold, Breiman et al. (1984) found that results from discriminant analysis were better than those from CART. Regardless of the problems with other procedures, Breiman et al. (1984) advise not to use CART "to the exclusion of other methods." Whenever possible, one of the other methods should be used for comparative purposes.

## SUMMING UP CART'S STRENGTHS AND WEAKNESSES

Breiman et al. (1984) and Steinberg and Colla (1995) provide a number of justifications for using CART. A few of them are listed below.

1. CART makes no distributional assumptions of any kind for dependent and independent variables. No variable in CART is assumed to follow any kind of statistical distribution.
2. The explanatory variables in CART can be a mixture of categorical and continuous.
3. CART has a built-in algorithm to deal with the missing values of a variable for a case, except when a linear combination of variables is used as a splitting rule (see Chapter 3).
4. CART is not at all affected by the outliers, collinearities, heteroskedasticity, or distributional error structures that affect parametric procedures. Outliers are isolated into a node and thus have no effect on splitting. Contrary to situations in parametric modeling, CART makes use of collinear variables in "surrogate" splits.
5. CART has the ability to detect and reveal variable interactions in the data set.
6. CART does not vary under a monotone transformation of independent variables; that is, the transformation of explanatory variables to logarithms or squares or square roots has no effect on the tree produced.
7. In the absence of a theory that could guide a researcher, in a famine vulnerability study, for example, CART can be viewed as an exploratory, analytical tool. The results can reveal many important clues about the underlying structure of famine vulnerability.
8. CART's major advantage is that it deals effectively with large data sets and the issues of higher dimensionality; that is, it can produce useful results from a large number of variables submitted for analysis by using only a few important variables.
9. The inverted-tree-structure results generated from CART analysis are easy for anyone to understand in any discipline.

CART analysis does have some limitations, however.[1] CART is a blunt instrument compared to many other statistical and analytical techniques. At each stage, the subdivision of data into two groups is based on only one value of only one of the potential explanatory variables. If a statistical model that appears to fit the data exists, and if its basic assumptions appear to be satisfied, that model would be preferable, in general, to a CART tree.

A weakness of the CART method and, hence, of the conclusions it may yield is that it is not based on a probabilistic model. There is no probability level or confidence interval associated with predictions derived from a CART tree that could help classify a new set of data. The confidence that an analyst can have in the accuracy of the results produced by a given CART tree is based purely on that tree's historical accuracy—how well it has predicted the desired response in other, similar circumstances. This is essentially how the structure of the tree is determined in the first place, through k-fold cross-validation, which will be discussed in Chapter 3.

[1] The following is adapted from Seyoum et al. 1995.

# 3 BASIC PRINCIPLES OF CART METHODOLOGY

## THE PREDICTIVE ACCURACY OF CART

Accuracy is the most important feature of a classification tree. All classification procedures, however, including CART, can produce errors. The CART procedure does not make any distributional assumptions on covariates; hence, hypothesis testing is not possible. Confidence in CART's performance, therefore, has to be based primarily on an assessment of the extent of misclassification it generates from data sets with known class distributions and on knowledge of and experience with the subject matter under study.

The best way to test the predictive accuracy of a tree is to take an independent test data set with known class distributions and run it down the tree and determine the proportion of cases missclassified. In empirical studies, the possibility of getting such a data set is remote. To overcome this difficulty, Breiman et al. (1984) provide three procedures for estimating the accuracy of tree-structured classifiers.

First, let

$c(X)$ or $c$ = a tree-structured classifier, where $X$ is a vector of characteristics variables that describe an observation;
$R^*[c(X)]$ = the classifier's "true" misclassification rate; and
$L$ = the learning sample (the sample data from which to construct a classification tree).

The three estimation procedures below have two objectives: constructing a classification tree, $c(X)$, and then finding an estimate of $R^*[c(X)]$.

1. *Resubstitution Estimate*, $R[c(X)]$. This estimates the accuracy of the true misclassification rate, $R^*[c(X)]$, as follows:

   1a. Build a classification tree, $c(X)$, from the learning sample $L$, and save it.
   1b. Apply the tree, $c(X)$, to the data set from which it is built. That is, let the observations in the sample run down the tree one at a time.
   1c. Compute the proportion of cases that are misclassified. This proportion is the resubstitution estimate, $R[c(X)]$, of the true misclassification rate, $R^*[c(X)]$.

The resubstitution estimate tests the accuracy of a classifier by applying it to observations for which the classes are known. The major weakness of this estimator of the error rate is that it is derived from the same data set from which the tree is built; hence, it underestimates the true misclassification rate. The error rate is always low in such cases.

2. *Test-sample estimate.* If the sample is large,

   2a. Divide the observations in the learning sample, $L$, into two parts: $L_1$ and $L_2$. $L_1$ and $L_2$ need not be equal. For example, two-thirds of the cases in $L$ can be chosen randomly to constitute $L_1$, and the remaining one-third can constitute cases in $L_2$.
   2b. Use $L_1$ to build the classifier, $c(X)$, and save it.
   2c. Run observations in $L_2$ down the tree, $c(X)$, one at a time.
   2d. Compute the proportion of cases that are misclassified. This proportion is the test-sample estimate, $R[c(X)]$, of the "true" misclassification rate, $R^*[c(X)]$. In large samples, this estimate provides an unbiased estimate of the true misclassification rate.

3. *K-fold cross-validation.* This is the recommended procedure for small samples and it works as follows:

   3a. Divide the learning sample, $L$, into $K$ subsets of an equal number of observations. Let $L_1, L_2, ..., L_k$ be the subsamples.
   3b. Construct a classifier, $c(X)$, from the $k$–1 subsamples by leaving out, say, the $k$th subsample, $L_k$.
   3c. Save the resulting classifier, $c(X)$.
   3d. Apply the saved classifier, $c(X)$, to the excluded subsample, $L_k$, and estimate $R[c(X)]$ as the proportion of misclassified observations. Denote this estimate as $R^{ts}[c^k(X)]$, where $k$ denotes $k$-fold cross-validation, and $ts$ denotes test sample.
   3e. Repeat steps 3b, 3c, and 3d, using all subsamples except the subsample $L_{k-1}$. The subsample $L_{k-1}$ now becomes a test sample. The process above is repeated until every subsample is used once in the construction of $c(X)$ and once as a test sample. The result is a series of test sample resubstitution estimates,

$$R^{ts}[c^k(X)],\ R^{ts}[c^{k-1}(X)],..,\ R^{ts}[c^1(X)].$$

   3f. Add the series of $R^{ts}[c^k(X)]$, $R^{ts}[c^{k-1}(X)],...,$ $R^{ts}[c^1(X)]$ generated from the $k$-fold cross-validation and get an estimate of $R[c(X)]$; that is, the $k$-fold cross-validation estimate $R^{ck}(c)$ of $R[c(X)]$ is given as

$$R^{ck}(c) = 1/k\sum_{k=1} R^{ts}[c^{(k)}],$$

which is an average of the error rates from $k$ cross-validation tests. For example, if $k$ =10, then the average is over the 10 test samples. Tenfold cross-validation is recommended.

## METHODOLOGY FOR BUILDING A CLASSIFICATION TREE

In constructing a classification tree, CART makes use of prior probabilities (priors). A brief review of priors and their variations as used in CART is provided.

Prior probabilities play a crucial role in the tree-building process. Three types of priors are available in CART: *priors data*, *priors equal*, and *priors mixed*. They are either estimated from data or supplied by the analyst.

In the following discussion, let

$N$ = number of cases in the sample,
$N_j$ = number of class $j$ cases in the sample, and
$\pi_j$ = prior probabilities of class $j$ cases.

- *Priors data* assumes that distribution of the classes of the dependent variable in the population is the same as the proportion of the classes in the sample. It is estimated as $\pi_j = N_j / N$.
- *Priors equal* assumes that each class of the dependent variable is equally likely to occur in the population. For example, if the dependent variable in the sample has two classes, then prob(class 1) = prob(class 2) = 1/2.
- *Priors mixed* is an average of *priors equal* and *priors data* for any class at a node.

## COMPONENTS FOR BUILDING A CLASSIFICATION TREE

Three components are required in the construction of a classification tree: (1) a set of questions upon which to base a split; (2) splitting rules and goodness-of-split criteria for judging how good a split is; and (3) rules for assigning a class to each terminal node. Each of these components are discussed below.

### Type and Format of Questions

Two question formats are defined in CART: (1) Is $X \leq d$?, if $X$ is a continuous variable and $d$ is a constant within the range of $X$ values. For example, is income ≤ 2,000? Or (2) is $Z = b$?, if $Z$ is a categorical variable and $b$ is one of the integer values assumed by $Z$. For example, is *sex* = 1?

The number of possible split points on each variable is limited to the number of distinct values each variable assumes in the sample. For example, if a sample size equals $N$, and if $X$ is a continuous variable and assumes $N$ distinct points in the sample, then the maximum number of split points on $X$ is equal to $N$. If $Z$ is a categorical variable with $m$ distinct points in a sample, then the number of possible split points on $Z$ equals $2^{m-1} - 1$ (Breiman et al. 1984, 30). Unless otherwise specified, CART software assumes that each split will be based on only a single variable.

### *Splitting Rules and Goodness-of-Split Criteria*

This component requires definition of the impurity function and impurity measure. Let

$j = 1,2,...,k$ be the number of classes of categorical dependent variables;

then define $p(j \mid t)$ as class probability distribution of the dependent variable at node $t$, such that $p(1 \mid t) + p(2 \mid t) + p(3 \mid t) + ... + p(k \mid t) = 1$, $j = 1, 2, ... , k$. Let $i(t)$ be the impurity measure at node $t$. Then define $i(t)$ as a function of class probabilities $p(1 \mid t)$, $p(2 \mid t)$, $p(3 \mid t)$, .... Mathematically, $i(t) = \phi\ [p(1 \mid t), p(2 \mid t), \ldots, p(j \mid t)]$. The definition of impurity measure is generic and allows for flexibility of functional forms.

***Splitting Rules.*** There are three major splitting rules in CART: the Gini criterion, the twoing rule, and the linear combination splits. In addition to these main splitting rules, CART users can define a number of other rules for their own analytical needs. CART uses the Gini criterion (also known as Gini diversity index) as its default splitting rule. The twoing rule is discussed in detail in Breiman et al. 1984 and will not be covered here. A brief exposition of the linear combination splits is provided later in this chapter.

The Gini impurity measure at node $t$ is defined as $i(t) = 1 - S$, where $S$ (the impurity function) = $\sum p^2(j \mid t)$, for $j = 1,2,\ldots,k$ (Steinberg and Colla 1995; Breiman et al. 1984).

The impurity function attains its maximum if each class (vulnerable or not) in the population occurs with equal probability. That is, $p(1 \mid t) = p(2 \mid t) = \ldots = p(j \mid t)$. On the other hand, the impurity function attains its minimum (= 0) if all cases at a node belong to only one class. That is, if node $t$ is a pure node with a zero misclassification rate, then $i(t) = 0$.

***Goodness-of-Split Criteria.*** Let $s$ be a split at node $t$. Then, the goodness of split "$s$" is defined as the decrease in impurity measured by

$$\Delta i(s,t) = i(t) - p_L[i(t_L)] - p_R[i(t_R)].$$

where

$s$ = a particular split,
$p_L$ = the proportion of the cases at node $t$ that go into the left child node, $t_L$,
$p_R$ = the proportion of cases at node $t$ that go into the right child node, $t_R$,
$i(t_L)$ = impurity of the left child node, and
$i(t_R)$ = impurity of the right child node.

