SLAC Report-261
STAN-ORION 025
UC-32
(M)

MULTIPLICATIVE MODELS IN PROJECTION PURSUIT

DAVID HAROLD HENRY*

Stanford Linear Accelerator Center
Stanford University
Stanford, California 94305

August 1983

Prepared for the department of Energy under contract number DE-AC03-76SF00515

Printed in the United States of America. Available from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, Virginia 22161. Price: Printed Copy 04; Microfiche A01.

* Ph.D Dissertation

TABLE OF CONTENTS

ABSTRACT

Friedman and Stuetzle (JASA, 1981) developed a methodology for modeling a response surface by the sum of general smooth functions of linear combinations of the predictor variables. Here multiplicative models for regression and categorical regression are explored. The construction of these models and their performance relative to additive models are examined.

CHAPTER 0
INTRODUCTION

In recent work, Jerome H. Friedman and Werner Stuetzle have developed a methodology of additive projection pursuit modelling. This dissertation examines the question for multiplicative modelling—how to accomplish it and when it is superior to additive modelling. Two general statistical problems are explored: categorical regression and classification, and regression.

Chapters 1 through 3 deal with categorical regression and classification. The first introduces the problem and briefly reviews the method of Friedman and Stuetzle. The second describes the multiplicative model and gives several examples of its application. Finally, Chapter 3 discusses four related topics: the generalization to multiple classes, use of a multiplicative model as an extension to discriminant analysis, the choice of minimization criterion, and the relative performance of the additive and multiplicative models.

Chapter 4 discusses the building of multiplicative models in regression and gives examples of their use. The appendix explains the numerical optimization techniques used by these procedures.

Routines implementing all of these procedures have been written and have been integrated into the framework designed by Friedman, Stuetzle and Roger Chaffee for additive projection pursuit.

CHAPTER ONE
CATEGORICAL REGRESSION AND PROJECTION PURSUIT

§1.1. The Categorical Regression and Classification Framework

The first situation to be considered here is that of categorical regression and classification. A training sample

$$(Y_1, \mathbf{x}_1), (Y_2, \mathbf{x}_2), \ldots, (Y_N, \mathbf{x}_N), \tag{1.1}$$

is observed, where $\mathbf{x}_n$ is a p–dimensional vector of predictor variables associated with the n^{th} observation. Y_n is a discrete variable indicating to which of K mutually exclusive classes the observation belongs (labelled 1 through K for convenience). The sample could be completely random or stratified on $\mathbf{x}$. If the marginal distribution of Y is known, it could instead be stratified on Y.

Categorical regression seeks to estimate the probability of the response Y falling into each class conditional on the value of $\mathbf{x}$:

$$p_k(\mathbf{x}) = \Pr\{Y = k \mid \mathbf{x}\} \qquad 1 \leq k \leq K. \tag{1.2}$$

For example, in a business application, class 1 might represent those loan applicants who would default if granted a loan, while class 2 denoted those who would repay it in full. The vector $\mathbf{x}$ might include income, job stability and other personal factors that could affect repayment. The function $\hat{p}_k(\mathbf{x})$ would indicate the probability of default given salary and other characteristics.

Many applications require a decision rule that will identify the response class Y based on the predictors $\mathbf{x}$. In the example such a rule would divide loan applications into "good credit risks" and "bad", hopefully protecting the bank from unwise loans and loss of money. Since any decision rule would not be completely accurate, classification errors would result in various losses. Labelling a good risk as bad deprives the bank of a profitable loan opportunity. Identifying a bad prospect as good may cost the sum loaned. A good decision rule seeks to minimize the probability and magnitude of such a loss. Let $L(i \mid k)$ denote the loss incurred by classifying an observation of class k as class i. Then the risk associated with the assignment of an observation with predictor $\mathbf{x}$ as class i is

$$R(i \mid \mathbf{x}) = \sum_{k=1}^{K} L(i \mid k) p_k(\mathbf{x}). \tag{1.3}$$

Were the conditional probabilities $p_k(\mathbf{x})$ known, the risk could be minimized by the Bayes' rule, which chooses that value i that minimizes (1.3). Since they are not known, the empirical Bayes' rule replaces them with the estimates $\hat{p}_k$ obtained in the categorical regression. The rule becomes: choose that value i which minimizes

$$\hat{R}(i \mid \mathbf{x}) = \sum_{k=1}^{K} L(i \mid k)\, \hat{p}_k(\mathbf{x}). \tag{1.4}$$

§1.2. Methods of Categorical Regression and Classification

Various methods have been used to estimate the conditional probabilities. Most common are linear discriminant analysis, quadratic discriminant analysis and logistic regression.

Linear discriminant analysis assumes that the distribution of $\mathbf{x}$ given Y is multivariate normal and that the covariance structure is the same for each class. The class means are estimated by the class sample mean $\mathbf{x}_k$ and the common covariance matrix by the pooled sample covariance $\hat{S}$. The conditional probabilities are taken to be

$$\hat{p}_k(\mathbf{x}) = \frac{\pi_k e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_k)'\hat{S}^{-1}(\mathbf{x}-\mathbf{x}_k)}}{\sum_{i=1}^{K} \pi_i e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_i)'\hat{S}^{-1}(\mathbf{x}-\mathbf{x}_i)}}, \tag{1.5}$$

where π_i is the marginal probability of class i. The estimates can depend heavily on the assumptions of normality and equal covariances. Deviations from these can distort the estimates.

Quadratic discriminant analysis also assumes multivariate normality, but allows the covariance structure to vary from class to class. The means and covariances for each class are estimated by the corresponding class sample means and covariances, and the probability estimate is

$$\hat{p}_k(\mathbf{x}) = \frac{\pi_k |\hat{S}_k|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_k)' \hat{S}_k^{-1} (\mathbf{x}-\mathbf{x}_k)}}{\sum_{i=1}^{K} \pi_i |\hat{S}_i|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_i)' \hat{S}_k^{-1} (\mathbf{x}-\mathbf{x}_i)}}. \tag{1.6}$$

Since individual class covariance matrics are estimated, many more observations are necessary. While freed from the assumption of equal covariances, deviations from normality can still greatly affect the perceived probabilities. Also, since frequently at least one predictor is binary, the assumption of normality is usually incorrect.

In the two class case, an attempt to generalize linear discriminant analysis brings about logistic regression. Rather than concentrating on estimating properties of the distribution of **x** given the class Y, logistic regression conditions on the observed combined sample predictor distribution and models the conditional probability more directly:

$$p_1(\mathbf{x}) = \frac{e^{b+\mathbf{a}'\mathbf{x}}}{1 + e^{b+\mathbf{a}'\mathbf{x}}}. \tag{1.7}$$

The maximum likelihood estimates of b and **a** are obtained numerically.

Logistic regression also makes several important assumptions. First, it assumes that $p_1(\mathbf{x})$ depends only on one linear combination of the predictors, $\mathbf{a}'\mathbf{x}$. All relevant information is assumed to be in that one projection. No other linear combination adds any further information. Second, it assumes that the dependency is through a member of one particular parametric family. If the true relationship is not logistic, the procedure may fail. For example, this could happen if the distributions of **x** given Y are normal with equal means and different variances.

§1.3. Projection Pursuit Categorical Regression

When samples are small, assumptions like the previous ones may be necessary. However, when larger data sets are available, it is desirable to seek methods that impose fewer restrictions and allow the data to reveal the structure.

To generalize (1.7), Friedman and Stuetzle (1980) replace the logistic curve by an unknown smooth function:

$$p_1(\mathbf{x}) = f(\mathbf{a}'\mathbf{x}). \tag{1.8}$$

Here both $\mathbf{a}$ and f are unknown, with the only restriction being the smoothness of f. That choice of $\mathbf{a}$ and f is sought that will minimize the distance between the observed data and the fit:

$$S(\mathbf{a}) = \sum_{n=1}^{N} v_n(Y_n - f(\mathbf{a}'\mathbf{x}_n))^2. \tag{1.9}$$

Here the $\{v_n\}$ are weights that take into account the disparity between the sample and population distributions:

$$v_n = \frac{\{prior\ probability\ of\ class\ Y_n\}}{\{number\ in\ class\ Y_n\}}. \tag{1.10}$$

If the priors are not specified, they are assumed to be those of the sample. In that case v_n is set equal to $\frac{1}{N}$.)

To obtain estimates of $\mathbf{a}$ and f, suppose first that the linear combination $\mathbf{a}$ has been chosen and the smooth function f for that combination is desired. Then Y can be compared to the single variable $\mathbf{a}'\mathbf{x}$. The estimate of f is now obtained by applying a weighted local linear smoother to the scatterplot of $(Y_n, \mathbf{a}'\mathbf{x}_n)_{n=1}^{N}$. At each abscissa value $\mathbf{a}'\mathbf{x}_n$, consider a window containing that observation and the h values on each side of it. Weighting each of these $2h+1$ observations by its weight v_i and ignoring all the other points, calculate the weighted regression line and define the function value to be the fitted value of the regression line at the point. Repeat the procedure for each value $\mathbf{a}'\mathbf{x}_n$, using only the observation and its h neighbors on each side. For observations near an end, where there are not h available on one side, use an asymmetrical window with h on one side and as many as are available on the other. In this way a smooth function is fit for the range of observations for this projection.

Such a smoother provides an estimate of f for this particular combination $\mathbf{a}$. The criterion (1.9) can be used as a measure of the fit for this combination $(\mathbf{a}, f)$. The optimal linear combination $\mathbf{a}$ can then be selected by numerical optimization over all possible combinations. (The numerical method used is discussed in the appendix.) The name projection pursuit is derived from this procedure of searching for (pursuing) the optimal linear combination (projection).

This approach eliminates the assumption of a specific parametric family of probability curves. It still retains the restriction that the relationship depends only on one linear combination $\mathbf{a}$. When larger data sets are available, better modelling of the true underlying probability surface can be obtained by eliminating this. In analogy with a similar method in projection pursuit regression, Friedman and Stuetzle (1980) suggest repeating the above procedure on the residuals $Y_n - f(\mathbf{a}'\mathbf{x}_n)$ to obtain further modifying projections until no appreciable decrease in squared distance is observed. In this manner a model

$$p_1(\mathbf{x}) = \sum_{m=1}^{M} f_m(\mathbf{a}_m'\mathbf{x}) \tag{1.10}$$

is constructed recursively. One problem that arises, however, is that $\hat{p}_k(\mathbf{x})$ need not lie between zero and one. When this occurs, $\hat{p}_k(\mathbf{x})$ is set to zero or one.

While natural in projection pursuit regression procedures, such an additive model does not seem to be the most natural extension in the case of categorical regression. The next chapter discusses an alternative approach using a multiplicative model.

CHAPTER TWO
THE MULTIPLICATIVE MODEL FOR CATEGORICAL REGRESSION

§2.1. Description of the Model

Since probabilities must lie between zero and one, the most natural way of modelling them is through the odds ratio. This approach models the odds ratio as the product of smooth functions

$$\frac{p_1(\mathbf{x})}{p_0(\mathbf{x})} = \prod_{m=1}^{M} h_m(\mathbf{a}'\mathbf{x}). \tag{2.1}$$

The unknowns $\{\mathbf{a}_m\}_{m=1}^M$ and $\{h_m\}_{m=1}^M$ must be estimated. Algebraic manipulation shows that this is equivalent to

$$p_1(\mathbf{x}) = \frac{\prod_{m=1}^M h_m(\mathbf{a}'\mathbf{x})}{1 + \prod_{m=1}^M h_m(\mathbf{a}'\mathbf{x})}. \tag{2.2}$$

Two approaches to estimating the parameters are now possible. Both are stepwise procedures. In the first, at step M, $\{\mathbf{a}_m\}_{m=1}^{M-1}$ and $\{h_m\}_{m=1}^{M-1}$ have previously been selected. Wanted are $\mathbf{a}_M$ and h_M so as to minimize

$$\begin{aligned} S_M(\mathbf{a}_M, h_m) &= \sum_{n=1}^N v_n(Y_n - \hat{p}_1(\mathbf{x}_n))^2 & (2.3) \\ &= \sum_{n=1}^N v_n\left(Y_n - \frac{\prod_{m=1}^M h_m(\mathbf{a}'\mathbf{x}_n)}{1 + \prod_{m=1}^M h_m(\mathbf{a}'\mathbf{x}_n)}\right)^2 & (2.4) \end{aligned}$$

subject to the previously chosen $\mathbf{a}_m$'s and h_m's.