### *Class Assignment Rule*

There are two rules for assigning classes to nodes. Each rule is based on one of two types of misclassification costs.

1. *The Plurality Rule: Assign terminal node* t *to a class for which* p(j | t) *is the highest.* If the majority of the cases in a terminal node belong to a specific class, then that node is assigned to that class. The rule assumes *equal* misclassification costs for each class. It does not take into account the severity of the cost of making a mistake. This rule is a special case of rule 2.
2. *Assign terminal node* t *to a class for which the expected misclassification cost is at a minimum.* The application of this rule takes into account the severity of the costs of misclassifying cases or observations in a certain class, and incorporates cost variability into a Gini splitting rule.

When dealing with famine vulnerability, for example, misclassifying a vulnerable household as nonvulnerable has more severe consequences than misclassifying a nonvulnerable household as vulnerable. Variable costs can be accounted for by defining a matrix of variable misclassification costs that can be incorporated into the splitting rules.

Let $c(i \mid j)$ = the cost of classifying a class $j$ case as a class $i$ case:

$$c(i \mid j) \geq 0 \text{ if } i \neq j,\ c(i \mid j) = 0 \text{ if } i = j.$$

Now, assume that there are two classes in a problem. Let

$\pi_t(1)$ = prior probability of class 1 at node $t$,
$\pi_t(2)$ = prior probability of class 2 at node $t$,
$\mathrm{r}_1(t)$ = the cost of assigning node $t$ to class 1, and
$\mathrm{r}_2(t)$ = the cost of assigning node $t$ to class 2.

Given priors and variable misclassification costs, $r_1(t)$ and $r_2(t)$ are estimated as follows:

$$r_1(t) = \pi(1) \cdot c(2 \mid 1),$$

and

$$r_2(t) = \pi(2) \cdot c(1 \mid 2).$$

According to rule 2, if at node $t$, $r_1(t) < r_2(t)$, node $t$ is assigned to class 1. If $c(2 \mid 1) = c(1 \mid 2)$, then rule (1) applies and a node is assigned to a class for which the prior probability is the highest.

## STEPS IN BUILDING A CLASSIFICATION TREE

The tree-building process starts by partitioning a sample or the root node into binary nodes based upon a very simple question of the form

$$\text{is } X \leq d?,$$

where $X$ is a variable in the data set and $d$ is a real number. Initially, all observations are placed in the root node. This node is impure or heterogenous because it contains observations of mixed classes. The goal is to devise a rule that will break up these observations and create groups or binary nodes that are internally more homogenous than the root node. CART uses a computer-intensive algorithm that searches for the best split at all possible split points for each variable. The methodology that CART uses for growing trees is technically known as binary recursive partitioning (Steinberg and Colla 1995). Starting from the root node, and using, for example, the Gini diversity index as a splitting rule, the tree building process is as follows:

1. CART splits the first variable at all of its possible split points (at all of the values the variable assumes in the sample). At each possible split point of a variable, the sample splits into binary or two child nodes. Cases with a “yes” response to the question posed are sent to the left node and those with “no” responses are sent to the right node.

2. CART then applies its goodness-of-split criteria to each split point and evaluates the reduction in impurity that is achieved using the formula

$$\Delta i(s,t) = i(t) - p_L[i(t_L)] - p_R[i(t_R)],$$

which was described earlier.
3. CART selects the best split of the variable as that split for which the reduction in impurity is highest.
4. Steps 1–3 are repeated for each of the remaining variables at the root node.
5. CART then ranks all of the best splits on each variable according to the reduction in impurity achieved by each split.
6. It selects the variable and its split point that most reduced the impurity of the root or parent node.
7. CART then assigns classes to these nodes according to the rule that minimizes misclassification costs. CART has a built-in algorithm that takes into account user-defined variable misclassification costs during the splitting process. The default is unit or equal misclassification costs.
8. Because the CART procedure is recursive, steps 1–7 are repeatedly applied to each nonterminal child node at each successive stage.
9. CART continues the splitting process and builds a large tree. The largest tree is built if the splitting process continues until every observation constitutes a terminal node. Obviously, such a tree will have a large number of terminal nodes, which will be either pure or have very few cases (less than 10; Steinberg and Colla 1995).

### Linear Combination Splits

This splitting rule is an alternative to CART's use of a single variable for splitting. It is designed for situations where the class structure of the data appears to depend on linear combinations of variables. In linear combination splits, the question posed for a node split takes the following form:

$$\text{Is } \alpha_1 \cdot X_1 + \alpha_2 \cdot X_2 \leq 40\,?$$

For example,

$$\text{is } .55 \cdot \text{consumption} + .05 \cdot \text{age} \leq 40?$$

If the response to the question is "yes," then the case is sent to the left node, and if the response is "no," then the case is sent to the right node.

This rule is valid only for cases with no missing values on predictor variables. Furthermore, if categorical variables have to be included in the model, they should be converted to sets of dummy variables. If this option is chosen as a splitting method, it should be specified on the command line. The syntax for the command line is provided in Chapter 5.

### *Missing Values and Splitting Points*[2]

Incompleteness of data may be a problem for conventional statistical analysis, but not for CART. It makes use of a "surrogate" variable splitting rule. A surrogate variable in CART is that variable that mimics or predicts the split of the primary variable. If a splitting variable used for tree construction has missing values for some cases, those cases are not thrown out. Instead, CART classifies such cases on the basis of the best surrogate variable (the variable with a close resemblance to the primary split variable). The surrogate may have a different cutoff point from the primary split, but the number of cases the surrogate split sends into left and right nodes should be very close to that with the primary split. By default, CART analysis produces five surrogate variables as part of its standard output. Surrogate splits are available only for splits based on a single variable. They are not available if the linear combination splitting rule is selected.

Apart from handling the missing data points of a case, surrogate variables can also be used for detecting the masking of variables and determining the rank of variables important either in actual or potential tree construction. Appendix 1, Example 1 provides a list of surrogates produced with the Ethiopian data and a column of variable importance (the relative importance of variables).

### Outliers and Splitting Points

Outliers among the independent variables rarely affect CART analysis, because splits are generally determined at non-outlier values. If outliers exist in the dependent variable, they are isolated in small nodes, where they do not affect the rest of the tree (Webb et al. 1994).

## TREE PRUNING

Large trees can have two problems: (1) Although they are highly accurate, with low or zero misclassification rates, large trees provide poor results when applied to new data sets (Steinberg and Colla 1995). And (2) understanding and interpreting trees with a large number of terminal nodes is a complicated process. Hence, large trees are referred to as complex trees. The complexity of a tree is measured by the number of its terminal nodes.

Departures from the ideal situation of low or zero misclassification entails a trade-off between accuracy and tree complexity. The relationship between tree complexity and accuracy can be understood with the cost complexity measure, which is defined as

$$\text{Cost Complexity} = \text{Resubstitution Misclassification Cost} \\ + \beta \cdot \text{Number of terminal nodes},$$

where $\beta$ is penalty per additional terminal node. If $\beta = 0$, then cost complexity attains its minimum for the largest possible tree. On the other hand, as $\beta$ increases and is sufficiently large (say, infinity), a tree with one terminal node (the root node) will have the lowest cost complexity. As values of $\beta$ decrease and approach zero, trees that minimize cost

[2] This section and the following one on outliers and split points come from Seyoum et al. 1995.

complexity become larger. The “right-sized” tree with “correct” complexity should lie between these two extremes. Breiman et al. 1984 discuss how to estimate $\beta$ and offer a detailed account of the pruning process.

The search for the “right-sized” tree starts by pruning or collapsing some of the branches of the largest tree ($T_{max}$) from the bottom up, using the cost complexity parameter and cross-validation or an independent test sample to measure the predictive accuracy of the pruned tree. Hypothetical examples of the largest possible tree ($T_{max}$), the pruned branch, and the pruned tree are given in Figures 2, 3, and 4, respectively. These examples illustrate only one of the many possibilities in the tree-growing and tree-pruning process.

The pruning process produces a series of sequentially nested subtrees along with two types of misclassification costs and cost-complexity-parameter values. These are the cross-validated relative-error cost from applying tenfold cross-validation and the resubstitution relative cost generated from the learning sample. The trade-off between cost complexity and tree size can be seen in the last column of Table 2. Using the resubstitution cost, CART ranks the subtrees and generates a tree sequence table ordered from the most complex tree at the top to a less complex tree with one terminal node at the bottom (Table 2). It is a real example, taken from the computer output that produced Figure 1.

In other words, the tree-sequence table provides subtrees with a decreasing complexity (a decreasing number of terminal nodes) and an increasing cost (resubstitution relative cost). CART finally identifies the minimum-cost tree, and picks an optimal tree as the tree within one standard error of the minimum-cost tree. The option of a one-standard-error rule can be changed by the data analyst. But the reason for using a one-standard-error rule is that there may be other trees with cross-validated error rates close to those of the minimum-cost tree. Breiman et al. (1984) suggest that an optimal tree should be the one with the smallest terminal nodes among those that lie within one standard error of the minimum-cost tree. The minimum-cost tree itself could become the “right-sized” or the optimal-cost tree.

**Figure 2—An example of the largest CART tree ($T_{max}$)**

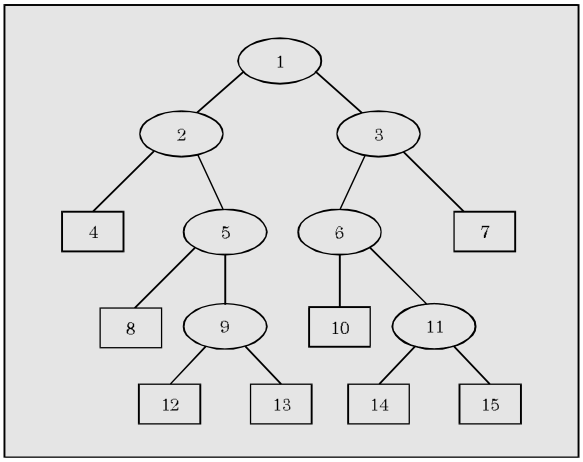


**Figure 3—Branch 3 of the largest CART tree**

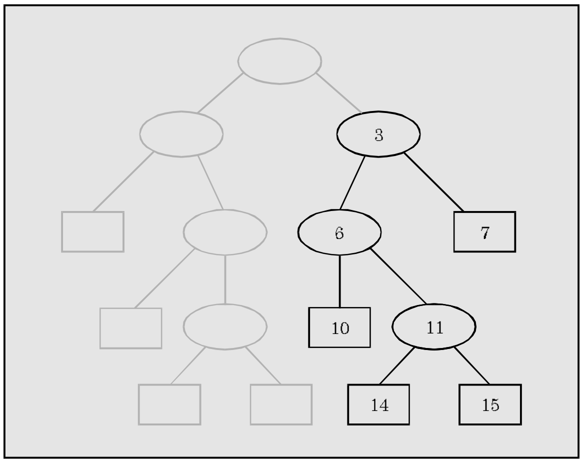
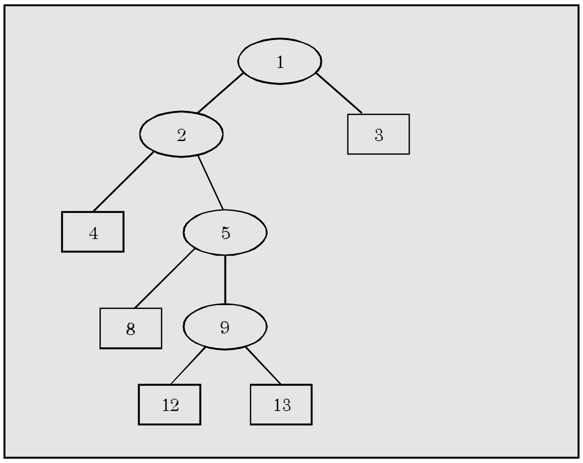


**Figure 4—Pruned tree**



### Table 2—Example of sequence of trees produced by pruning

**Dependent variable: *CUTDUM*2**

| Tree | Number of terminal nodes | Cross-validated relative cost | Resubstitution relative cost | Complexity parameter |
|---|---|---|---|---|
| 1 | 32 | 0.704 +/– 0.060 | 0.145 | 0.000 |
| 8 | 16 | 0.639 +/– 0.058 | 0.244 | 0.008 |
| 9 | 14 | 0.635 +/– 0.058 | 0.276 | 0.008 |
| 10 | 12 | 0.632 +/– 0.058 | 0.310 | 0.008 |
| 11* | 11 | 0.603 +/– 0.057 | 0.332 | 0.011 |
| 12** | 8 | 0.634 +/– 0.058 | 0.430 | 0.016 |
| 13 | 7 | 0.668 +/– 0.059 | 0.464 | 0.017 |
| 14 | 5 | 0.687 +/– 0.059 | 0.540 | 0.019 |
| 15 | 3 | 0.700 +/– 0.058 | 0.619 | 0.020 |
| 16 | 2 | 0.729 +/– 0.048 | 0.696 | 0.038 |
| 17 | 1 | 1.000 +/– 0.000 | 1.000 | 0.152 |

Initial misclassification cost = 0.500
Initial class assignment = 0

* indicates minimum-cost tree.
** indicates optimum-cost tree.