Suppose first that the projection $\mathbf{a}_M$ has been chosen, and that the function h_M is sought. For brevity, let $g_n = \prod_{m=1}^{M-1} h_m(\mathbf{a}'_m\mathbf{x}_n)$ (the previous step's model for the odds ratio at $\mathbf{x}_n$). Then (2.4) can be reexpressed as

$$\sum_{n=1}^N v_n\left(\frac{g_n}{1 + g_n h_M(\mathbf{a}'_M\mathbf{x}_n)}\right)^2 \left(\frac{Y_n(1 + g_n h_M(\mathbf{a}'_M\mathbf{x}_n))}{g_n} - h_M(\mathbf{a}'_M\mathbf{x}_n)\right)^2. \tag{2.5}$$

This reduces the problem to the estimation of a smooth function h_M so as to minimize its weighted distance from a set of points $\{Z_n\}$:

$$\sum_{n=1}^{N} w_n (Z_n - h_M(\mathbf{a}'_M \mathbf{x}_n))^2 . \tag{2.6}$$

If the unknown function appeared only in the rightmost term and not in the weights, it could be estimated by applying a weighted local linear smoother to the points $\{Z_n\}$. In the present situation, however, both the weights w_n and the $\{Z_n\}$ depend on the unknown h_M. Hence the function can not be immediately estimated. Instead an iterative procedure with successive reweighting is necessary. An initial function $h_M^{(0)}(z) \equiv 1$ is chosen and used to define the weights w_n. These are then used to compute a weighted smooth $h_M^{(1)}$. New weights are calculated using $h_M^{(1)}$, and a new estimate $h_M^{(2)}$ is obtained. This continues until a convergence criterion is satisfied, such as

$$\sup_s \mid h_M^{(j+1)}(s) - h_M^{(j)}(s) \mid \leq T. \tag{2.7}$$

In this way an estimate of h_M is obtained for the given linear combination $\mathbf{a}_M$. A criterion of fit for this choice of $\mathbf{a}_M$ can then be calculated:

$$S_M(\mathbf{a}_M) = \sum_{n=1}^{N} v_n \left(Y_n - \frac{g_n h_M(\mathbf{a}'_M \mathbf{x}_n)}{1 + g_n h_M(\mathbf{a}'_M \mathbf{x}_n)} \right)^2 , \tag{2.8}$$

and the best choice of $\mathbf{a}_M$ can be determined by numerical optimization.

This approach can be quite expensive computationally. While the convergence is usually quite fast, it need not be. Also, the use of iteration within a numerical optimization (which also uses iteration) greatly increases the computer time required compared with a noniterative smooth. Hence an alternate noniterative one is preferred. Using an approximation, this other method comes close to the exact iterative method with a considerable reduction in computation expense.

This second method reparametrizes the model. For each $m \leq M$, define the functions f_{1m} and f_{0m} such that

$$f_{1m}(z) = h_m(z) f_{0m}(z) \tag{2.9}$$

and

$$E\left(\prod_{m=1}^{m'} f_{1m}(\mathbf{a}'_m\mathbf{x}) + \prod_{m=1}^{m'} f_{0m}(\mathbf{a}'_m\mathbf{x})\right) = 1 \qquad 1 \leq m' \leq M. \tag{2.10}$$

Now

$$p_1(\mathbf{x}) = \frac{\Pi_{m=1}^{M} f_{1m}(\mathbf{a}'_m\mathbf{x})}{\Pi_{m=1}^{M} f_{1m}(\mathbf{a}'_m\mathbf{x}) + \Pi_{m=1}^{M} f_{0m}(\mathbf{a}'_m\mathbf{x})}. \tag{2.11}$$

This procedure is also stepwise. At the M^{th} step, the algorithm has already estimated $\{\mathbf{a}_m\}_{m=1}^{M-1}$ and $\{f_{km}\}_{m=1,M-1}^{k=0,1}$. Being sought are $\mathbf{a}_M$, f_{0M} and f_{1M} so as to minimize

$$S_M = \sum_{n=1}^{N} v_n \left(Y_n - \frac{\Pi_{m=1}^{M} f_{1m}(\mathbf{a}'_m\mathbf{x}_n)}{\Pi_{m=1}^{M} f_{1m}(\mathbf{a}'_m\mathbf{x}_n) + \Pi_{m=1}^{M} f_{2m}(\mathbf{a}'_m\mathbf{x}_n)}\right)^2. \tag{2.12}$$

For brevity, designate $t_n = \Pi_{m=1}^{M} f_{1m}(\mathbf{a}'_m\mathbf{x}_n) + \Pi_{m=1}^{M} f_{0m}(\mathbf{a}'_m\mathbf{x}_n)$ and $s_n = \Pi_{m=1}^{M-1} f_{1m}(\mathbf{a}'_m\mathbf{x}_n)$. Then

$$S_M = \sum_{n=1}^{N} v_n \left(\frac{s_n}{t_n}\right)^2 \left(\frac{Y_n t_n}{s_n} - f_{1M}(\mathbf{a}'_m\mathbf{x}_n)\right)^2. \tag{2.13}$$

Again, the weights $v_n\,(\frac{s_n}{t_n})^2$ depend on the unknown f_{1M} and f_{2M}. To avoid having to iterate, the following approximation will be used. Since the constraint (2.10) states the $E(t_n) = 1$, t_n shall be approximated by 1 in the weights, leaving

$$S_m \approx \sum_{n=1}^{N} s_n^2 v_n \left(\frac{Y_n}{s_n} - f_{1M}(\mathbf{a}'_M\mathbf{x}_n)\right)^2. \tag{2.14}$$

A noniterative local linear smooth of $\frac{Y_n}{s_n}$ with weights $v_n s_n^2$ gives the estimate of f_{1M}. An estimate of f_{0M} can be obtained from the constraint (2.10). For simplicity and to offset any bias introduced by the approximation, a procedure similar to that for f_{1M} will be used. Since S_M can also be expressed as

$$S_M = \sum_{n=1}^{N} v_n \left((1 - Y_n) - \frac{\Pi_{m=1}^{M} f_{0M}(\mathbf{a}'_M\mathbf{x}_n)}{t_n}\right)^2, \tag{2.15}$$

the same procedure can be used as was used to determine f_{1M}. A local linear smooth of $\frac{1-Y_N}{\prod_{m=1}^{M-1} f_{0m}(\mathbf{a}'_m\mathbf{x}_n)}$ with weights $v_n(\prod_{m=1}^{M-1} f_{0m}(\mathbf{a}'_m\mathbf{x}_n))^2$ gives the estimate of f_{0M}. Again, the optimal projection $\mathbf{a}_M$ is found through numerical optimization.

The method just described can be viewed in another way. By reparametrizing and approximating, it becomes equivalent to the following procedure:

1.) Model both $p_1(\mathbf{x})$ and $p_0(\mathbf{x})$ simultaneously as products of smooth functions of linear combinations

$$\tilde{p}_1(\mathbf{x}) = \prod_{m=1}^{M} f_{1m}(\mathbf{a}'_m\mathbf{x}) \tag{2.16}$$

$$\tilde{p}_0(\mathbf{x}) = \prod_{m=1}^{M} f_{0m}(\mathbf{a}'_m\mathbf{x}), \tag{2.17}$$

with the same $\mathbf{a}_m$ at each step in both models and ignoring the dependence of the two $(p_1(\mathbf{x}) + p_0(\mathbf{x}) = 1)$.

2.) In the resulting model, $\tilde{p}_1$ and $\tilde{p}_0$ will not sum identically to one over the full range of $\mathbf{x}$. Hence they are not probabilities, although they do conform quite closely to p_1 and p_0. Therefore they are normalized to obtain probability estimates

$$\hat{p}_1(\mathbf{x}) = \frac{\tilde{p}_1(\mathbf{x})}{\tilde{p}_1(\mathbf{x}) + \tilde{p}_0(\mathbf{x})} \tag{2.18}$$

$$\hat{p}_0(\mathbf{x}) = \frac{\tilde{p}_0(\mathbf{x})}{\tilde{p}_1(\mathbf{x}) + \tilde{p}_0(\mathbf{x})}. \tag{2.19}$$

From a practical point of view, this method gives results quite close to the iterative method. Because it is so much faster, it is the one adopted. It also has the advantage that the functions for the first projection (f_{11} and f_{01}) agree exactly with those obtained from the one projection procedure of Friedman and Stuetzle described in chapter 1.

§2.2 Numerical Examples

To see how the procedure operates, several examples are now given. The first consists of 500 four-dimensional observations. Half of these were generated from a multivariate Gaussian distribution with mean $(0,0,0,0)$ and identity covariance matrix (group 1). The group 0 observations were generated from a Gaussian with mean $(1.5,0,0,0)$ and covariance matrix

$$\Sigma = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}.$$

There is a location shift in the first variable and a dispersion difference in the remaining three. The data was analyzed by various methods, with the results summarized in Table 2.1. Since the observations are Gaussian, it is not surprising that quadratic discriminant analysis does the best. It misclassifies only 13.2% of the training sample, and obtains a squared distance of .096. The Bayes' rule values (where the parameters are known and not estimated) are 14.3% and .101. Linear discriminant analysis detects the location shift, but it misses the dispersion difference. It misclassifies 22.4% of the training sample, with a criterion of .152. This is identical with the results of logistic regression, which also misclassifies 22.4%, with criterion .152. Its best linear combination is $(1.59,-0.4,.03,.115)$. The first (location shift) variable dominates the linear combination.

A multiplicative projection pursuit model was applied with a smoother using a window containing 30% of the observations. At the first step it finds the direction $(.999,-.03,0,.04)$, with the criterion .1524. It now seeks to simplify the model by removing unnecessary components in the direction vector. Each nonzero component is individually set to zero, and the criterion with that component deleted is calculated. The component that causes the criterion to increase the least is temporarily removed from the model. The best direction only involving the remaining components is obtained numerically, and its criterion calculated. If the difference between this value and that obtained using all components is below a threshold, the component is permanently eliminated. This continues until no component can be eliminated without increasing the criterion beyond the threshold. Here the direction is reduced to $(1.0, 0, 0, 0)$, only increasing the criterion to .1526. The optimal function for the first class is plotted in Figure 2.1. The algorithm now examines the reduction in criterion from the initial variance to the first step. Since it is larger than the user-defined threshold of 5% of the initial variance, the projection is accepted and the procedure continues.

Having captured the effect of the shift variable in the first projection, it now deals with the dispersion variables. As the best second projection it chooses $(.04, .97, .12, .21)$. As described above, it now eliminates the first and third variables to leave $(0, .98, 0, .21)$. The function is plotted in Figure 2.2. From this it can be seen that observations in the tails (larger dispersion) are inclined to belong to group 0, since the value of f_{12} is close to zero there. Again the reduction in criterion is sufficient to allow the procedure to continue.

The third and fourth steps reduce the criterion to .105. At the fifth step the reduction is below 5%. The algorithm then makes several attempts to find a larger reduction by varying the starting location in its numerical search for the best projection. Since still no sufficiently large reduction is obtained, the procedure rejects the fifth step and terminates with a four projection model.

As Table 2.1 indicates, multiplicative projection pursuit's final model has criterion that is not far from the optimal parametric procedure. Its apparent risk also compares well. The slightly larger values are the result of having to estimate the functional forms for each projection, rather than simply estimating parameters for a specified parametric family. Since the smoother locally fits weighted least squares lines to obtain the function, there is some linearization of the true underlying exponential. Varying the proportion of observations utilized by the smoother will vary the degree of linearization. Had a smaller proportion been used, the linear effect would have been lessened. However, the resulting functions would have been less smooth, with increased variance. These two factors must be balanced in setting the proportion to be used.