In Table 2, the cross-validated relative-cost column shows that cross-validation error initially decreases as complexity decreases, reaches a minimum, and then increases. CART picks the tree with the minimum cross-validated cost as the minimum-cost tree, which is marked by an asterisk. The minimal-cost tree has 11 terminal nodes and a cross-validated cost of 0.603 +/– 0.057. The optimal tree is obtained by applying the one-standard-error rule to the minimum-cost tree. Tree number 12 with 8 terminal nodes meets the criteria of an optimal-cost tree and it is identified by two asterisks. Tree number 10 with 12 terminal nodes is another candidate for an optimal tree. However, it is more complex than tree number 12.

# 4 REGRESSION TREES: AN OVERVIEW

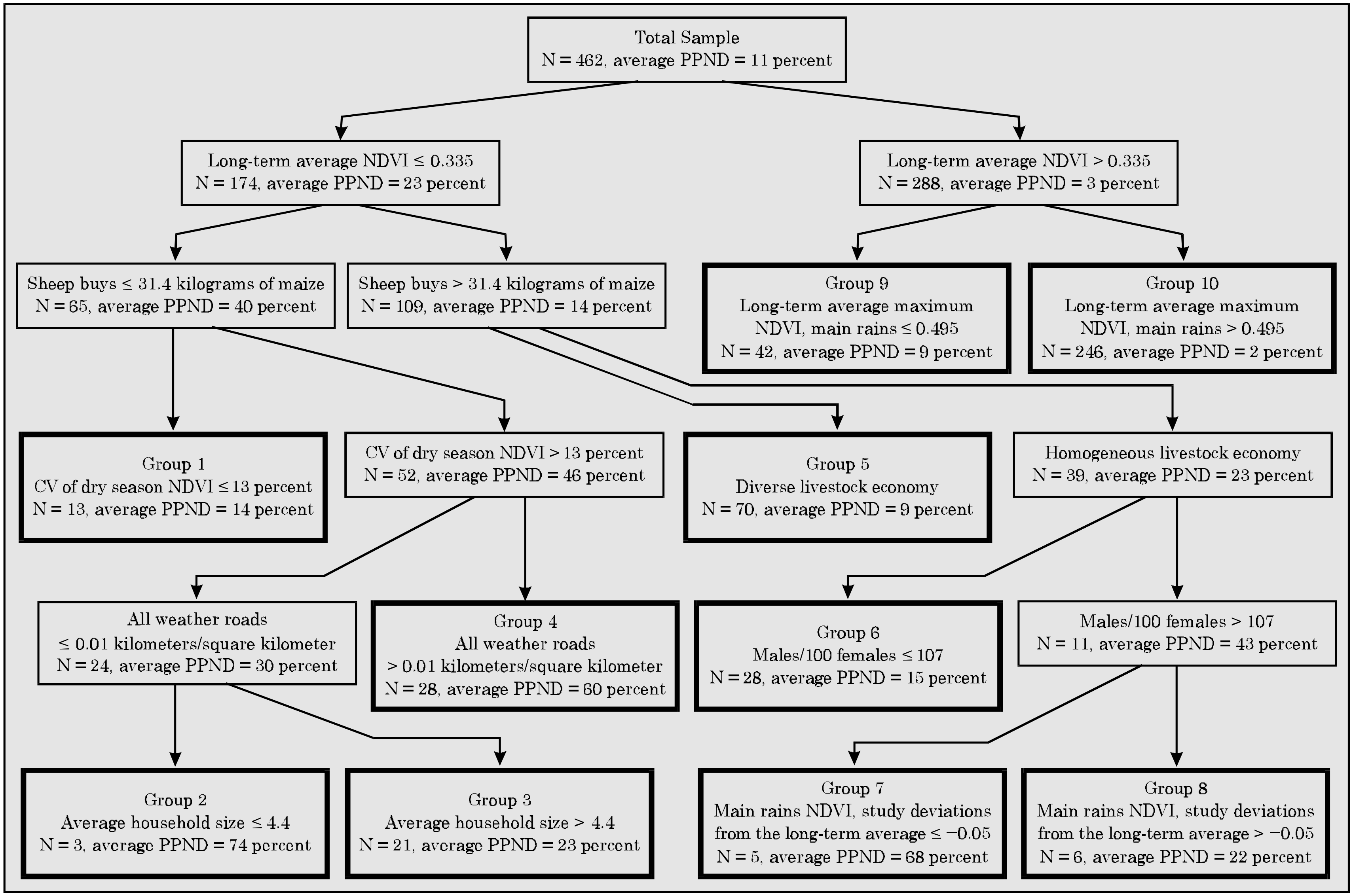
Recall from Chapter 1 that CART produces a classification tree when the dependent variable is categorical and a regression tree when the dependent variable is continuous. The process of constructing a regression tree is similar to that for a classification tree. But in building a regression tree, there is no need to use priors and class assignment rules. Splitting rules, goodness-of-fit criteria, as well as measures of accuracy of a tree in regression tree differ from those for a classification tree. These issues will all be discussed in detail in the two subsections that follow the regression tree example below.

As with classification, regression-tree building centers on three major components: (1) a set of questions of the form, Is $X \leq d$?, where $X$ is a variable and $d$ is a constant; the reponse to such questions is yes or no; (2) goodness-of-split criteria for choosing the best split on a variable; and (3) generation of summary statistics for terminal nodes. The latter component is unique to a regression tree. In classification trees, the terminal nodes are assigned to a specific class according to the class assignment rule. In regression trees, however, there are no classes to which terminal nodes are assigned. Instead, for each of the terminal nodes produced by CART regression, summary statistics of the dependent variable are computed.

The main purpose of CART regression is to produce a tree-structured predictor or prediction rule (Breiman et al. 1984). This predictor serves two major goals: (1) to predict accurately the dependent variable from the future or new values of the predictor variables; and (2) to explain the relationships that exist between the dependent and predictor variables. The CART regression predictor is constructed by detecting the heterogeneity (in terms of variance of the dependent variable) that exists in the data set and then purifyng the data set. CART does this by recursively partitioning a data set into groups or terminal nodes that are internally more homogenous than their ancestor nodes. At each terminal node, the mean value of the dependent variable is taken as the predicted value. If the objective of a regression tree is explanation, then this is achieved by tracking the paths of a tree to a specific terminal node.

An example of a regression tree is given in Figure 5, and the list of variables supplied for generating the tree is given in Table 3.

Figure 5—CART analysis of 77 *awrajas,* 1982–87


Total Sample
N = 462, average PPND = 11 percent
Long-term average NDVI ≤ 0.335
N = 174, average PPND = 23 percent
Long-term average NDVI > 0.335
N = 288, average PPND = 3 percent
Sheep buys ≤ 31.4 kilograms of maize
N = 65, average PPND = 40 percent
Sheep buys > 31.4 kilograms of maize
N = 109, average PPND = 14 percent
Group 9
Long-term average maximum
NDVI, main rains ≤ 0.495
N = 42, average PPND = 9 percent
Group 10
Long-term average maximum
NDVI, main rains > 0.495
N = 246, average PPND = 2 percent
Group 1
CV of dry season NDVI ≤ 13 percent
N = 13, average PPND = 14 percent
CV of dry season NDVI > 13 percent
N = 52, average PPND = 46 percent
Group 5
Diverse livestock economy
N = 70, average PPND = 9 percent
Homogeneous livestock economy
N = 39, average PPND = 23 percent
All weather roads
≤ 0.01 kilometers/square kilometer
N = 24, average PPND = 30 percent
Group 4
All weather roads
> 0.01 kilometers/square kilometer
N = 28, average PPND = 60 percent
Group 6
Males/100 females ≤ 107
N = 28, average PPND = 15 percent
Males/100 females > 107
N = 11, average PPND = 43 percent
Group 2
Average household size ≤ 4.4
N = 3, average PPND = 74 percent
Group 3
Average household size > 4.4
N = 21, average PPND = 23 percent
Group 7
Main rains NDVI, study deviations
from the long-term average ≤ −0.05
N = 5, average PPND = 68 percent
Group 8
Main rains NDVI, study deviations
from the long-term average > −0.05
N = 6, average PPND = 22 percent

Note: An *awraja* is an administrative district of Ethiopia. PPND stands for the percentage of people in need; NDVI stands for normalized difference vegetation index, an estimate of vegetation health; and CV stands for coefficient of variation.

### Table 3—Variables for *awraja*-level analysis

| Variable | Definition |
|---|---|
| *MZSHTTRD* | Retail price of maize/producer price of sheep terms of trade |
| *MZSHTTMN* | Average of *MZSHTTRD* during 1981–87 |
| *MZSHTTDV* | Standard Deviations of __*RD* from __*MN* |
| *MZSHTTCV* | Coefficient of variation of __*RD* during 1981–87 |
| *CERLPROD* | Gross production of all cereals in tons |
| *CERLMN* | Mean of *CERLPROD* during 1981–87 |
| *CERLDV* | Standard Deviations of *CERLPROD* from *CERLMN* |
| *CERLCV* | Coefficient of variation of *CERLPROD* during 1981–87 |
| *PCTBELG* | Percent of annual cereal production from Belg season |
| *PCTBLGMN* | Average of *PCTBELG* during 1981–87 |
| *PCTBLGDV* | Standard Deviations of *PCTBELG* from *PCTBLGMN* |
| *PCTBLGCV* | Coefficient of variation of *PCTBLG* during 1981–87 |
| *CERLPP* | Gross production of all cereals per capita rural population |
| *AVGFAMSZ* | Average size of rural household |
| *DEPRATIO* | Dependency ratio ( and 60 years old /total population 15–59 years old) |
| *LITERATE* | Literacy ratio of males 15 years old /total population 15 years old |
| *TOTFERTR* | Total fertility rate |
| *GENFERTR* | General fertility rate |
| *PAR*4549*R* | Average parity (45–49 years) |
| *ASDRRURL* | Age-specific death rates in rural areas |
| *IMRRURAL* | Infant mortality rate in rural areas |
| *NPERRMRU* | Average people sharing bedroom in rural areas |
| *LIFEEXPR* | Life expectancy in rural areas |
| *CRDBRTHR* | Crude birth rate in rural areas |
| *GRRERRUR* | Gross reproductive rate |
| *MLUPSLRM* | Soil loss rate estimates from Master Land Use Plan |
| *POPUME* | Urban male population |
| *POPUFE* | Urban female population |
| *POPURME* | Rural male population |
| *POPRFE* | Rural female population |
| *ALLKMKM*2 | All-weather road/square kilometer |
| *AVGEP*84*R* | Average land elevation weighted by rural population |
| *HLTHFIND* | Index of health infrastructure based on need |
| *PRPRFHHD* | Share of female heads in total number of household heads |
| *PERENNLO* | Percent farmers with no perennial crops |
| *PERENNL*1-5 | Percent farmers with 1–5 perennial crops |
| *ANNUAL*0 | Percent farmers with no annual crops |
| *ANNUAL*1-8 | Percent farmers with 1–8 annual crops |
| *DISTBGMK* | Distance to large market (kilometers) |
| *DISTSMMK* | Distance to small market (kilometers) |
| *AVGHHINC* | Average household income |
| *GINIHINC* | Gini coefficient of average household income by *awraja* |
| *PCTFRMRS* | Percent rural population who are farmers |
| *AVGPCINC* | Average farm income per capita |

*(continued)*

**Table 3—(continued)**

| Variable | Definition |
|---|---|
| *GINIPINC* | Gini coefficient of *AVGPCINC* by *woreda* weighted by population |
| *PCTFRALW* | Share farmers that always or sometimes plant belg crop |
| *PCTFRSOM* | Share farmers that never plant belg crop |
| *AVGNOXEN* | Average number of oxen owned |
| *PCT0OXEN* | Percent households with no oxen |
| *ANNLPCHA* | Average area cultivated with annual crops per capita |
| *PRNLPCHA* | Average area cultivated with perennial crops per capita |
| *ANLAVG* | Average area cultivated with annual crops by household |
| *PERLAVG* | Average area cultivated with perennial crops by household |
| *FALAVGHA* | Average area fallowed by household |
| *AVGARAHA* | Average arable land owned |
| *PCTIRRIG* | Percent of farmers using irrigation |
| *IRRIGHA* | Total irrigated area |
| *GINITLU* | Gini coefficient of *TLU* ownership (all species) |
| *GINIPCMK* | Gini coefficient of percent crop marketed |
| *PRIM*0014 | Percent children  years old with any schooling |
| *BELGMN* | Average NDVI for Belg season by year |
| *BELGMX* | Maximum NDVI for the season, average for all pixels by *awraja* |
| *BELGMNMN* | __*MN* average for 1982–90 |
| *BELGMXMN* | __*MX* average for 1982–90 |
| *BELGMNCV* | __*MN* coefficient of variation for 1982–90 |
| *BELGMXCV* | __*MX* coefficient of variation for 1982–90 |
| *BELGMNDV* | Standard deviations of __*MN* from __*MNMN* |
| *BELGMXDV* | Standard deviations of __*MX* from __*MXMN* |
| *BELGSDMN* | Standard deviations of season average during 1982–90 |
| *BELGSXMN* | Standard deviations of season maximum during 1982–90 |
| *KIREMMN* | Average NDVI for Kirempt season by year |
| *KIREMMX* | Maximum NDVI for the season, average for all pixels by *awraja* |
| *KIRMNMN* | __*MN* average for 1982–90 |
| *KIRMXMN* | __*MX* average for 1982–90 |
| *KIRMNCV* | __*MN* coefficient of variation for 1982–90 |
| *KIRMXCV* | __*MX* coefficient of variation for 1982–90 |
| *KIRMNDV* | Standard deviations of __*MN* from __*MNMN* |
| *KIRMXDV* | Standard deviations of __*MX* from __*MXMN* |
| *KIRMSDMN* | Standard deviations of season average during 1982–90 |
| *KIRMSXMN* | Standard deviations of season maximum during 1982–90 |
| *BEGAMN* | Average NDVI for Bega season by year |
| *BEGAMX* | Maximum NDVI for the season, average for all pixels by *awraja* |
| *BEGAMNMN* | __*MN* average for 1982–90 |
| *BEGAMXMN* | __*MX* average for 1982–90 |
| *BEGAMNCV* | __*MN* coefficient of variation for 1982–90 |
| *BEGAMXCV* | __*MX* coefficient of variation for 1982–90 |
| *BEGAMNDV* | Standard deviations of __*MN* from __*MNMN* |
| *BEGAMXDV* | Standard deviations of __*MX* from __*MXMN* |

*(continued)*

**Table 3—(continued)**

| Variable | Definition |
|---|---|
| *BEGASDMN* | Standard deviations of season average during 1982–90 |
| *BEGASXMN* | Standard deviations of season maximum during 1982–90 |
| *NDVIMNMX* | Maximum of mean NDVIs for 3 seasons averaged for 1982–90 |
| *NDVIMXMX* | Maximum of season NDVI maxima averaged for 1982–90 |
| *URBPOPSR* | Percent urban population by *awraja* |

Note: An *awraja* is an administrative district in Ethiopia below the province level; a *woreda* is an administrative district below the *awraja* level.

## BUILDING A REGRESSION TREE

The process of constructing a regression tree is similar to that for building a classification tree. Regression-tree building centers on three major components: (1) a set of questions of the form,

$$\text{Is } X \leq d?,$$

where $X$ is a variable and $d$ is a constant. As with classification, the response to such questions is yes or no; (2) goodness-of-split criteria for choosing the best split on a variable; and (3) the generation of summary statistics for terminal nodes (unique to a regression tree).

An example of a regression tree is given in Figure 5, and the list of variables supplied for generating the tree is given in Table 3.

## REGRESSION TREE: EXAMPLE

The regression tree in Figure 5 is based on analysis from a regional vulnerability study in Ethiopia (Seyoum et al. 1995) that uses six years (1982–87) of time-series data collected from 77 administrative regions (*awrajas*) of Ethiopia. The data contain 92 variables, all listed in Table 3. This study of famine (Seyoum et al. 1995) had two specific goals: (1) to determine whether it is possible to estimate or predict the percent of sedentary population in need of food assistance, and (2) to understand the variability in percentages of people in need (*PPND*) across *awrajas* and years. The dependent variable in the study is *PPND*.

The top rectangle in Figure 5 contains a total number of 462 observations (N = 462) with an average *PPND* of 11 percent. (During the six-year period of the study, an average of 11 percent of the population was in need of food assistance.) The regression tree produces 10 terminal nodes or homogenous groups or *awraja* strata. Each group is identified by a number from 1 to 10. The specific path leading from the root node to the terminal node for each group characterizes that group. In Figure 5, *NDVI* (normalized difference vegetation index) is a crude estimate of vegetation health, and is used as an index of greenness. The possible range of values for *NDVI* is between –1 and 1. However, its typical range is between –0.1 (for not a green area) and 0.6 (for a very green area). The higher the index, the greener the vegetation.

The first split of the root node is based on the long-term average *NDVI* variable. This split successfully separates *awrajas* with less green vegetation from *awrajas* with very green vegetation. The long-term average *NDVI* is indeed a powerfully discriminating variable for studying regional vulnerability. In *awrajas* with very green vegetation, average *PPND* is 3 percent, which is much lower than *awrajas* with less green vegetation. *Awrajas* with greener vegetation are further separated using the variable for the long-term average maximum *NDVI* of the main rainy season. This split results in two terminal nodes: Group 9 and Group 10. Predicted *PPND* is 9 percent in Group 9 and 2 percent in Group 10. The low *PPND* for these two groups should not be surprising. It can be argued that these regions have better supplies of food and, hence, food accessibility, than *awrajas* with less green vegetation. Indeed, it turns out that these *awrajas* extend west, south, southwest, and northwest from central Ethiopia (Webb et al. 1994, Map 6.0). These *awrajas* also produce surplus grain in the country. Some *awrajas* in Group 9 do represent pockets of vulnerability in this surplus-producing region.

*Awrajas* in Groups 1 through 8 have at least one characteristic in common. They all descend from *awrajas* with a less green vegetation index (long-term average *NDVI* ≤ 0.335). Group 1 *awrajas* are characterized by low long-term average *NDVI*, low sheep-to-maize terms of trade, and low coefficient of variation of dry season *NDVI*. There are 13 *awrajas* at this terminal node with a predicted *PPND* of 14 percent. The fact that the long-term average *NDVI* is low suggests that the long-term annual average rainfall in these *awrajas* is very low and crop production is limited. This observation is justified by the low sheep-to-maize terms of trade. A household can only buy 31.4 kilograms or less of maize with one sheep, indicating that maize is scarce in these areas. These *awrajas* are in south Gamgofa, northeast Shoa, northeast Bale, and west Hararge regions of Ethiopia. Generally, rainfall in these regions is far below the national average.

*Awrajas* in Group 2 and Group 3 are both characterized by low long-term average *NDVI*, low sheep-to-maize terms of trade, a high coefficient of variation of dry season *NDVI*, and low density of all-weather roads per square kilometer. They are distinct from each other only because of household size. Group 2 *awrajas* have a lower household size than those in Group 3. For the three awrajas in Group 2, predicted *PPND* equals 74 percent. For the 21 *awrajas* in Group 3, predicted *PPND* equals 23 percent. The *awrajas* in these two groups are located in southern Bale, southern Sidamo, eastern Gondar, western Wollo, northeast Wollo, and north Harerge regions of Ethiopia. The transportation network in these regions is limited due to land topography. Not surprisingly, CART characterizes these two groups as low in the density of all-weather roads per square kilometer. The regions in these two groups are also known for being among the most vulnerable to famine in Ethiopia. The remaining terminal nodes can be analyzed in a similar way.

Figure 5 displays the power of CART analysis as did Figure 1. It shows that CART has successfully identified 10 groups of *awrajas* by using only 9 out of the 92 variables submitted for analysis (Table 3). Each group is identified by the path that begins at the root node and ends at its terminal node. The 9 variables along with their split points carry all the information that is needed to differentiate groups of *awrajas* from each other.

## The Steps to Building a Regression Tree

The mechanism for building a regression tree is similar to that for a classification tree. But with a regression tree there is no need to specify priors and misclassification costs. Furthermore, the dependent variable in a regression tree is numeric or continuous. The splitting criterion employed is the within-node sum of squares of the dependent variable and the goodness of a split is measured by the decrease achieved in the weighted sum of squares. Detailed discussion on splitting criteria will be provided further below. The following list highlights the key steps in constructing a regression tree.

1. Starting with the root node, CART performs all possible splits on each of the predictor variables, applies a predefined node impurity measure to each split, and determines the reduction in impurity that is achieved.
2. CART then selects the "best" split by applying the goodness-of-split criteria and partitions the data set into left- and right-child nodes.
3. Because CART is recursive, it repeats steps 1 and 2 for each of the nonterminal nodes and produces the largest possible tree.
4. Finally, CART applies its pruning algorithm to the largest tree and produces a sequence of subtrees of different sizes from which an optimal tree is selected.