To validate the previous estimates of misclassification risk and criterion, five thousand new observations from each distribution were classified using the models obtained above. The results, also in Table 2.1, support the previous results. No overfitting is evident here. Multiplicative projection pursuit retains its position relative to the other methods.

Also of interest is the relative performance of the multiplicative and additive models. The results with one projection are the same, since the procedures are identical at the first step. At the second and succeeding steps, the additive model falls behind, finishing with a squared distance approximately 10% higher than that of the multiplicative model. (The sampling variation due to the generation of the normal sample appears to be small. To examine its effect, ten additional training and validation samples were generated, and the performances of the models compared. The average validated difference in criterion between the multiplicative and additive models was −.0061, with standard deviation .0060. The difference in misclassification rates was −1.4%. (standard deviation .36%). Comparing the multiplicative model with quadratic discriminant analysis, the difference in criterion was .014 (standard deviation .005). For misclassification rates, the difference was 1.6% (standard deviation 1.1%). So the performance of the multiplicative model relative to the optimal procedure (quadratic discriminant analysis) appears to be real, and not a result of sampling variation.)

The second example is data from Delury (1973, see Press and Wilson, 1978, for a logistic regression analysis). The 1970 census data are used to attempt to classify each of the fifty U.S. states according to its population growth between 1960 and 1970. States whose growth was below the median change among all states are classified as 0, with the others classified as 1. The predictor variables are per capita income (in thousands of dollars), birth rate (percent), death rate (percent), urbanization (1 if more than 70% of the population was urban), and presence of ocean coastline (1 if present).

Discriminant and logistic results are summarized in Tables 2.2 and 2.3. Birth rates were higher and death rates lower among the high-growth states. Higher income, coastline and a less urban environment were also associated with those states.

The data was also analyzed using multiplicative projection pursuit. Because of the smaller size of the training sample here, a smoother using 40% of the observations was employed. Also, the number of projections was limited to two. (The small sample size wuld allow a large degree of overfitting otherwise.) To enhance the effectiveness of the numerical optimizer, the predictor variables were standardized so that each variable had zero median and unit interquartile range. (While the procedure is equivariant under location and scale changes, the numerical optimizer behaves best when the relative scales of the variables do not vary greatly.)

At the first step, the algorithm selects the projection $(-.45, 0, .84, 0, -.30)$: concentrating heavily on the death rate with negative coefficients for income and coastline. The corresponding function for the higher growth group is shown in Figure 2.5. For low values of its argument (corresponding to high income, coastline, low death rate) the function takes large values, indicating higher growth states. It descends sharply as the argument increases, with an unusual rise at the end. (This end effect is caused by two outliers. After the minimum of the function, there are only two observations. These influential observations are isolated near the value 2, one from each group. Because they are separated from the other points, they have large leverage, pulling the function up toward .5. Such end effects are common to techniques using smoothing.)

The second projection concentrates on income along with coastline and a small birth rate coefficient $(.86, .14, 0, 0, .49)$. Increasing values of these variables tend to increase the probability of large growth, as seen in Figure 2.6. End effects are apparent after .5.

As Table 2.4 indicates, multiplicative projection pursuit with two projections performed better on the training sample than linear discriminant analysis or logistic regression, but slightly worse than quadratic discriminant analysis. However, this was based on the training sample. Since projection pursuit and quadratic discriminant analysis estimate more parameters, they would be expected to appear to perform better. To better evaluate the performance of the methods, the data was randomly divided into five groups (Table 2.5). Each group was then classified according to the model generated by the other four groups combined. The results are in Table 2.4. The criterion increases for all methods, with that of quadratic discriminant analysis increasing most markedly. The multiplicative projection pursuit model obtains the lowest values for both the criterion and misclassification error. Of particular interest here is the cross-validated risk for projection pursuit with one projection. Because the larger window of observations (40%) was used to compensate for the relative sparseness of the data, it could not perform as well as the logistic, which it generalizes. The ability to add a second direction, however, allowed this to be overcome and a much better fit to be obtained. This is an example where, even for an only moderately sized data set, multiplicative projection pursuit is able to model the underlying structure more effectively.

Table 2.1. Criterion and Misclassification Rates for Training and Validation Samples Example 1 (Gaussian)

	Criterion		Misclassification Rate (%)	
Method	*Training*	*Validation*	*Training*	*Validation*
Linear Discriminant Analysis	.152	.156	22.4	22.9
Logistic Regression	.152	.156	22.4	22.9
Additive Projection Pursuit				
One Projection	.152	.155	22.2	22.7
Two Projections	.133	.140	19.8	20.6
Three Projections	.118	.128	18.0	18.8
Four Projections	.110	.123	17.4	17.9
Multiplicative Projection Pursuit				
One Projection	.152	.155	22.2	22.7
Two Projections	.130	.136	19.0	20.0
Three Projections	.120	.122	17.8	17.7
Four Projections	.105	.110	14.8	15.5
Quadratic Discriminant Analyis	.096	.100	13.2	14.1

Table 2.2 Discriminant and Logistic Coefficients Example 2 (Population Growth)

Procedure	*Constant*	*Income*	*Births*	*Coast*	*Urban*	*Deaths*
Logistic Regression	−13.45	3.04	4.92	1.63	−1.03	−7.87
Linear Discriminant	−7.62	1.86	3.06	1.33	−0.26	−5.96

Table 2.3. Quadratic Discriminant Estimates
Example 2 (Population Growth)

Group	*Income*	*Births*	*Coast*	*Urban*	*Deaths*
Low Growth Mean	3.52	1.83	.32	.28	.99
High Growth Mean	3.95	1.92	.64	.56	.88

Covariance Matrices

.297	−.048	.029	.194	.009		.268	−.056	.050	.129	−.029
−.048	.025	.003	−.025	−.008		−.056	.092	−.009	−.020	.006
.029	.003	.227	.073	.008		.050	−.009	.240	.017	.006
.194	−.025	.073	.210	.012		.129	−.020	.017	.257	−.032
.009	−.008	.008	.012	.011		−.029	.006	.006	−.032	.021
		Low Growth						High Growth		

Table 2.4 Criterion and Misclassification Rates
for Training Sample and Crossvalidation
Example 2 (Population Growth)

	Criterion		Misclassification Rate (%)	
Method	*Training*	*Crossval.*	*Training*	*Crossval.*
Linear Discriminant Analysis	.153	.210	28	34
Logistic Regression	.148	.211	20	28
Additive Projection Pursuit				
One Projection	.148	.238	16	36
Two Projections	.122	.166	16	22
Multiplicative Projection Pursuit				
One Projection	.148	.238	16	36
Two Projections	.112	.151	16	16
Quadratic Discriminant Analysis	.120	.186	14	26

Table 2.5 Raw Data for Example 2

State	*Class*	*Income*	*Births*	*Coast*	*Urban*	*Deaths*
		Set 1				
Arkansas	0	2.878	1.8	0	0	1.1
Colorado	1	3.855	1.9	0	1	0.8
Delaware	1	4.524	1.9	1	1	0.9
Georgia	1	3.354	2.1	1	0	0.9
Idaho	0	3.290	1.9	0	0	0.8
Iowa	0	3.751	1.7	0	0	1.0
Mississippi	0	2.626	2.2	1	0	1.0
New Jersey	1	4.701	1.6	1	1	0.9
Vermont	1	3.468	1.8	0	0	1.0
Washington	1	4.053	1.8	1	1	0.9
		Set 2				
Kentucky	0	3.112	1.9	0	0	1.0
Louisiana	1	3.090	2.7	1	0	1.3
Minnesota	1	3.859	1.8	0	0	0.9
New Hampshire	1	3.737	1.7	1	0	1.0
North Dakota	0	3.086	1.9	0	0	0.9
Ohio	0	4.020	1.9	0	1	1.0
Oklahoma	0	3.387	1.7	0	0	1.0
Rhode Island	0	3.959	1.7	1	1	1.0
South Carolina	0	2.990	2.0	1	0	0.9
West Virginia	0	3.061	1.7	0	0	1.2
		Set 3				
Connecticut	1	4.917	1.6	1	1	0.8
Maine	0	3.302	1.8	1	0	1.1
Maryland	1	4.309	1.5	1	1	0.8
Massachusetts	0	4.340	1.7	1	1	1.0
Michigan	1	4.180	1.9	0	1	0.9
Missouri	0	3.781	1.8	0	1	1.1
Oregon	1	3.719	1.7	1	0	0.9
Pennsylvania	0	3.971	1.6	1	1	1.1
Texas	1	3.606	2.0	1	1	0.8
Utah	1	3.227	2.6	0	1	0.7

Table 2.5 (cont.)

State	*Class*	*Income*	*Births*	*Coast*	*Urban*	*Deaths*
		Set 4				
Alabama	0	2.948	2.0	1	0	1.0
Alaska	1	4.644	2.5	1	0	1.0
Arizona	1	3.665	2.1	0	1	0.9
California	1	4.493	1.8	1	1	0.8
Florida	1	3.738	1.7	1	1	1.1
Nevada	1	4.563	1.8	0	1	0.8
New York	0	4.712	1.7	1	1	1.0
South Dakota	0	3.123	1.7	0	0	0.9
Wisconsin	1	3.812	1.7	0	0	0.9
Wyoming	0	3.815	1.9	0	0	0.9
		Set 5				
Hawaii	1	4.623	2.2	1	1	0.5
Illinois	0	4.507	1.8	0	1	1.0
Indiana	1	3.772	1.9	0	0	0.9
Kansas	0	3.853	1.6	0	0	1.0
Montana	0	3.500	1.8	0	0	0.9
Nebraska	0	3.789	1.8	0	0	1.1
New Mexico	0	3.077	2.2	0	0	0.7
North Carolina	1	3.252	1.9	1	0	0.9
Tennessee	0	3.119	1.9	0	0	1.0
Virginia	1	3.712	1.8	1	0	0.8