## Splitting Rules and Goodness-of-Fit Criteria

There are two splitting rules or impurity functions for a regression tree. These are (1) the Least Squares (LS) function and (2) the Least Absolute Deviation (LAD) function. Since the mechanism for both rules is the same, only the LS impurity measure will be described. Under the LS criterion, node impurity is measured by within-node sum of squares, $SS(t)$, which is defined as

$$SS(t) = \sum (y_{i(t)} - \bar{y}_{(t)})^2, \text{ for } i = 1, 2, \ldots, N_t\,,$$

where $y_{i(t)}$ = individual values of the dependent variable at node $t$, and $\bar{y}_{(t)}$ = the mean of the dependent variable at node $t$. Given the impurity function, $SS(t)$, and split $s$ that sends cases to left ($t_L$) and right ($t_R$) nodes, the goodness of a split is measured by the function

$$\phi(s,t) = SS(t) - SS(t_R) - SS(t_L),$$

where $SS(t_R)$ is the sum of squares of the right child node, and $SS(t_L)$ is the sum of squares of the left child node.

The best split is that split for which $\phi(s,t)$ is the highest. From the series of splits generated by a variable at a node, the rule is to choose that split that results in the maximum reduction in the impurity of the parent node.

An alternative to $SS(t)$ is to use the weighted variance of left and right nodes, where the weights are proportions of cases at nodes $t_L$ and $t_R$: let $p(t) = N_t/N$ be the proportion of cases at node $t$, and let $s^2(t)$ be the variance of the dependent variable at node $t$. The variance is defined as

$$s^2(t) = \frac{1}{N_{(t)}} \sum_{i=1}^{N_t} [y_i - \bar{y}_{(t)}]^2.$$

The goodness of a split is now measured by

$$\phi(s,t) = s^2(t) - [p_L s^2(t_L) + p_R s^2(t_R)].$$

The best split is the one for which $\phi(s,t)$ is the highest or for which the weighted sum of the variances $[p_L s^2(t_L) + p_R s^2(t_R)]$ is the smallest. The procedure successfully separates high values of the dependent variable from its low values and results in left and right nodes that are now internally more homogenous than the parent node. It should be noted that as each split sends observations to the left and right nodes, the mean of the dependent variable in one of the resulting nodes is lower than the mean at the parent node (see the example in Figure 5).

## TREE PRUNING

After building the largest possible tree, CART applies its pruning algorithm by using either cross-validation or an independent test sample to measure the goodness of fit of the tree. LS uses Mean Squared Error (MSE) to measure the accuracy of the predictor in order to rank the sequence of trees generated by pruning. LAD employs Mean Absolute Deviation (MAD). Once a minimal-cost tree (the tree with the lowest MSE OR MAD) is identified, an optimal tree is chosen by applying the one-standard-error rule to the minimal-cost tree. The one-standard-error rule is optional and can be changed by the analyst.

After choosing an optimal tree or, for that matter, any subtree from the sequence of subtrees generated in the pruning process, CART computes summary statistics for each of the terminal nodes. If LS is chosen as a splitting rule, CART computes mean and standard deviations of the dependent variable. The mean of the terminal node becomes the predicted value of the dependent variable for cases in that terminal node. If LAD is selected, CART generates median and average absolute mean deviations of the dependent variable. As with LS, the median becomes the predicted value of the dependent variable for that terminal node.

This form of generating predictions may sound crude to those who are familiar with predictions from parametric models. But it should be noted that CART regression predictions are arrived at by recursively splitting the sample and creating groups or clusters that are progressively more homogenous than their ancestor nodes. Breiman et al. (1984) suggest running OLS models in each group created by the regression tree and comparing the OLS predictions against each other. A considerable difference between the predicted values of OLS models for each group is an indication that CART has succeeded in uncovering the complex structure existing in the data set.

# 5 CART SOFTWARE AND PROGRAM CODES

CART software is currently available for different platforms, as shown in Table 4. Details on the current versions of CART software that are compatible with different platforms may be obtained from the vendor listed in Table 4.

The software comes with two completely documented manuals that are easy to follow. The first manual (Steinberg and Colla 1995) provides a comprehensive background and conceptual basis for understanding CART. It also discusses the art of tree-structured data analysis, provides detailed listings and explanations of CART commands in SYSTAT syntax, and explains how to use CART techniques and interpret results. Even though CART commands are in SYSTAT syntax, CART software is a stand-alone application that does not need SYSTAT software. The second manual (Steinberg, Colla, and Martin 1998) is for the Windows operating systems (Windows 3.x and Windows 95/NT). A detailed tutorial covers the use of menus, the mouse, the graphic interface, and many other features that are specific to the Windows version.

The graphic interface feature of Windows is an extremely useful tool for CART data analysts. Windows enables CART simultaneously to show tree topology and the quality of an optimal tree through a graphic display of relative costs of trees versus the number of terminal nodes. CART's node navigator feature enables the analyst to immediately perform exploratory work on trees of different sizes and determine node summary information for each examined tree. Thus the analyst can inspect different trees immediately in case the optimal tree becomes unsatisfactory. Any tree can be inspected by clicking on a tree from the series displayed graphically at the lower panel of the node navigator. Node summary information for each tree can be generated for the level of detail desired. The results are displayed graphically in the form of an inverted tree. This is an improvement over earlier versions of CART, in which tree-structured graphs had to be produced manually. In the Windows version the analyst is not limited to using only menus. He/she can write CART commands in batch mode and submit them for analysis while making use of all other features available in Windows.

The rest of this chapter introduces basic CART commands and batch mode programs written in SYSTAT syntax. A few basic CART commands are provided in Table 5. For greater detail about CART commands, the reader should refer to Steinberg and Colla (1995) or contact the vendor listed in Table 4.

**Table 4—Hardware and software requirements of CART for personal computers**

| Hardware and software | |
|---|---|
| Hardware requirements: | Intel PCs, SUN, SGI, HP, Digital Alpha and VAX, IBMRS600 |
| Operating systems supported: | Windows 3.X, Windows 95, Windows NT, MacOS, UNIX, IBM MVS and CMS |
| Memory requirements: | May vary with versions of CART software. CART for Windows is compiled for machines with at least 32 megabytes of RAM. For optimal performance, Pentium machines with at least 32 megabytes of RAM are recommended. |
| Hard disk space: | At least 10 megabytes for software storage |
| Company name: | Salford Systems |
| Address: | 8888 Rio San Diego Dr., Suite 1045<br>San Diego, California 92108 U.S.A. |
| Web address: | http://www.salford-systems.com |
| Telephone: | (619) 543–8880 |
| Fax: | (619) 543–8888 |
| Technical support: | Available either by telephone, fax, or letter. |
| Number of variables and observations: | Computing requires a minimum of 16 megabytes of free memory. Number of observations and variables supported depend on the available memory. |

Source: Fax message received from Salford Systems, February 1998, and http://www.salford-systems.com/technical-CART.html, July 9, 1998.

## PREPARATION OF CART DATA FILES

CART can only read and process data files that are in SYSTAT format. Therefore, the data for analysis should be prepared in SYSTAT. If data are in other formats, they should be converted to a SYSTAT format using either DBMSCOPY or the translation utility that comes with CART software.

## ACCESSING CART

CART can be invoked in two ways. The DOS version can be accessed by typing CART at the prompt of the operating system and pressing the enter key. In the Windows version, CART is invoked by double-clicking on the CART icon.

## CART COMMANDS IN BATCH MODE

CART commands should be written in SYSTAT syntax using any available editor. The following commands produce a classification tree.

```
USE 'D:\CART1989\POOLSUB5.SYS'
CATEGORY CUTDUM2
MODEL CUTDUM2
BUILD
```

### Table 5—Basic CART software commands in SYSTAT

| Command | Syntax | Function (purpose) | Examples |
|---|---|---|---|
| USE | USE *filename* | Specifies a file to read | USE c:\CART\test1.sys |
| EXCLUDE | EXCLUDE *variable list* | Excludes from file the variables not needed in the analysis | EXCLUDE hhid code |
| KEEP | KEEP *variable list* | Reads from the file only the variables needed in the analysis | KEEP age sex income |
| CATEGORY | Category *variable list* | Specifies list of categorical variables in the data set, including the dependent variable—this is compulsory in a classification tree | CATEGORY sex |
| DEL | MODEL *variable name* | Specifies dependent variable | MODEL vulner10 |
| BUILD | BUILD | Tells CART to produce a tree | BUILD |
| QUIT | QUIT | If submitted while in BUILD, it tells CART to quit the session; if submitted after CART session, it tells CART to go to DOS. | |
| SELECT | SELECT *variable name* relation operator or constant/character<br>or | Selects a subset of the data set for analysis | SELECT age > 15<br>SELECT sex=1<br>SELECT X≥20<br>SELECT x1= 'M' |
| SELECT | SELECT *variable name* relation operator or constant/character, *variable name* relation operator or constant/character | Selects a subset of the data set for analysis | SELECT age > 15, Wage > 300 |
| PRIORS | PRIORS *option* (Choose 1 option only) | Specifies which PRIORS to use | PRIORS data<br>PRIORS equal<br>PRIORS mixed<br>PRIORS=n1, n2,,..,na (n's are real numbers) |
| MISCLASS COST | MISCLASS COST=n classify I as k1,k2,k3/, Cost=m classify I as k1/, Cost=l classify k1,k2,..,kn as x | Assigns nonunit misclassification costs | Misclass cost=2 classify 1 as 2,3,4/, Cost=5 classify 3 as 1 Cost=3 classify 1,2,3 as 4 |
| METHOD | METHOD=options (choose 1 option only) | Specifies splitting rule | Method=gini (default) or<br>Method=twoing or<br>Method=LS or LAD<br>Method=LINEAR |
| OUTPUT | OUTPUT *filename* | Sends output to a named file | OUTPUT=LMS |
| TREE | TREE *tree file name* | Specifies a file name of a tree to save | TREE vulner1 |
| SAVE | SAVE filename options | Specifies file name of a data set with predicted class(es), select options to save | SAVE predct1 |
| CASE | CASE options | Runs data one by one down a tree, select option(s) to use | CASE |

These four lines are mandatory. They are the only commands needed to produce a classification tree. For a regression tree, the CATEGORY command line is not needed at all, and the dependent variable that follows the MODEL command should be a continuous variable. To produce a regression tree, the only three commands needed are USE, MODEL, and BUILD. Examples of regression-tree command lines are provided toward the end of this chapter.

The data analyst has many options to modify this program. All optional command lines are additions to this basic program. Any optional command line(s) should be entered before the BUILD command. For example, if the analyst wants to save the output to a file, the OUTPUT command should be inserted as follows:

```
Syntax:    OUTPUT 'd:\cart1989\any name'
```

With the addition of the OUTPUT command, the entire program would read:

```
USE 'D:\CART1989\POOLSUB5.SYS'
CATEGORY CUTDUM2
MODEL CUTDUM2
OUTPUT 'D:\CART1989\VPDAT.DAT'
BUILD
```

The OUTPUT command sends the output results to a file named VPDAT.DAT.

## PROGRAM REFINEMENTS

Sometimes the initial program may not produce a satisfactory tree. In such cases, the program can be modified in a number of ways. The easiest way is to change either priors or misclassification costs or both. If priors are not specified by the analyst, the default is priors equal. The analyst can also change the default splitting rules, the one-standard-error rule, the complexity parameter, and so on. This manual covers only the simplest options.

### Refinement 1

The default priors can be changed by choosing either PRIORS DATA or PRIORS MIXED and adding it into the batch program. For example, if PRIORS DATA is chosen, the modified program will look like this:

```
USE 'D:\CART1989\POOLSUB5.SYS'
CATEGORY CUTDUM2
MODEL CUTDUM2
PRIORS DATA
OUTPUT 'D:\CART1989\VPDAT.DAT'
BUILD
```

### Refinement 2

In addition to changing priors to “data” or “mixed,” the analyst can also incorporate external information into the program by assigning explicit values to priors. In such cases, the underlying assumption is that the distribution of observations into classes of the dependent variable may occur in proportions other than priors equal, priors data, or priors mixed.