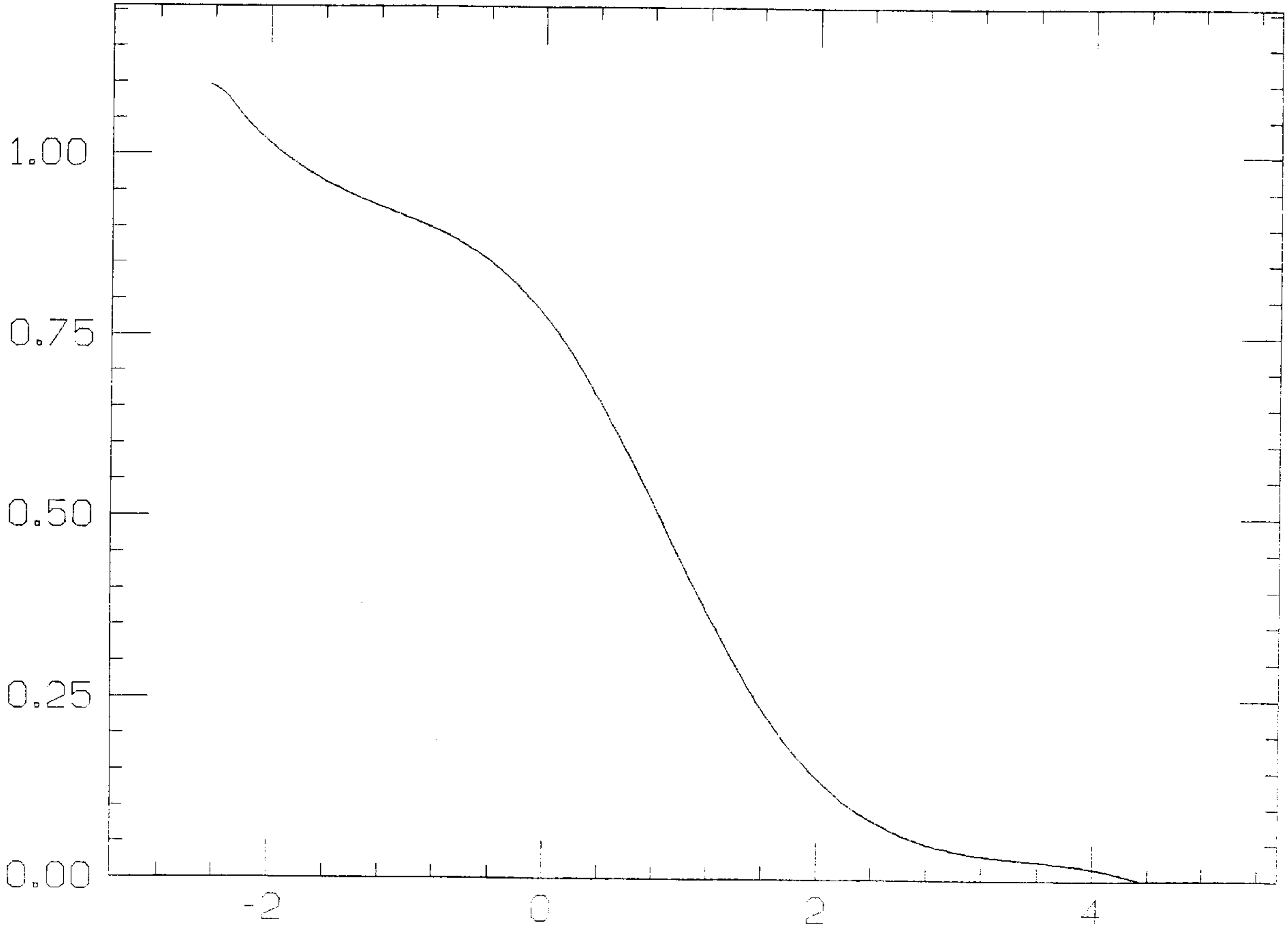

FIGURE 2.1. ØPTIMAL FUNCTIØN FØR GRØUP ØNE
FIRST PRØJECTIØN. EXAMPLE 1
PRØJECTIØN = X1

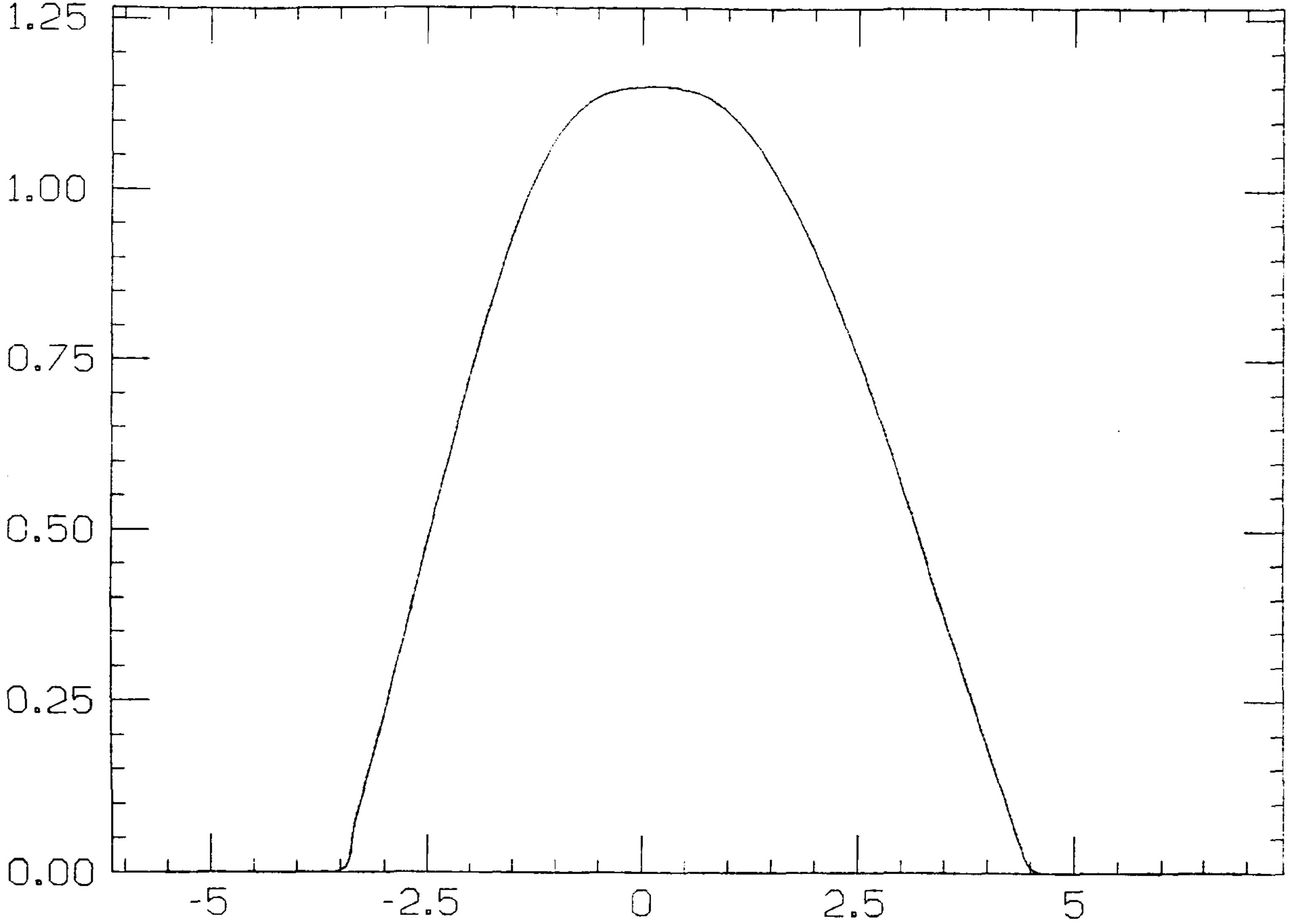

FIGURE 2.2. ØPTIMAL FUNCTIØN FØR GRØUP ØNE
SECØND PRØJECTIØN, EXAMPLE 1
PRØJECTIØN = .98 * X1 + .21 * X4

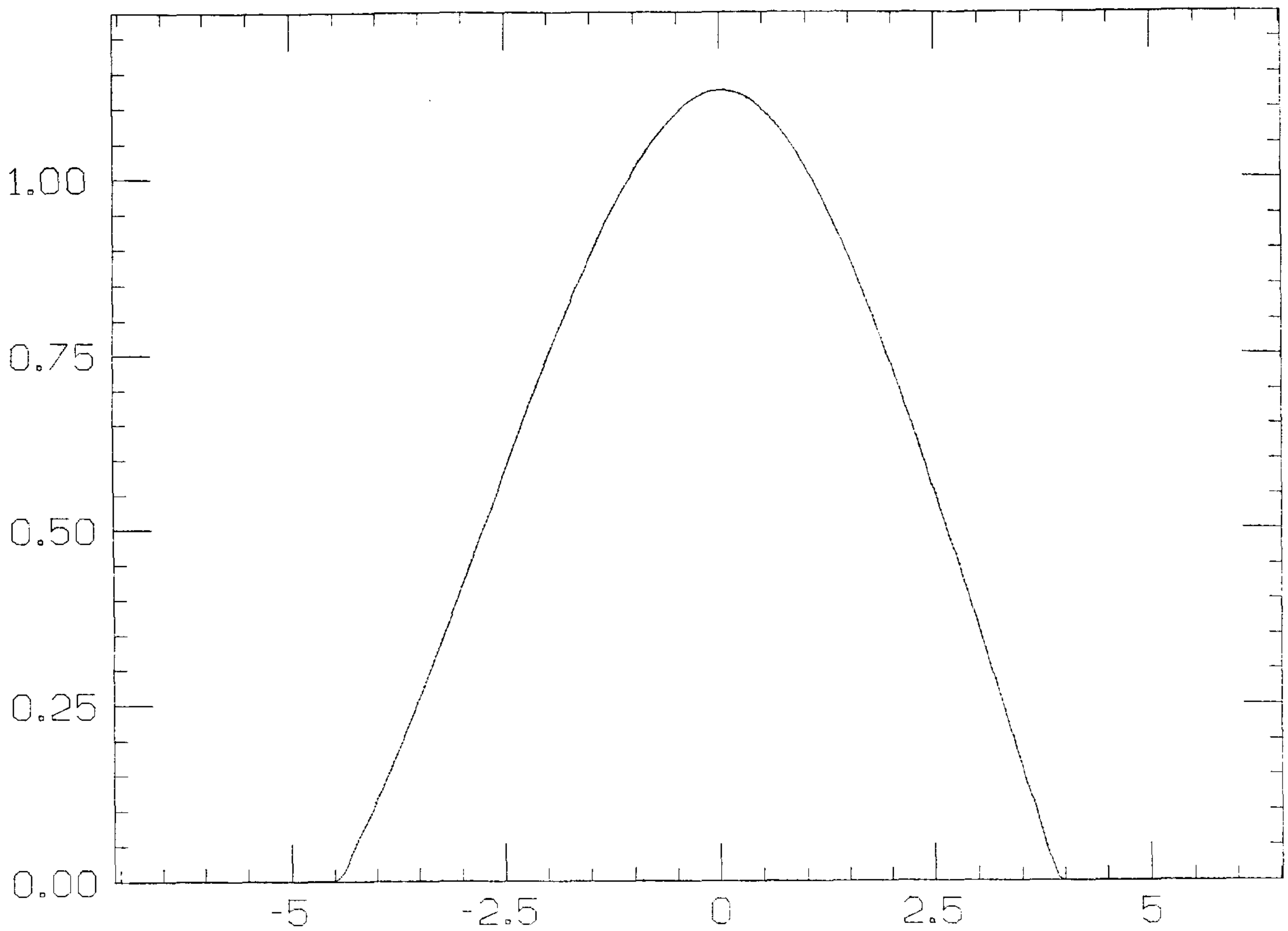

FIGURE 2.3. ØPTIMAL FUNCTIØN FØR GRØUP ØNE
THIRD PRØJECTIØN, EXAMPLE 1
PRØJECTIØN = X4

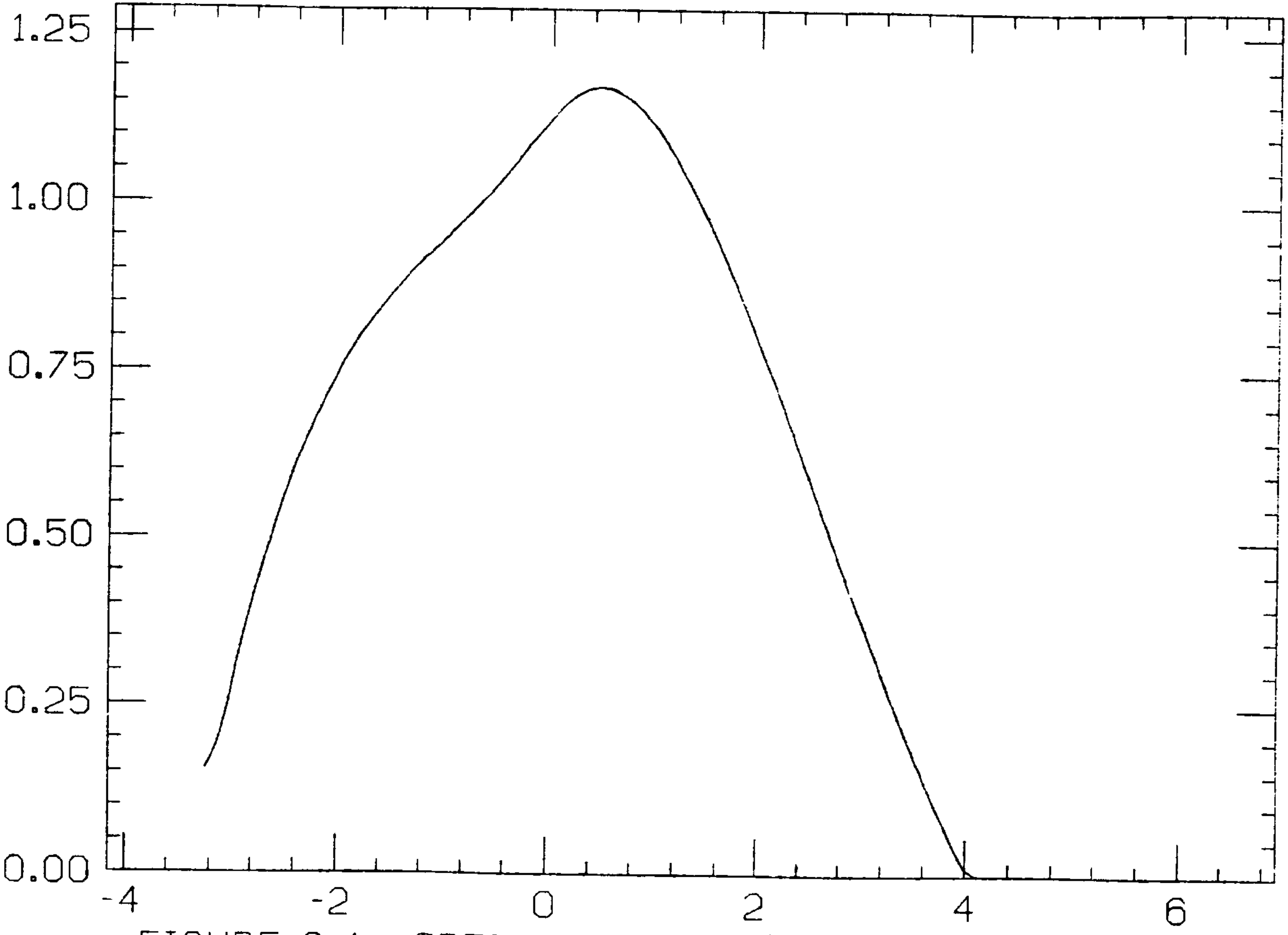

FIGURE 2.4. ØPTIMAL FUNCTIØN FØR GRØUP ØNE
FØURTH PRØJECTIØN, EXAMPLE 1
PRØJECTIØN = .63 * X1 + .78 * X3

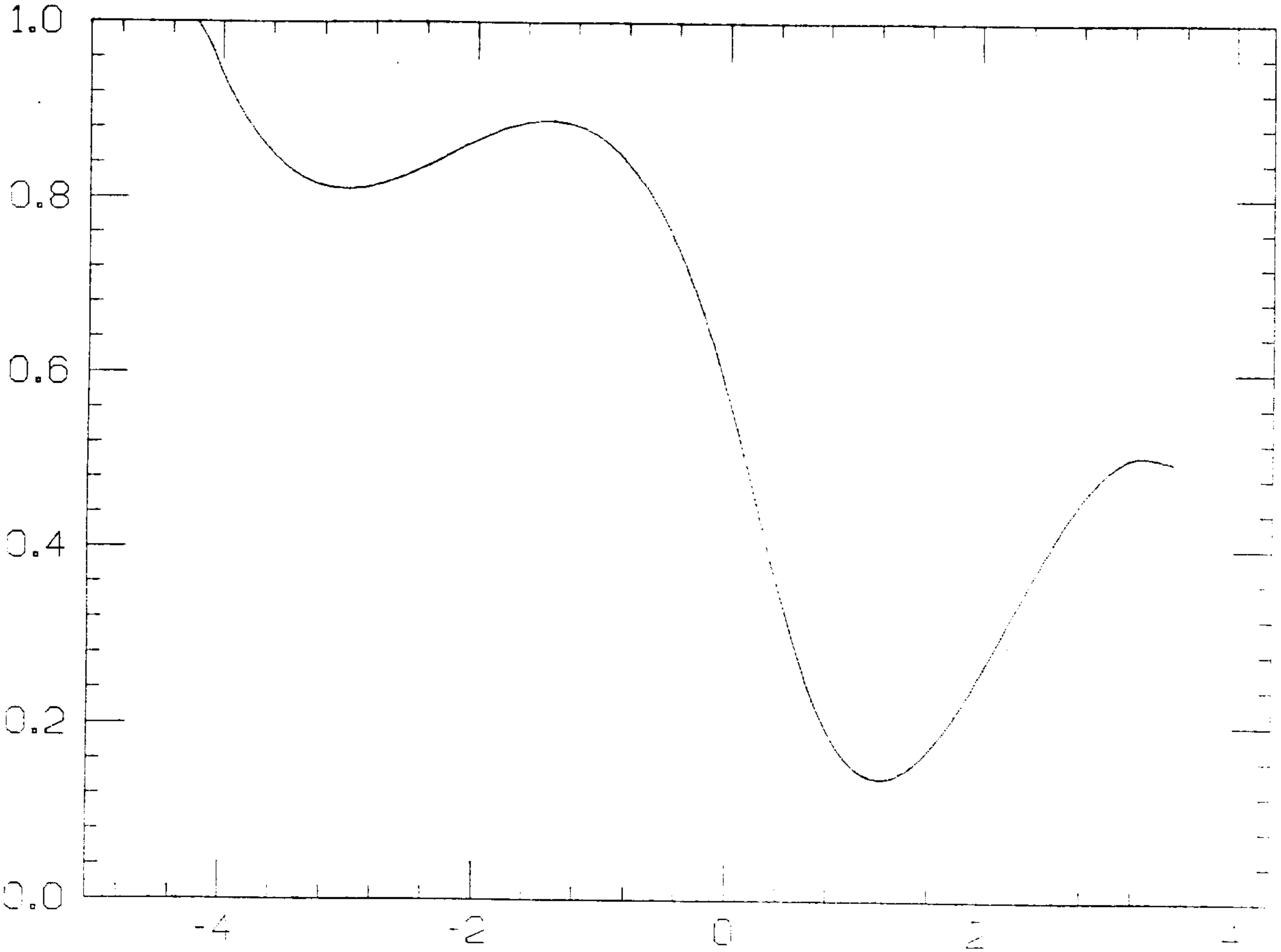

FIGURE 2.5. ØPTIMAL FUNCTIØN FØR HIGH GRØWTH GRØUP
FIRST PRØJECTIØN, EXAMPLE 2
PRØJECTIØN = -.45 * INCØME + .84 * DEATHRATE - .30 * CØASTLINE

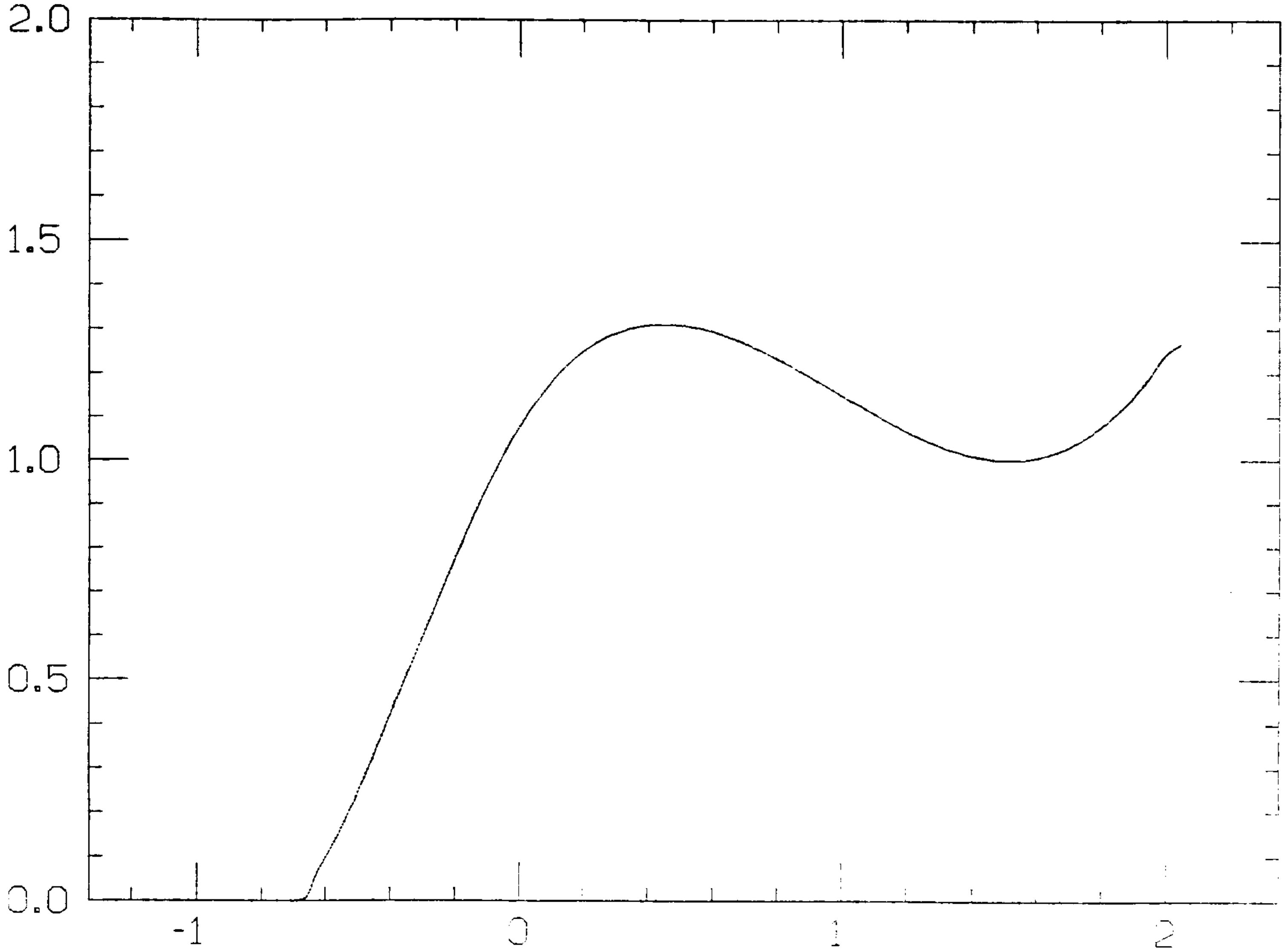

FIGURE 2.6. ØPTIMAL FUNCTIØN FØR HIGH GRØWTH GRØUP
SECØND PRØJECTIØN, EXAMPLE 2
PRØJECTIØN = .86 * INCØME + .14 * BIRTHRATE - .49 * CØASTLINE

CHAPTER THREE
REMARKS ON THE MULTIPLICATIVE MODEL

§3.1 Generalization to Multiple Classes

The procedure discussed in Chapter 2 can be generalized to the case of more than two classes. Let there be K classes, denoted 1 through K. The model then becomes