For example, in a two-class problem, the analyst may assign

```
PRIORS = .2, .8, or
PRIORS = 1, 5, or
PRIORS = 1.2, 1, and so on.
```

The latter priors says that the proportion of Class 0 cases in the population from which the sample is drawn is 20 percent higher than the proportion of Class 1 cases.

With these changes, the program looks like this:

```
USE 'D:\CART1989\POOLSUB5.SYS'
CATEGORY CUTDUM2
MODEL CUTDUM2
PRIORS = 1, 5
OUTPUT 'D:\CART1989\VPDAT.DAT'
BUILD
```

### Refinement 3

So far, the analysis is based on equal or unit misclassification costs, which is the default setting. This setting can be changed by imposing severe costs for misclassifying certain serious cases. If a heart-attack patient is misclassified as a healthy individual during medical diagnosis, the cost is far more serious than the cost of classifying a healthy individual as a heart-attack patient. In vulnerability studies, classifying food-insecure households as food-secure is more costly than classifying food-secure households as food-insecure. Two options are available for reducing the misclassification of such serious cases.

1. Change the misclassification costs via altered priors. For example, suppose classifying Class 1 cases as Class 0 is three times more costly than classifying Class 0 cases as Class 1. This situation can be treated as if the distribution of Class 1 cases in the population is three times as large as that of Class 0. This information is entered in the PRIORS command line, and the entire batch program now reads as follows:

```
USE 'D:\CART1989\POOLSUB5.SYS'
CATEGORY CUTDUM2
MODEL CUTDUM2
PRIORS = 1, 3
OUTPUT 'D:\CART1989\VPDAT.DAT'
BUILD
```

2. Introduce misclassification costs explicitly into the command line.

```
Example:    MISCLASS COST = 5 CLASSIFY 0 AS 1,
                     COST = 2 CLASSIFY 1 AS 0.
```

This means that the cost of classifying a Class 0 case as Class 1 = 5, while the cost of classifying a Class 1 case as Class 0 is 2. The example associates different penalties or costs with each misclassification error.

With these additions, the program looks like the following:

```
USE 'D:\CARAT\POOLSUB5.SYS'
EXCLUDE SITE HHID
CATEGORY CUTDUM2
MODEL CUTDUM2
PRIORS DATA
MISCLASS COST = 5 classify 0 as 1,
COST = 2 classify 1 as 0
OUTPUT 'D:\CARAT\VPDAT.DAT'
BUILD
```

### Refinement 4

This refinement involves the MODEL command. The analyst may limit the number of variables in the analysis by explicitly specifying the model as in a parametric model. This option is helpful especially in cases where it may not be possible to access a computer with a large memory.

```
Example:   MODEL CUTDUM2 = NCERYL80 + PCLSU80
               + GINI + PCDCALS + FARMINC + HHSIZE.
```

One can also use the EXCLUDE command to exclude variables that are not needed in the analysis.

### Refinement 5

The data analyst may change the default splitting rule (Gini criteria) by using the METHOD command. For example, METHOD = LINEAR changes the default splitting criteria to linear combination splits. In this case, the METHOD command should follow the MODEL command. Under this splitting criteria, CART assumes that all of the variables in the linear combination are numeric. Therefore, unless categorical variables are transformed into sets of dummy variables, they will be treated as numeric variables.

## REGRESSION TREE PROGRAM CODES

The commands needed for producing a regression tree are basically the same as that for a classification tree. There is no need to specify the CATEGORY and MISCLASS COST commands in regression tree programs. As pointed out earlier, the three basic commands that are needed for producing a regression tree are the USE, MODEL, and BUILD commands.

Consider the following typical regression-tree programs:

(A)

```
USE 'D:\CARAT\YEAR8187.SYS'
MODEL PPND  =  NDVIMNMX KRMTMNMN NDVIMXMX
               KRMTMXMN BEGAMNMN BEGAMXMN
               MZSHTTRD MZSHTTDV BEGAMNDV
               BELGMNDV KRMTMXDV
OUTPUT 'D:\CARAT\YEAR8187.OUT'
BUILD
```

(B)

```
USE 'D:\CARAT\YEAR8187.SYS'
MODEL PPND
OUTPUT 'D:\CARAT\YR01.OUT'
BUILD
```

As with classification trees, the OUPUT command is optional. The analyst can modify this basic program by adding any of the available optional command lines into the program. In example (A), the dependent and independent variables are specified in the MODEL command. This option is useful in situations where access to a computer with large memory is limited. Option (B) uses all of the available variables in the data set and produces a regression tree. This option is especially useful if the analyst does not have any prior information about which predictors or potential predictors to use in the model.

## SAVING CART TREES FOR FUTURE USE

It maybe useful to recall that the main objective of running either classification or regression trees is to use the resulting tree for classifying data or predicting the class of a new observation. CART does this by dropping the data down the tree case by case, beginning from the root node. At each stage the splitting criteria are applied until the observations end up in any one of the terminal nodes. This task is accomplished by using only the USE, TREE, SAVE, and CASE commands. It should be noted that the extension of the filename created by the TREE command is always TR1 and cannot be changed. The complete program for building and saving a tree is as follows:

```
USE 'D:\CARAT\POOLSUB5.SYS'
CATEGORY CUTDUM2
MODEL CUTDUM2
TREE SECUR1
BUILD
```

The TREE command produces a file called SECUR1.TR1.

Suppose the analyst has a new data set called DATANEW.SYS, which contains the characteristics of new cases with an unknown class distribution. The analyst now wants to run this data down the saved tree (SECUR1.TR1) to find out the classes into which the new cases fall, and to save the case-by-case results in a data file called PREDCT.SYS. Using the CASE command line, this is written as follows:

```
USE 'D:\DATANEW.SYS'
TREE SECUR1
SAVE PREDCT / SINGLE
IDVAR HHID
CASE
```

The IDVAR command line adds the identification variable (HHID) to the file PREDCT.SYS, which is created by the CASE command. The contents of the PREDCT.SYS file include the original variables used in the model and a few new variables created by CART. The RESPONSE and CORRECT variables are the most useful of the new variables. The RESPONSE variable contains the class assigned to an observation by CART. The CORRECT variable is an indicator variable. It equals 1 for correct prediction and 0 for incorrect prediction.

# 6 REFINING CART ANALYSES

At times, it may not be possible to get the desired results from the first CART session. CART may not even produce any tree at all. To overcome these problems, some of the alternative refinements introduced in Chapter 5 may need to be applied. The structure of the trees produced may differ with each alternative. That is, the variables upon which the splits are made and the number of terminal nodes may change. Even the removal of a single variable from analysis produces a tree with a different structure. For these reasons, CART reports the cross-validated relative-error costs for a tree along with the standard errors (Breiman et al. 1984). The contingent structure of the trees raises the issue of which classification tree to choose and how to choose it. CART does a good job of producing a number of useful classification tables for each alternative based on the learning sample and cross-validation tests (see Appendix 1, Example 1). Since the goal of a classification tree is to enable the analyst to predict the class of future observations, more attention should be paid to the analysis of cross-validation classification and cross-validation classification probability tables. Of course, the choice of the tree ultimately depends on what the analyst intends to do with the tree.

To illustrate the issue of choice, several alternatives to the CART results discussed in Figure 1 in Chapter 2 are produced. The complete CART output is provided in Examples 1, 2, and 3 of Appendix 1 on the diskette. Condensed versions are provided in Examples 1, 2, and 3 of the hard copy of Appendix 1. For comparative analysis, the cross-validation classification probability is extracted from the output of the three alternative models and given below in Table 6.

Example 1 in Table 6 is based on the assumption of PRIORS EQUAL, Example 2 is based on PRIORS DATA, and Example 3 on PRIORS MIXED. For the tree in Example 1, the cross-validated error rate equals 0.634 +/– 0.058, the resubstitution estimate is 0.430, and the total correct classification is 69.2 percent (see Appendix 1, Example 1). For the tree in Example 2, the cross-validated error rate is 0.921 +/– 0.077, the resubstitution estimate is 0.663, and the total correct classification is 75.7 percent (see Appendix 1, Example 2). And finally, for the tree in Example 3, the cross-validated error rate is 0.782 +/– 0.066, the resubstitution estimate is 0.537, and the total correct classification is 73.7 percent (see Appendix 1, Example 3).

In Table 6, a matrix of predicted class probabilities is provided for each example. Under Example 1, the classification tree predicted 70.3 percent of the nonvulnerable households as nonvulnerable and 66.3 percent of the vulnerable households as vulnerable. These are very encouraging results. But can the predictions be improved? Under example 2, 88.4 percent of the nonvulnerable households were predicted to be

Table 6—Cross-validation classification probability comparisons

| | | Predicted Class | | |
|---|---|---|---|---|
| Example | Actual Class | 0 | 1 | Actual total |
| 1 | Priors equal | | | |
| | 0 | 0.703 | 0.297 | 1.00 |
| | 1 | 0.337 | 0.663 | 1.00 |
| 2 | Priors data | | | |
| | 0 | 0.884 | 0.116 | 1.00 |
| | 1 | 0.596 | 0.404 | 1.00 |
| 3 | Priors mixed | | | |
| | 0 | 0.815 | 0.185 | 1.00 |
| | 1 | 0.483 | 0.517 | 1.00 |

nonvulnerable, but only 40.4 percent of the vulnerable households were predicted as vulnerable. This is not a desirable outcome because of the high error rate in predicting vulnerable households. The analyst has to think of which error rate is more costly in terms of misclassification. The results for Example 3 fall between the results of Examples 1 and 2.

The classification tree produced under the assumption of PRIORS DATA provides a better overall correct classification rate (75.7 percent) than the other trees (see Appendix 1, Examples 1, 2, and 3). But the tree in Example 1 performs best when it comes to classifying the vulnerable group. This tree correctly classifies 66.3 percent of the vulnerable households. Furthermore, comparative analysis of the predictive error rates of the three examples clearly shows that the tree of Example 1 has the smallest error rates. Thus, the classification tree in Example 1 provides the best classifiers or indicators of vulnerability. However, the final choice depends on the analyst.

There are still many other options available to the analyst. The results for some of these options are given in Examples 1, 2, and 3 on the diskette (Appendix 3, which only appears on the diskette). In these optional runs, alternative misclassification costs were added to the program to see if there were any improvements in the overall misclassification rate. No improvements resulted.

# 7 CONCLUSIONS

This manual has laid out the fundamental theory underlying Classification and Regression Tree (CART) analytical techniques, and also explained how such techniques can be applied in practice. Concrete examples were presented from research at IFPRI. This research has explored the potential of CART to provide a less subjective framework for the selection of famine risk indicators and determine the relative importance of such indicators in explaining vulnerability across years and regions in Ethiopia.

The theoretical exposition and the results from applied CART analysis suggest that this methodology offers considerable potential for assisting in the analysis of large and complex data sets. CART also offers a transparent, “objective” methodology upon which planners can base their decisions.

That said, CART should be seen as *one* tool that can be used, in conjunction with others, for analyzing data, assessing risk, and planning development. The technique is extremely data-intensive and, hence, labor-intensive (in terms of the time an analyst spends collating, preparing, and analyzing the data). What is more, there remains a need for further research into the definition of appropriate benchmark indicators (such as the “population in need” figures used here), against which multiple variables can be tested. In the short run, the choice of indicators will most likely be driven by data availability. But in the longer run, such choices should be made as a result of assessments of the reliability and sensitivity of alternatives.

Further exploration of the gains and drawbacks inherent in CART are therefore encouraged, and not just in relation to research on food security. As IFPRI and others have demonstrated, CART can be usefully applied to a wide range of uses.

# APPENDIX 1: CONDENSED EXAMPLES OF CLASSIFICATION-TREE OUTPUT

*(Full output on diskette)*

## EXAMPLE 1: CLASSIFICATION-TREE OUTPUT BASED ON *PRIORS EQUAL*

Tree sequence and cross-validation tables are extracted from Appendix 1, Example 1 on the diskette. Partial CART output is based on *priors equal* (for details, see attached diskette).

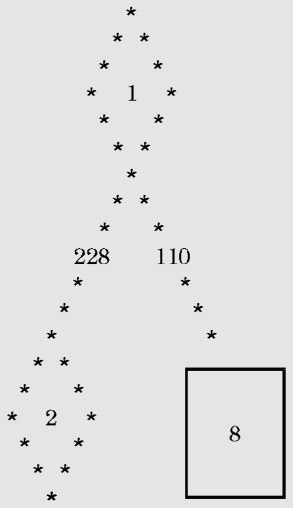
### Tree sequence

| Tree | Number of terminal nodes | Cross-validated relative cost | Resubstitution relative cost | Complexity parameter |
|---|---|---|---|---|
| 1 | 32 | 0.704 +/– 0.060 | 0.145 | 0.000 |
| 8 | 16 | 0.639 +/– 0.058 | 0.244 | 0.008 |
| 9 | 14 | 0.635 +/– 0.058 | 0.276 | 0.008 |
| 10 | 12 | 0.632 +/– 0.058 | 0.310 | 0.008 |
| 11* | 11 | 0.603 +/– 0.057 | 0.332 | 0.011 |
| 12** | 8 | 0.634 +/– 0.058 | 0.430 | 0.016 |
| 13 | 7 | 0.668 +/– 0.059 | 0.464 | 0.017 |
| 14 | 5 | 0.687 +/– 0.059 | 0.540 | 0.019 |
| 15 | 3 | 0.700 +/– 0.058 | 0.619 | 0.020 |
| 16 | 2 | 0.729 +/– 0.048 | 0.696 | 0.038 |
| 17 | 1 | 1.000 +/– 0.000 | 1.000 | 0.152 |

Initial misclassification cost = 0.500
Initial class assignment = 0

* indicates minimum-cost tree.
** indicates optimum-cost tree.

## Node information



Node 1 was split on variable *NCERYL*80.
A case goes left if variable *NCERYL*80 ≤ 4.714.
Improvement = 0.061 C. T. = 0.152

| Node | Cases | Class | Cost |
|---|---|---|---|
| 1 | 338 | 0 | 0.500 |
| 2 | 228 | 1 | 0.398 |
| −8 | 110 | 0 | 0.200 |

| | Number of cases | | | Within-node probability | | |
|---|---|---|---|---|---|---|
| Class | Top | Left | Right | Top | Left | Right |
| 0 | 249 | 148 | 101 | 0.500 | 0.398 | 0.800 |
| 1 | 89 | 80 | 9 | 0.500 | 0.602 | 0.200 |

| Surrogate | | Split | Association | Improvement |
|---|---|---|---|---|
| 1 *NCERAR*80 | s | 0.026 | 0.777 | 0.058 |
| 2 *PCAGINC* | s | 400.220 | 0.051 | 0.002 |
| 3 *PCDCALS* | s | 5757.189 | 0.032 | 0.000 |
| 4 *CERLAR*80 | s | 3.486 | 0.030 | 0.001 |
| 5 *PCFRMINC* | s | 345.744 | 0.030 | 0.006 |

| Competitor | Split | Improvement |
|---|---|---|
| 1 *NCERAR*80 | 0.026 | 0.058 |
| 2 *FARMYRAT* | 87.298 | 0.041 |
| 3 *HHSIZE* | 5.500 | 0.033 |
| 4 *LVSLSU*80 | 0.600 | 0.030 |
| 5 *PCLSU*80 | 2.893 | 0.027 |

Note: C.T. stands for Complexity Threshold (the complexity parameter used in tree pruning).

## Terminal node information

| Node | N | Probability | Class | Cost | Class | N | Probability | Complexity threshold |
|---|---|---|---|---|---|---|---|---|
| 1 | 88 | 0.242 | 0 | 0.418 | 0 | 70 | 0.582 | 0.039 |
| | | | | | 1 | 18 | 0.418 | |
| 2 | 24 | 0.106 | 1 | 0.152 | 0 | 8 | 0.152 | 0.039 |
| | | | | | 1 | 16 | 0.848 | |
| 3 | 50 | 0.227 | 1 | 0.133 | 0 | 15 | 0.133 | 0.019 |
| | | | | | 1 | 35 | 0.867 | |
| 4 | 19 | 0.038 | 0 | 0.000 | 0 | 19 | 1.000 | 0.028 |
| | | | | | 1 | 0 | 0.000 | |
| 5 | 9 | 0.040 | 1 | 0.152 | 0 | 3 | 0.152 | 0.028 |
| | | | | | 1 | 6 | 0.848 | |
| 6 | 35 | 0.078 | 0 | 0.145 | 0 | 33 | 0.855 | 0.017 |
| | | | | | 1 | 2 | 0.145 | |
| 7 | 3 | 0.017 | 1 | 0.000 | 0 | 0 | 0.000 | 0.017 |
| | | | | | 1 | 3 | 1.000 | |
| 8 | 110 | 0.253 | 0 | 0.200 | 0 | 101 | 0.800 | 0.152 |
| | | | | | 1 | 9 | 0.200 | |

## Misclassification by class

| | Cross-validation | | | | Learning sample | | |
|---|---|---|---|---|---|---|---|
| Class | Prior probability | N | Number misclassified | Cost | N | Number misclassified | Cost |
| 0 | 0.500 | 249 | 74 | 0.297 | 249 | 26 | 0.104 |
| 1 | 0.500 | 89 | 30 | 0.337 | 89 | 29 | 0.326 |
| Total | 1.000 | 338 | 104 | | 338 | 55 | |

### Cross-validation classification

| | | Predicted class | | |
|---|---|---|---|---|
| **Actual Class** | | **0** | **1** | **Actual total** |
| 0 | | 175.000 | 74.000 | 249.000 |
| 1 | | 30.000 | 59.000 | 89.000 |
| Predicted total | | 205.000 | 133.000 | 338.000 |
| Correct | | 0.703 | 0.663 | |
| Success indicator | | −0.034 | 0.400 | |
| Total correct | | 0.692 | | |
| Sensitivity | 0.703 | | | |
| Specificity | 0.663 | | | |
| False reference | 0.146 | | | |
| False response | 0.556 | | | |
| Reference = Class 0 | | | | |
| Response = Class 1 | | | | |

### Cross-validation classification probability

| | Predicted class | | |
|---|---|---|---|
| **Actual class** | **0** | **1** | **Actual total** |
| 0 | 0.703 | 0.297 | 1.000 |
| 1 | 0.337 | 0.663 | 1.000 |

### Learning-sample classification

| | | Predicted class | | |
|---|---|---|---|---|
| **Actual class** | | **0** | **1** | **Actual total** |
| 0 | | 223.000 | 26.000 | 249.000 |
| 1 | | 29.000 | 60.000 | 89.000 |
| Predicted total | | 252.000 | 86.000 | 338.000 |
| Correct | | 0.896 | 0.674 | |
| Success indicator | | 0.159 | 0.411 | |
| Total correct | | 0.837 | | |
| Sensitivity | 0.896 | | | |
| Specificity | 0.674 | | | |
| False reference | 0.115 | | | |
| False response | 0.302 | | | |
| Reference = Class 0 | | | | |
| Response = Class 1 | | | | |

### Learning-sample classification probability

| | Predicted class | | |
|---|---|---|---|
| **Actual class** | **0** | **1** | **Actual total** |
| 0 | 0.896 | 0.104 | 1.000 |
| 1 | 0.326 | 0.674 | 1.000 |

### Relative importance of variables

| Variable | Relative importance |
|---|---|
| *NCERYL*80 | 100.000 |
| *NCERAR*80 | 94.086 |
| *PCLSU*80 | 78.891 |
| *PCFRMINC* | 68.257 |
| *LVSLSU*80 | 66.941 |
| *OXQ*80 | 65.267 |
| *PCAGINC* | 55.699 |
| *PCFRMAST* | 54.965 |
| *FARMYRAT* | 47.661 |
| Gini | 44.670 |
| *PCINC* | 44.368 |
| *AGINCRAT* | 37.161 |
| *HHSIZE* | 35.707 |
| *NFRMYRAT* | 33.440 |
| *FRMASRAT* | 27.147 |
| *NFRMASRA* | 25.433 |
| *PCNNFINC* | 24.009 |
| *PCNFRAST* | 21.933 |
| *PCLIVINC* | 16.946 |
| *CERYLD*80 | 12.768 |
| *PCDCALS* | 9.451 |
| *PCAST*80 | 6.156 |
| *CERLAR*80 | 1.405 |
| *LIVSYRAT* | 0.842 |
| *HHEADSEX* | 0.000 |
| *CALDUM* | 0.000 |

## EXAMPLE 2: CLASSIFICATION-TREE OUTPUT BASED ON *PRIORS DATA*

Tree sequence and cross-validation tables are extracted from Appendix 1, Example 2 on the diskette. Partial CART output is based on *priors data* (for details, see attached diskette).

### Tree sequence

| Tree | Number of terminal nodes | Cross-validated relative cost | Resubstitution relative cost | Complexity parameter |
|---|---|---|---|---|
| 1 | 34 | 1.011 +/– 0.089 | 0.180 | 0.000 |
| 2 | 31 | 1.000 +/– 0.088 | 0.180 | 0.000 |
| 3 | 26 | 0.955 +/– 0.086 | 0.225 | 0.002 |
| 4 | 19 | 0.933 +/– 0.084 | 0.303 | 0.003 |
| 5 | 17 | 0.910 +/– 0.082 | 0.337 | 0.004 |
| 6* | 12 | 0.865 +/– 0.078 | 0.449 | 0.006 |
| 7 | 8 | 0.888 +/– 0.077 | 0.584 | 0.009 |
| 8** | 6 | 0.921 +/– 0.077 | 0.663 | 0.010 |
| 9 | 5 | 1.000 +/– 0.069 | 0.719 | 0.015 |
| 10 | 1 | 1.000 +/– 0.000 | 1.000 | 0.019 |

Initial misclassification cost = 0.263
Initial class assignment = 0

* indicates minimum-cost tree.
** indicates optimum-cost tree.

### Cross-validation classification

| | | Predicted class | | Actual |
|---|---|---|---|---|
| Actual class | | 0 | 1 | total |
| 0 | | 220.000 | 29.000 | 249.000 |
| 1 | | 53.000 | 36.000 | 89.000 |
| Predicted total | | 273.000 | 65.000 | 338.000 |
| Correct | | 0.884 | 0.404 | |
| Success indicator | | 0.147 | 0.141 | |
| Total correct | | 0.757 | | |
| Sensitivity | 0.884 | | | |
| Specificity | 0.404 | | | |
| False reference | 0.194 | | | |
| False response | 0.446 | | | |
| Reference = Class 0 | | | | |
| Response = Class 1 | | | | |

### Cross-validation classification probability

| | Predicted class | | |
|---|---|---|---|
| Actual class | 0 | 1 | Actual total |
| 0 | 0.815 | 0.185 | 1.000 |
| 1 | 0.483 | 0.517 | 1.000 |

## EXAMPLE 3: CLASSIFICATION-TREE OUTPUT BASED ON *PRIORS MIXED*

Tree sequence and cross-validation tables are extracted from Appendix 1, Example 3 on the diskette. Partial CART output is based on *priors mixed* (for details, see attached diskette).