$$\frac{p_i(\mathbf{x})}{p_K(\mathbf{x})} = \prod_{m=1}^{M} h_{im}(\mathbf{a}_m'\mathbf{x}), \qquad 1 \leq i \leq K-1 \tag{3.1}$$

As previously, it can be reparametrized by defining $\{f_{km}\}_{m=1,M}^{k=1,K}$ such that

$$f_{km}(z) = h_{km}(z) f_{Km}(z), \qquad 1 \leq k \leq K-1,\ 1 \leq m \leq M \tag{3.2}$$

and

$$E\left(\sum_{k=1}^{K} \prod_{m=1}^{m'} f_{km}(\mathbf{a}_m'\mathbf{x}) \right) = 1, \qquad 1 \leq m' \leq M. \tag{3.3}$$

Then

$$p_i(\mathbf{x}) = \frac{\Pi_{m=1}^{M} f_{km}(\mathbf{a}_m'\mathbf{x})}{\Sigma_{k=1}^{K} \Pi_{m=1}^{M} f_{km}(\mathbf{a}_m'\mathbf{x})}. \tag{3.4}$$

The N variables $\{Y_n\}$ are now replaced by the NK binary variables

$$I_{kn} = \begin{cases} 1 & if\ \ Y_n = k \\ 0 & otherwise \end{cases} \qquad 1 \leq k \leq K, \quad 1 \leq n \leq N. \tag{3.4}$$

The categorical regression will be applied to these NK variables rather than to the Y_n's.

The criterion that must be minimized is the analog to (2.4):

$$S_M = \frac{1}{K} \sum_{k=1}^{K} \sum_{n=1}^{N} v_n \left(I_{kn} - \frac{\Pi_{m=1}^{M} f_{km}}{\Sigma_{i=1}^{K} \Pi_{m=1}^{M} f_{im}} \right)^2. \tag{3.6}$$

(The argument of $f_{km}(\mathbf{a}_m'\mathbf{x}_n)$ has been suppressed.)

The procedure is again stepwise. At the M^{th} step, the procedure has already chosen $\{\mathbf{a}_m\}_{m=1}^{M-1}$ and $\{f_{km}\}_{m=1,M-1}^{k=1,K}$. The approximation $\sum_{i=1}^{K}\prod_{m=1}^{M} f_{im} \approx 1$ now gives

$$S_M \approx \frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N} v_n\left(I_{kn} - \prod_{m=1}^{M-1} f_{km}\right)^2. \tag{3.7}$$

Each term in k is then minimized separately:

$$\sum_{n=1}^{N} v_n\left(I_{kn} - \prod_{m=1}^{M} f_{km}\right)^2 = \sum_{n=1}^{N} v_n\left(\prod_{m=1}^{M-1} f_{km}\right)^2\left(\frac{I_{kn}}{\prod_{m=1}^{M-1} f_{km}} - f_{kM}\right)^2 \tag{3.8}$$

The estimate of f_{kM} is found by taking a local linear smooth of $\frac{I_{kn}}{\prod_{m=1}^{M-1} f_{km}(\mathbf{a}_m'\mathbf{x}_n)}$ with weights $v_n\left(\prod_{m=1}^{M-1} f_{km}\left(\mathbf{a}_m'\mathbf{x}_n\right)\right)^2$.

As before, this procedure can be viewed as modelling each probability individually as a product of smooth functions

$$\tilde{p}_i(\mathbf{x}) = \prod_{m=1}^{M} f_{km} \tag{3.9}$$

and then normalizing to account for the small discrepancies from summing to one:

$$\hat{p}_i(\mathbf{x}) = \frac{\tilde{p}_i(\mathbf{x})}{\sum_{k=1}^{K}\tilde{p}_k}. \tag{3.10}$$

§3.2. Projection Pursuit as an Extension to Discriminant Analysis

As mentioned in Chapter 1, the classical method of discrimination is linear discriminant analysis. This method is motivated by the assumption that the observations come from multivariate normal populations with means differing among groups but with identical covariance matrices. For each class k the mean vector is estimated by the group mean $\mathbf{x}_k$ and the pooled sample covariance matrix $\hat{S}$ is calculated. The conditional probability of class k given $\mathbf{x}$ is then estimated by

$$\hat{p}_k(\mathbf{x}) = \frac{\pi_k e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_k)'\hat{S}^{-1}(\mathbf{x}-\mathbf{x}_k)}}{\sum_{i=1}^{K} \pi_i e^{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_i)'\hat{S}^{-1}(\mathbf{x}-\mathbf{x}_i)}}, \tag{3.11}$$

where π_i is the prior probability of class i.

The projection pursuit algorithm can also be applied after a linear discriminant analysis. Let $\tilde{p}_k(\mathbf{x})$ denote the linear discriminant estimate (3.11) of the conditional probability of class k. Projection pursuit could build upon this model just as it added projections in the model described before. It would first seek the projection $\mathbf{a}_1$ and functions $\{f_{k1}\}_{k=1}^{K}$ that minimize

$$S(\mathbf{a}_1) = \frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N} v_n\left(I_{kn} - \tilde{p}_k(\mathbf{x}_n) f_{k1}(\mathbf{a}_1'\mathbf{x}_n)\right)^2 \tag{3.12}$$

$$= \frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N} v_n(\tilde{p}_k(\mathbf{x}_n))^2\left(\frac{I_{kn}}{\tilde{p}_k(\mathbf{x}_n)} - f_{k1}(\mathbf{a}_1'\mathbf{x}_n)\right)^2 \tag{3.13}$$

So the estimate of f_{k1} would be the local linear smooth of $\frac{I_{kn}}{\tilde{p}_k(\mathbf{x}_n)}$ with weights $v_n(\tilde{p}_k(\mathbf{x}_n))^2$. If no choice of $\mathbf{a}_1$ reduces $S(\mathbf{a}_1)$ below a threshhold, the linear discriminant model is judged adequate for the data. Otherwise, the new projection is added to the model, and the procedure continues. At the M^{th} step it seeks $\mathbf{a}_M$ and $\{f_{kM}\}_{k=1}^{K}$ that minimize

$$S_M(\mathbf{a}_M) = \frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N} v_n\left(I_{kn} - \tilde{p}_k(\mathbf{x}_n)\prod_{m=1}^{M-1} f_{km}(\mathbf{a}_m'\mathbf{x}_n)\; f_{kM}(\mathbf{a}_M'\mathbf{x}_n)\right)^2. \tag{3.14}$$

Then f_{kM} is taken to be the local linear smooth of $\frac{I_{kn}}{\tilde{p}_k(\mathbf{x}_n)\prod_{m=1}^{M-1} f_{km}(\mathbf{a}_m'\mathbf{x}_n)}$ with weights $v_n(\tilde{p}_k(\mathbf{x}_n)\prod_{m=1}^{M-1} f_{km}(\mathbf{a}_m'\mathbf{x}_n))^2$, and $\mathbf{a}_M$ is optimized numerically. The estimates $\hat{p}_i(\mathbf{x})$ are then taken to be

$$\hat{p}_i(\mathbf{x}) = \frac{\tilde{p}_i(\mathbf{x})\prod_{m=1}^{M} f_{im}(\mathbf{a}_m'\mathbf{x})}{\sum_{k=1}^{K}\tilde{p}_k(\mathbf{x})\prod_{m=1}^{M} f_{km}(\mathbf{a}_m'\mathbf{x})} \tag{3.15}$$

If the data do come from a Gaussian distribution with equal covariances, the procedure will usually not find any projection that would noticeably improve the fit. The linear discriminant estimates would be left intact. If major discrepancies were present, the procedure would attempt to find projections that would correct for these differences. Thus the procedure could be used as an extension and safeguard to discriminant analysis, correcting for discrepancies from normality. This could also be applied in the same manner to quadratic discriminant analysis.

As an example of this, consider the first example from chapter 2. Half of the 500 observations were from a standard four dimensional Gaussian, and the rest were from a Gaussian with mean $(1.5, 0, 0, 0)$ and covariance matrix

$$\Sigma = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}.$$

The linear discriminant analysis correctly found the shift in the first variable, but did not detect the dispersion difference. Its squared distance was .152, with misclassification risk of 22.4%.

Applying projection pursuit after this analysis, the procedure selects as its first correcting projection $(0, 1, 0, 0)$. The function f_{11} is shown in Figure 3.1. It takes high values in the center, adjusting the class 1 probability upward there. In the tails it lowers the probability, essentially to zero in the extremes. The squared distance is reduced to .131 and the misclassification risk to 18.6%. A second projection, (0,0,0,1), reduces the distance to .120 and the risk to 17.4%. A third, (0,0,1,0) brings them down to .110 and 14.4%. The functions are shown in Figures 3.2 and 3.3.

When applied to a quadratic discriminant analysis (the correct model), no projection was found that would decrease sufficently the squared distance. Hence the quadratic discriminant model would be judged adequate, and no correction would be necessary.

§3.3. Choice of Minimization Criterion

In determining which projection best fits the data, a mean squared criterion was utilized

$$S = \sum_{n=1}^{N} v_n (Y_n - \hat{p}_1(\mathbf{x}))^2,$$

with the weights depending only on the priors π_k and the number in each class. This is an estimate of the quantity

$$\int (p_1(\mathbf{x}) - \hat{p}_1(\mathbf{x}))^2 \, dF(\mathbf{x}) + \int p_1(\mathbf{x})(1 - p_1(\mathbf{x})) \, dF(\mathbf{x}). \tag{3.16}$$

Since the second term in (3.16) does not depend on the estimate $\hat{p}_1$, minimizing S minimizes the estimated weighted L_2 distance between p_1 and $\hat{p}_1$.

There are several other possible criteria. Most notable among these is a variance–weighted squared distance. Since $var(Y_n) = p_1(\mathbf{x}_n)(1 - p_1(\mathbf{x}_n))$, it might seem reasonable to instead minimize

$$S' = \frac{\sum_{n=1}^{N} \frac{v_n}{\hat{p}_1(\mathbf{x}_n)(1-\hat{p}_1(\mathbf{x}_n))} (Y_n - \hat{p}_1(\mathbf{x}_n))^2}{\sum_{n=1}^{N} \frac{1}{\hat{p}_1^2(\mathbf{x}_n)(1-\hat{p}_1(\mathbf{x}_n))^2}}. \tag{3.17}$$

Such a criterion, however, gives too much weight to areas of very high or low probability. If a projection can be found in which a small section of the smoothed function is near zero or one, the weight assigned to the observations in that section can completely dominate the contributions from other parts. As an extreme example, suppose that in one projection, a small region of the range has $\hat{p}_1(\mathbf{x}) = 0$. Then the observations there would receive infinite weight, causing all other observations to be ignored, no matter how well or poorly they fit. In classification applications, it is much more important for the estimate to differentiate well in regions of overlap than in areas of extreme probabilities. These are the sections where there is the greatest doubt as to the best classification. Whether an observation has $\hat{p}_1 = .01$ or $.001$ will usually not affect its classification; whether it is .45 or .55 is much more likely to. Yet a variance–weighted criterion weights the observation in the first situation much heavier than one in the second, increasing the prospect that that projection would be selected.

As an example, consider Figures 3.4 and 3.5, representing two possible selections as the first projection in a model. Assume that the observations are uniformly spaced between zero and one. The first projection gives quite good separation of the groups over most of the range. The second offers excellent discrimination near one edge, but hardly any elsewhere. Yet the variance–weighted criterion would prefer the second, despite its failure on most of the data. (The unweighted distances are .156 and .210, while the variance–weighted ones are .142 and .120.) This reason is more compelling when using smoothing algorithms than when fitting parametric families. The smoother must fit the data locally, and not globally as in most parametric models. So it has much more potential of finding projections where the function fits very well in small regions. Heavily weighting these regions can give poor overall results.

§3.4. Comparison of the Additive and Multiplicative Models.

In comparing the additive and multiplicative models over a large number of real and simulated data sets, a general pattern developed. When the degree of discrimination between groups was small or moderate, both methods performed roughly the same. Frequently the multiplicative model fit slightly better, but only marginally so. As the separation increased, however, the discrimination of the multiplicative model improved much more markedly than did that of the additive.

As an example consider Table 3.2. It contains the validated squared distances and risks provided by various simulated Gaussians for each model. The first group consisted of 250 four–dimensional Gaussians with mean (0,0,0,0) and identity covariance matrix. The second group of 250 has mean (α,0,0,0) and covariance

$$\Sigma = \begin{pmatrix} \alpha & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda \end{pmatrix}.$$

As the separation between groups sharpens (larger α and λ), the multiplicative model improves its discrimination much faster than the additive.

An extreme case occurs in Table 3.3. Here the two groups are completely distinct. Group 1 consists of 250 observations uniformly distributed over the four-dimensional sphere $|\mathbf{x}| \leq 1$, and group 2 contains 250 distributed uniformly in the annulus $1 \leq |\mathbf{x}| \leq 3$. The additive model is unable to lower the misclassification rate below 12%, while the multiplicative model is able to do eight times better. (The rate does not go to zero in the multiplicative model because the smoother blurs somewhat the sharp boundary between the two groups.)

A closer look at this extreme case may help to explain the performances of the two methods. Consider the points A and B in the cross section formed by the first two projections (Figure 3.6). In the first step $f_{11}(B) \approx 0$, while $f_{11}(A)$ is rather large (roughly .7). Both are at the same position in the second projection. Considering for the moment only these two points, the multiplicative model's smooth at that position would be

$$f_{12} = \frac{f_{11}^2(A)\frac{1}{f_{11}(A)} + f_{11}^2(B)\frac{0}{f_{11}(B)}}{f_{11}^2(A) + f_{11}^2(B)} = \frac{f_{11}(A)}{f_{11}^2(A) + f_{11}^2(B)}. \tag{3.18}$$

So the estimated probabilities would be

$$\tilde{p}_A = \frac{f_{11}^2(A)}{f_{11}^2(A) + f_{11}^2(B)} \tag{3.19}$$

and

$$\tilde{p}_B = \frac{f_{11}^2(B)}{f_{11}^2(A) + f_{11}^2(B)} \tag{3.20}$$

As $f_{11}(B)$ is very small, these approach the correct $\tilde{p}_A \approx 1$ and $\tilde{p}_B \approx 0$. Note that the effect of B on the second smooth was minimal, because of its light weight. Despite its lack of effect on the smooth, its fit is still good, since the small first term $f_{11}(B)$ keeps the product near zero regardless of the second term. So, in the multiplicative model, if any projection reveals an area of near zero probability, the points in that area are downweighted to make little effect on future smooths. Also, their already good fit can not be drastically altered by the addition of the new projection, since the previous term keeps the product very small.