### Tree sequence

| Tree | Number of terminal nodes | Cross-validated relative cost | Resubstitution relative cost | Complexity parameter |
|---|---|---|---|---|
| 1 | 30 | 0.757 +/– 0.067 | 0.190 | 0.000 |
| 6 | 22 | 0.764 +/– 0.067 | 0.250 | 0.006 |
| 7* | 20 | 0.748 +/– 0.067 | 0.284 | 0.006 |
| 8 | 18 | 0.761 +/– 0.067 | 0.318 | 0.007 |
| 9 | 16 | 0.757 +/– 0.067 | 0.359 | 0.008 |
| 10 | 10 | 0.764 +/– 0.067 | 0.490 | 0.008 |
| 11** | 8 | 0.782 +/– 0.066 | 0.537 | 0.009 |
| 12 | 6 | 0.820 +/– 0.067 | 0.593 | 0.011 |
| 13 | 4 | 0.830 +/– 0.065 | 0.654 | 0.012 |
| 14 | 3 | 0.910 +/– 0.062 | 0.764 | 0.042 |
| 15 | 1 | 1.000 +/– 0.000 | 1.000 | 0.045 |

Initial misclassification cost = 0.382
Initial class assignment = 0

* indicates minimum-cost tree.
** indicates optimum-cost tree.

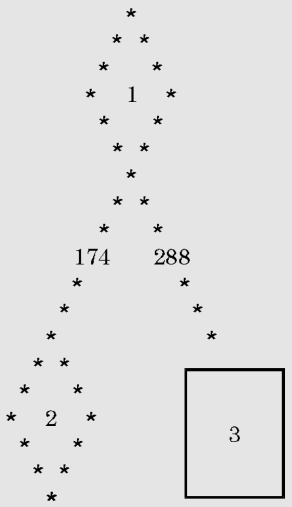
### Cross-validation classification

| | | Predicted class | | Actual total |
|---|---|---|---|---|
| **Actual class** | | **0** | **1** | |
| 0 | | 203.000 | 46.000 | 249.000 |
| 1 | | 43.000 | 46.000 | 89.000 |
| Predicted total | | 246.000 | 92.000 | 338.000 |
| Correct | | 0.815 | 0.517 | |
| Success indicator | | 0.079 | 0.254 | |
| Total correct | | 0.737 | | |
| Sensitivity | 0.815 | | | |
| Specificity | 0.517 | | | |
| False reference | 0.175 | | | |
| False response | 0.500 | | | |
| Reference = Class 0 | | | | |
| Response = Class 1 | | | | |

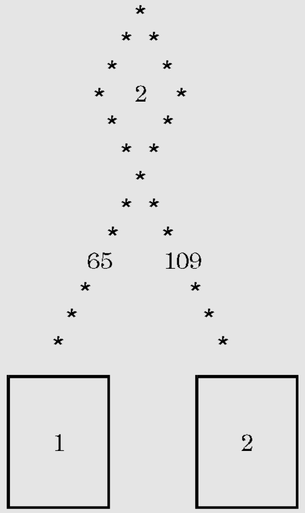
### Cross-validation classification probability

| | Predicted class | | |
|---|---|---|---|
| **Actual class** | **0** | **1** | **Actual total** |
| 0 | 0.815 | 0.185 | 1.000 |
| 1 | 0.483 | 0.517 | 1.000 |

# APPENDIX 2: A CONDENSED EXAMPLE OF REGRESSION-TREE OUTPUT *(Full output on diskette)*



```
Node 1 was split on variable NDVIMNMX
A case goes left if variable NDVIMNMX ≤ 0.335000
Improvement = 95.212097      C. T. = 0.439885E + 0.05

Node          Cases          Average          Standard Deviation

   1            462        10.902165                19.447115
   2            174        23.455744                25.835199
  –3            288         3.317708                 7.119483

Surrogate                   Split      Association      Improvement

1 KRMTMNMN       s      0.335000        0.931034        88.178185
2 NDVIMXMX       s      0.475000        0.827586        84.152443
3 BEGAMXMN       s      0.365000        0.793103        62.045887
4 BEGAMNMN       s      0.300000        0.793103        68.704895
5 KRMTMXMN       s      0.475000        0.793103        79.163147

Competitor                  Split                       Improvement

1 KRMTMNMN              0.335000                        88.177315
2 NDVIMXMX              0.475000                        84.151741
3 KRMTMX                0.435000                        81.044876
4 KRMTMXMN              0.475000                        79.162216
5 KRMTMN                0.285000                        78.268150
```



```
Node 2 was split on variable MZSHTTRD
A case goes left if variable MZSHTTRD ≤ 31.389999
Improvement = 58.391998       C. T. = 0.269771E + 0.05

Node          Cases          Average          Standard Deviation

   2            174        23.455744                25.835199
  –1             65        39.580002                28.492435
  –2            109        13.840366                18.272223

Surrogate                   Split      Association      Improvement

1 MZSHTTDV       s     –0.780000        0.861539        38.172043
2 CERLPPDV       s     –0.085000        0.492308        26.714172
3 BELGMNDV       s     –0.855000        0.492308        39.251301
4 BEGAMNDV       s     –0.620000        0.369231        18.076315
5 KRMTMXDV       s     –1.150000        0.323077        27.505999

Competitor                  Split                       Improvement

1 BELGMNDV             –0.905000                        43.058678
2 BEGAMNDV             –1.055000                        40.379753
3 MZSHTTDV             –1.185000                        39.435616
4 KRMTMXDV             –1.905000                        37.680218
5 KRMTMNDV             –0.090000                        32.776260
```

Note: C.T. stands for Complexity Threshold (the complexity parameter used in tree pruning).

# REFERENCES

Bloch, D. A., and M. R. Segal. 1989. Empirical comparison of approaches of forming strata: Using classification trees to adjust for covariates. *Journal of the American Statistical Association* 84 (408): 897–905.

Borton, J., and J. Shoham. 1991. *Mapping vulnerability to food insecurity: Tentative guidelines for WFP country offices*. London: Relief and Development Institute.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and regression trees.* Monterey, Calif., U.S.A.: Wadsworth, Inc.

Currey, B. 1978. Mapping of areas liable to famine in Bangladesh. Ph.D. thesis (unpublished). Geography Department, University of Hawaii, Honolulu, Hawaii, U.S.A.

Cutler, B. 1986. Mapping risks of potential food crises in Ethiopia. A report to the Food Policy and Nutrition Division of the Food and Agricultural Organization of the United Nations. Addis Ababa. Photocopy.

Downing, T. E. 1990. *Assessing socioeconomic vulnerability to famine: Frameworks, concepts, and applications.* Working Paper No. 2. Arlington, Va., U.S.A.: Famine Early Warning System Project, Tulane University/Pragma Corporation.

———. 1993. Regions/vulnerable groups in FEWS methodology. Memorandum for USAID FEWS Project, Rosslyn, Va., U.S.A.

FAO (Food and Agriculture Organization of the United Nations). 1998. Guidelines for national food insecurity and vulnerability information and mapping systems (FIVIMS): Background and principles. Report to the Committee on World Food Security, Twenty-Fourth Session, Rome, 2–5 June. FAO, Rome.

FEWS (Famine Early Warning System Project). 1993. *Vulnerability assessment.* Report prepared for the United States Agency for International Development’s Bureau for Africa, Office of Analysis Research and Technical Services. Washington, D.C.: Tulane University/Pragma Corporation.

Frankenberger, T. 1992. Indicators and data collection methods for assessing household food security. In *Household food security: Concepts, indicators, and methods,* ed. S. Maxwell and T. Frankenberger. Rome: United Nations Childrens Fund/International Fund for Agricultural Development.

Frydman, H., E. I. Altman, and D.-L. Kao. 1985. Introducing recursive partitioning for financial classification: The case of financial distress. *Journal of Finance* 40 (1): 269–291.

Hutchinson, C., P. Gilruth, R. Hay, S. Marsh, and C. Lee. 1992. Geographic information systems applications in crop assessment and famine early warning. Report for the Food and Agriculture Organization of the United Nations. University of Arizona, Tucson, Ariz., U.S.A.

Kass, G. V. 1980. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics* 29 (2): 119–127.

Maddala, G. S. 1983. *Limited dependent and qualitative variables in econometrics.* Cambridge: Cambridge University Press.

Marais, M. L., J. M. Patell, and M. A. Wolfson. 1985. The experimental design of classification models: An application of recursive partitioning and bootstrapping to commercial bank loan classifications. *Journal of Accounting Research* 22 (Supplement 1984): 87–114.

Maxwell, S. 1989. *Food insecurity in North Sudan.* IDS Discussion Paper 262. Brighton, U.K.: Institute of Development Studies, University of Sussex.

Morgan, J. N., and R. C. Messenger. 1973. THAID: A sequential search program for the analysis of nominal scale dependent variables. Technical report. Institute for Social Research, University of Michigan, Ann Arbor, Mich., U.S.A.

Morgan, J. N., and J. A. Sonquist. 1963. Problems in the analysis of survey data, and a proposal. *Journal of American Statistical Association* 58: 415–434.

Riely, F. 1993. Vulnerability analysis in the FEWS project. Report to the United States Agency for International Development. Tulane University, New Orleans, La., U.S.A. Photocopy.

Seaman, J., J. Holt, and P. Allen. 1993. A new approach to vulnerability mapping for areas at risk of food crisis. Interim report on the Risk-Mapping Project. Save the Children Fund (UK), London. Photocopy.

Seyoum, S., E. Richardson, P. Webb, F. Riely, and Y. Yohannes. 1995. Analyzing and mapping food insecurity: An exploratory "CART" methodology applied to Ethiopia. Final report to the United States Agency for International Development. International Food Policy Research Institute, Washington, D.C. Photocopy.

Srinivasan, V., and Y. H. Kim. 1987. Credit granting: A comparative analysis of classification procedures. *Journal of Finance* 42 (3): 665–683.

Steinberg, D., and P. Colla. 1995. *CART: Tree-structured nonparametric data analysis*. San Diego, Calif., U.S.A.: Salford Systems.

Steinberg, D., P. Colla, and K. Martin. 1998. *CART—Classification and regression trees: Supplementary manual for Windows*. San Diego, Calif., U.S.A.: Salford Systems.

Teklu, T., J. von Braun, and E. Zaki. 1991. *Drought and famine relationships in Sudan: Policy implications*. Research Report 88. Washington, D.C.: International Food Policy Research Institute.

Webb, P., and J. von Braun. 1994. *Famine and food security in Ethiopia: Lessons for Africa*. London: John Wiley.

Webb, P., J. von Braun, and Y. Yohannes. 1992. *Famine in Ethiopia: Policy implications of coping failure at national and household levels*. Research Report 92. Washington, D.C.: International Food Policy Research Institute.

Webb, P., E. Richardson, S. Seyoum, and Y. Yohannes. 1994. Vulnerability mapping and geographical targeting: An exploratory method applied to Ethiopia. International Food Policy Research Institute, Washington, D.C. Photocopy.

Williams, M. A., H. de Silva, M. F. Koehn, and S. I. Ornstein. 1991. Why did so many savings and loans go bankrupt? *Economics Letters* (36): 61–66.

IFPRI

2033 K STREET, N.W. • WASHINGTON, D.C. 20006-1002 • U.S.A. • 1-202/862-5600
FAX: 1-202/467-4439 • E-MAIL: ifpri@cgiar.org • WEB: www.ifpri.org

IFPRI is a member of the Consultative Group on International Agricultural Research (CGIAR).