In the additive model this downweighting and protection do not occur. Again considering only the two points A and B, the local smooth would give the value $\frac{f_{12}=1-f_{11}(A)-f_{11}(B)}{2}$. Both points would be equally weighted. The new additive fits would be

$$\hat{p}_A = \frac{1+f_{11}(A)-f_{11}(B)}{2} \tag{3.21}$$

and

$$\hat{p}_B = \frac{1+f_{11}(B)-f_{11}(A)}{2}. \tag{3.22}$$

Since $f_{11}(B)$ is small,

$$\hat{p}_A \approx \frac{1+f_{11}(A)}{2} \approx .85 \tag{3.23}$$

$$\hat{p}_B \approx \frac{1-f_{11}(A)}{2} \approx .15. \tag{3.24}$$

The fit of A has improved, but that of B has deteriorated substantially. While seeking to fit A somewhat better, the procedure was disrupting the good fit of B. The multiplicative model avoided disturbing B's fits, while at the same time providing a superior fit for A. When several projections are used (as in this example) such effects can accumulate, preventing any net improvement. So in data sets where very good discrimination is not possible along more than one projection, the two methods yield comparable results. However, when several projections provide sharp, complementary distinctions, a multiplicative model performs substantially better. No patterns of additive superiority were observed.

Table 3.1 Criterion and Misclassification Rates for Training and Validation Samples Example 1 (Gaussian, Expanded)

	Criterion		Misclassification Rate (%)	
Method	*Training*	*Validation*	*Training*	*Validation*
Linear Discriminant Analysis				
Alone	.152	.156	22.4	22.9
One Projection Added	.131	.138	18.6	22.9
Two Projections Added	.120	.122	17.4	17.8
Three Projections Added	.110	.110	14.4	15.5
Logistic Regression	.152	.156	22.4	22.9
Additive Projection Pursuit				
One Projection	.152	.155	22.2	22.7
Two Projections	.133	.140	19.8	20.6
Three Projections	.118	.128	18.0	18.8
Four Projections	.110	.123	17.4	17.9
Multiplicative Projection Pursuit				
One Projection	.152	.155	22.2	22.7
Two Projections	.130	.136	19.0	20.0
Three Projections	.120	.122	17.8	17.7
Four Projections	.105	.110	14.8	15.5
Quadratic Discriminant Analyis	.096	.100	13.2	14.1
One Projection Added	.094	.100	13.0	14.2

Table 3.2 Validated Squared Distances and Misclassification Rates For Additive and Multiplicative Models For Various Values of Shift and Dispersion

		Distance		Misclassification Rate (%)	
α	λ	*Additive*	*Multiplicative*	*Additive*	*Multiplicative*
.5	2	.222	.229	36.4	38.1
1.0	2	.184	.192	28.1	30.0
1.0	4	.146	.131	21.1	18.5
1.5	4	.123	.122	17.9	15.5
1.5	9	.105	.090	15.1	11.9
2.0	9	.086	.071	12.2	9.6
2.0	16	.078	.054	11.2	7.1
2.5	16	.061	.041	8.2	5.0
2.5	25	.059	.028	7.8	2.7
3.0	25	.043	.021	5.4	1.7

Table 3.3 Validated Squared Distances and Misclassification Rates For Additive and Multiplicative Models Example 3 (Sphere and Annulus)

	Distance		Misclassification Rate (%)	
Projection	*Additive*	*Multiplicative*	*Additive*	*Multiplicative*
0	.25	.25	50.	50.
1	.16	.16	23.	23.
2	.12	.09	18.	12.
3	.10	.04	15.	5.
4	.082	.022	12.4	2.2
5	.080	.019	12.2	2.0
6	.082	.014	12.2	1.6
7	.081	.014	12.2	1.5

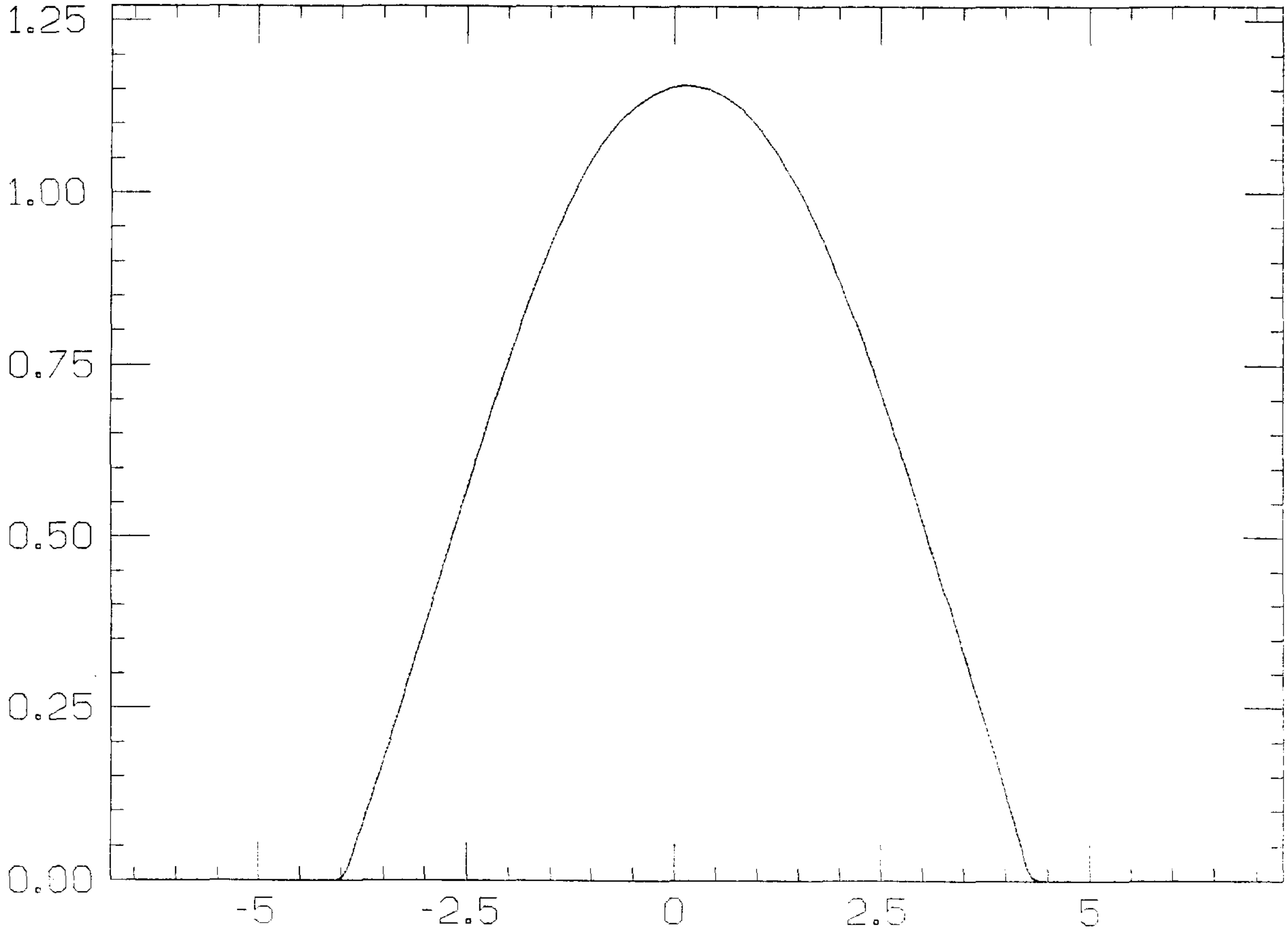

FIGURE 3.1. EXTENSIØN TØ LINEAR DISCRIMINANT ANALYSIS.
FIRST PRØJECTIØN, EXAMPLE 1
PRØJECTIØN = X2

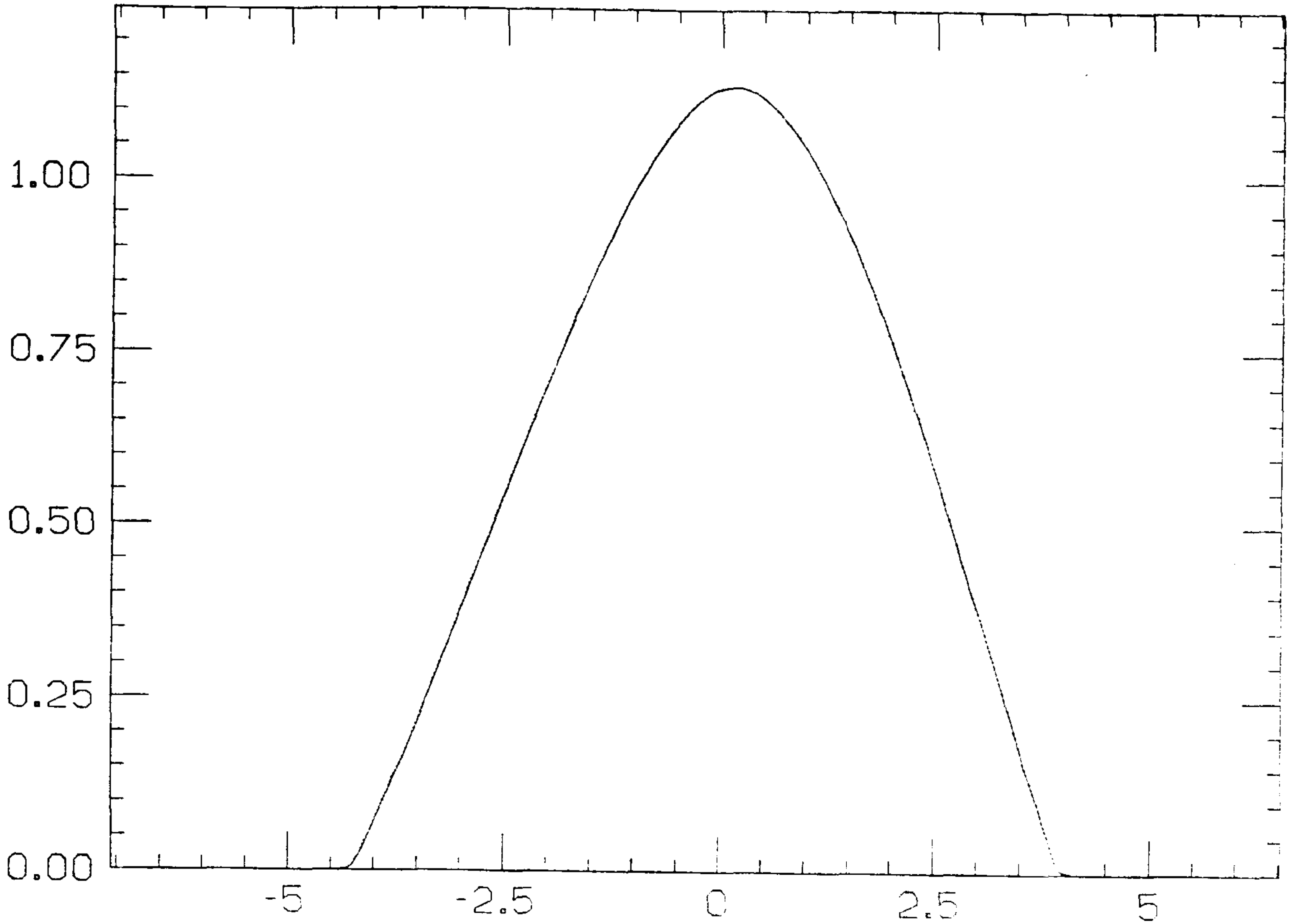

FIGURE 3.2. EXTENSIØN TØ LINEAR DISCRIMINANT ANALYSIS.
SECØND PRØJECTIØN, EXAMPLE 1
PRØJECTIØN = X4

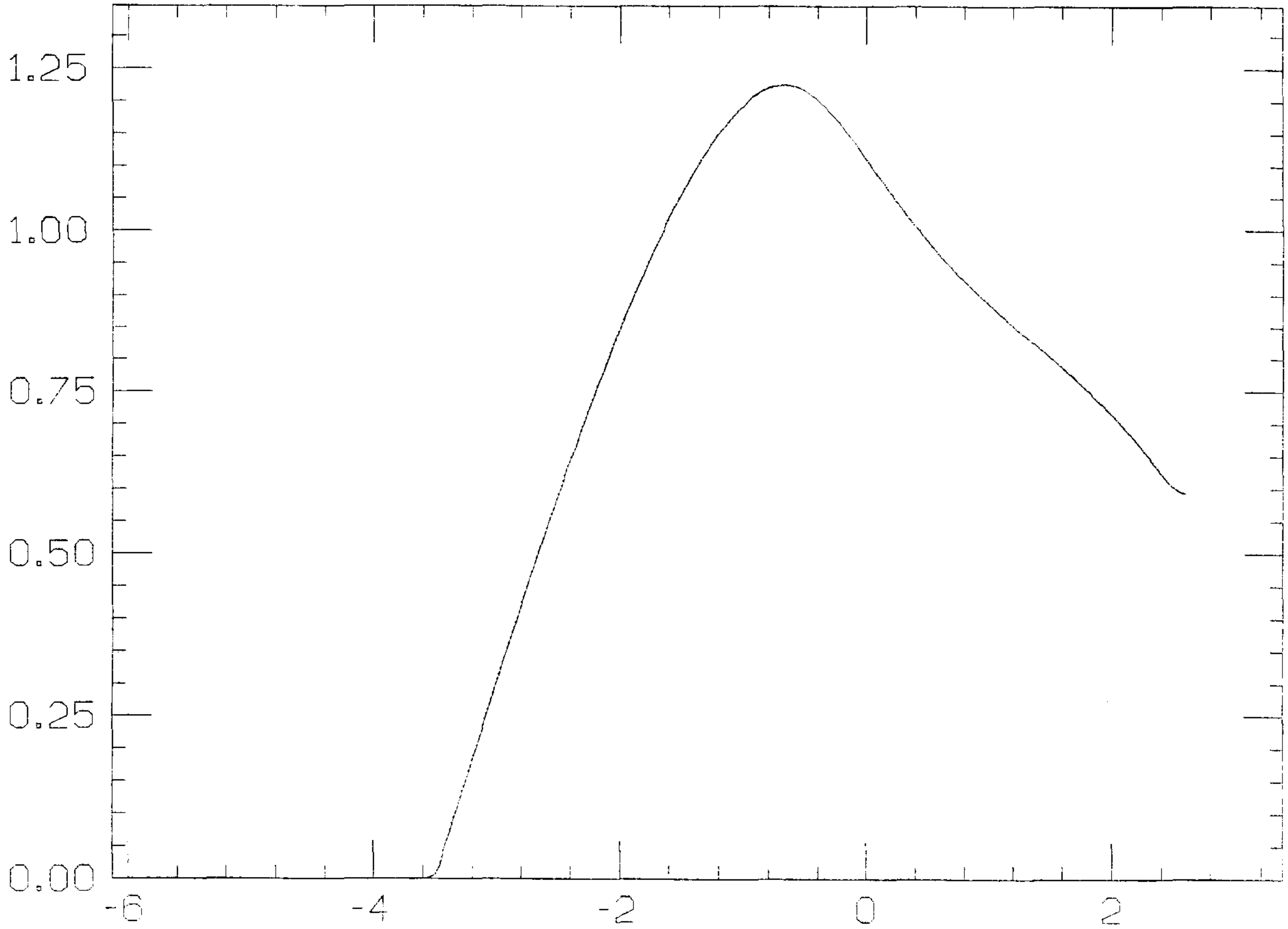

FIGURE 3.3. EXTENSIØN TØ LINEAR DISCRIMINANT ANALYSIS.
THIRD PRØJECTIØN, EXAMPLE 1
PRØJECTIØN = X3

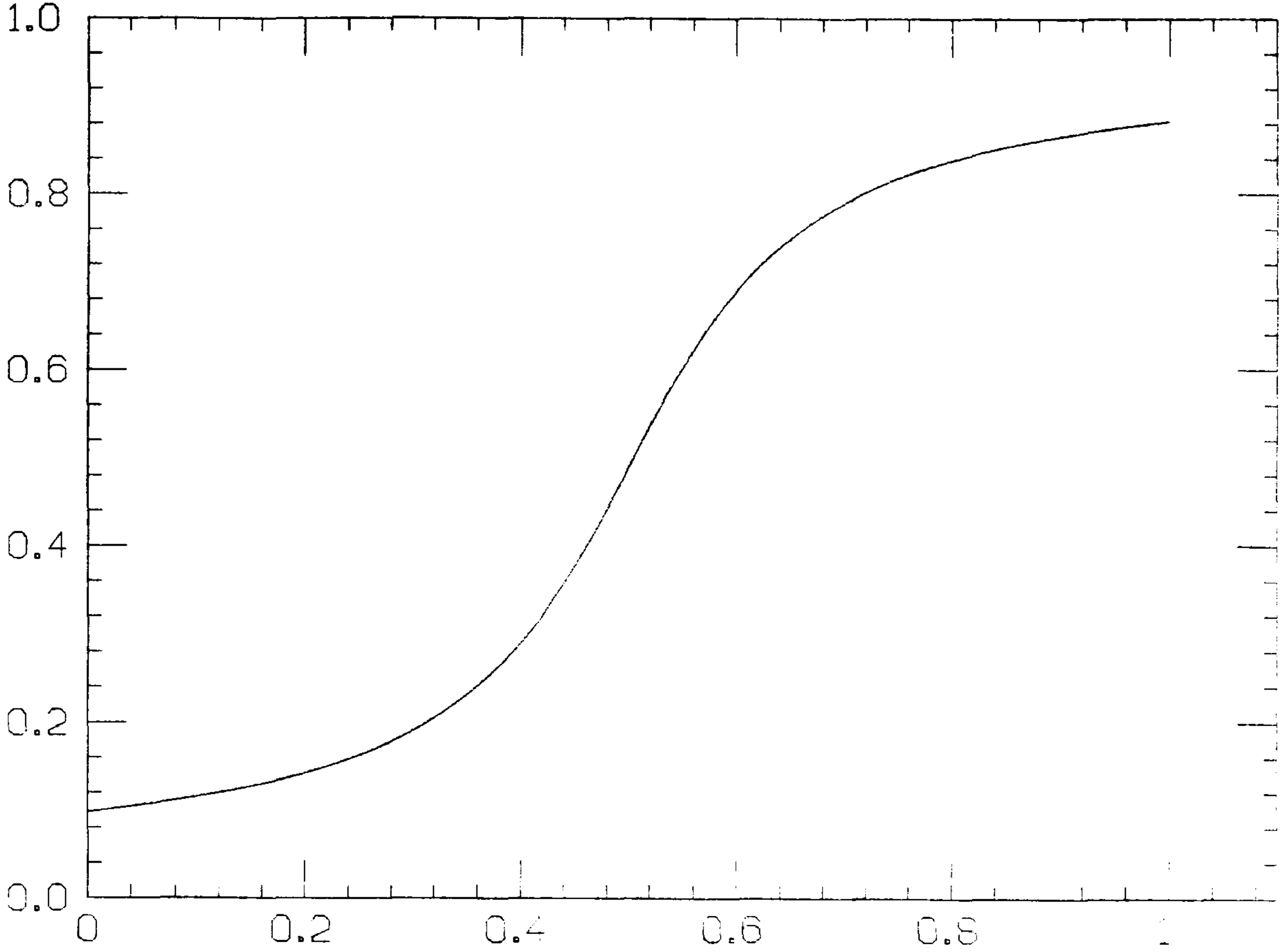

FIGURE 3.4. SMØØTH FØR FIRST PRØJECTIØN CANDIDATE.

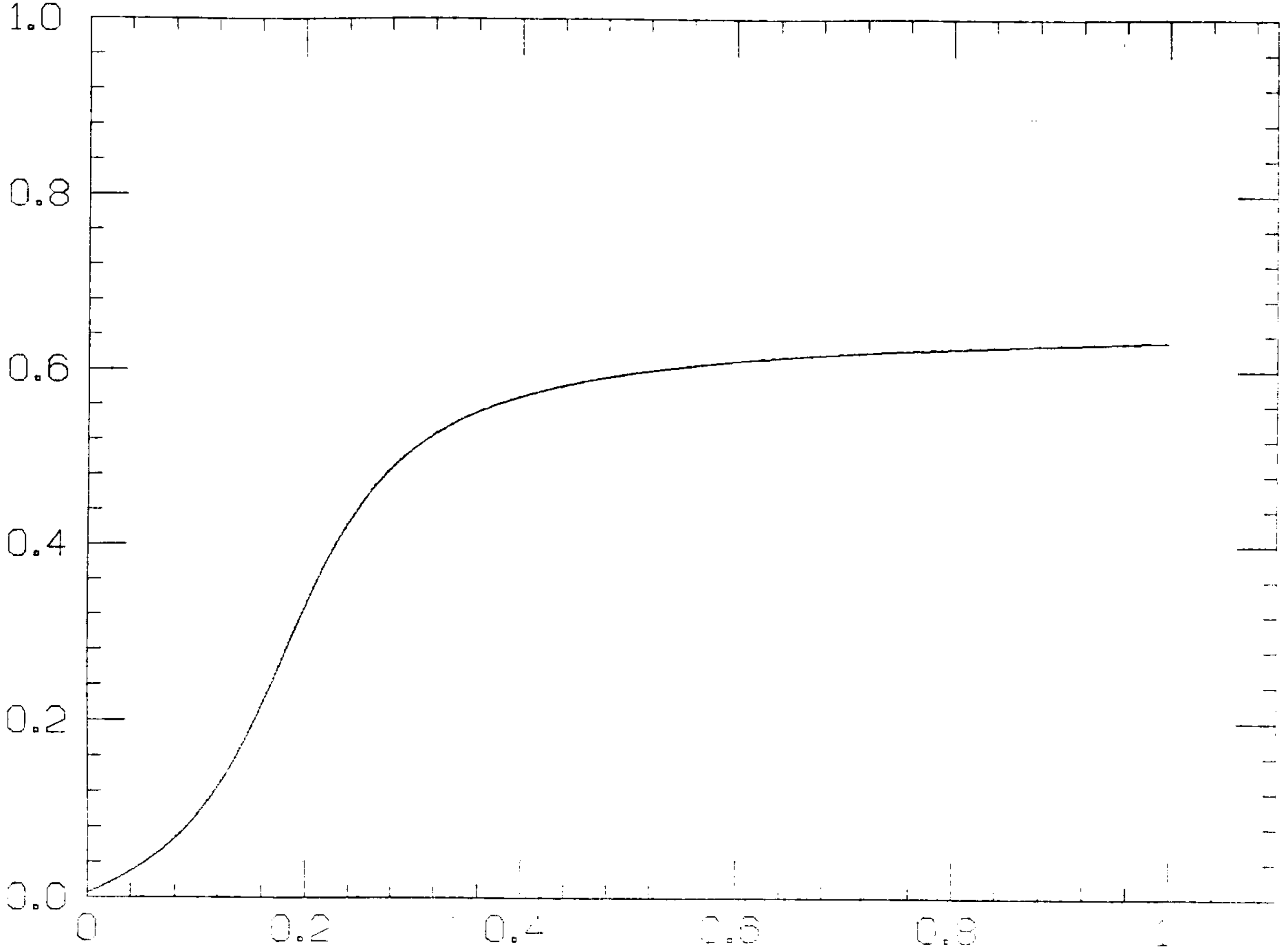

FIGURE 3.5. SMØØTH FØR SECØND PRØJECTIØN CANDIDATE.

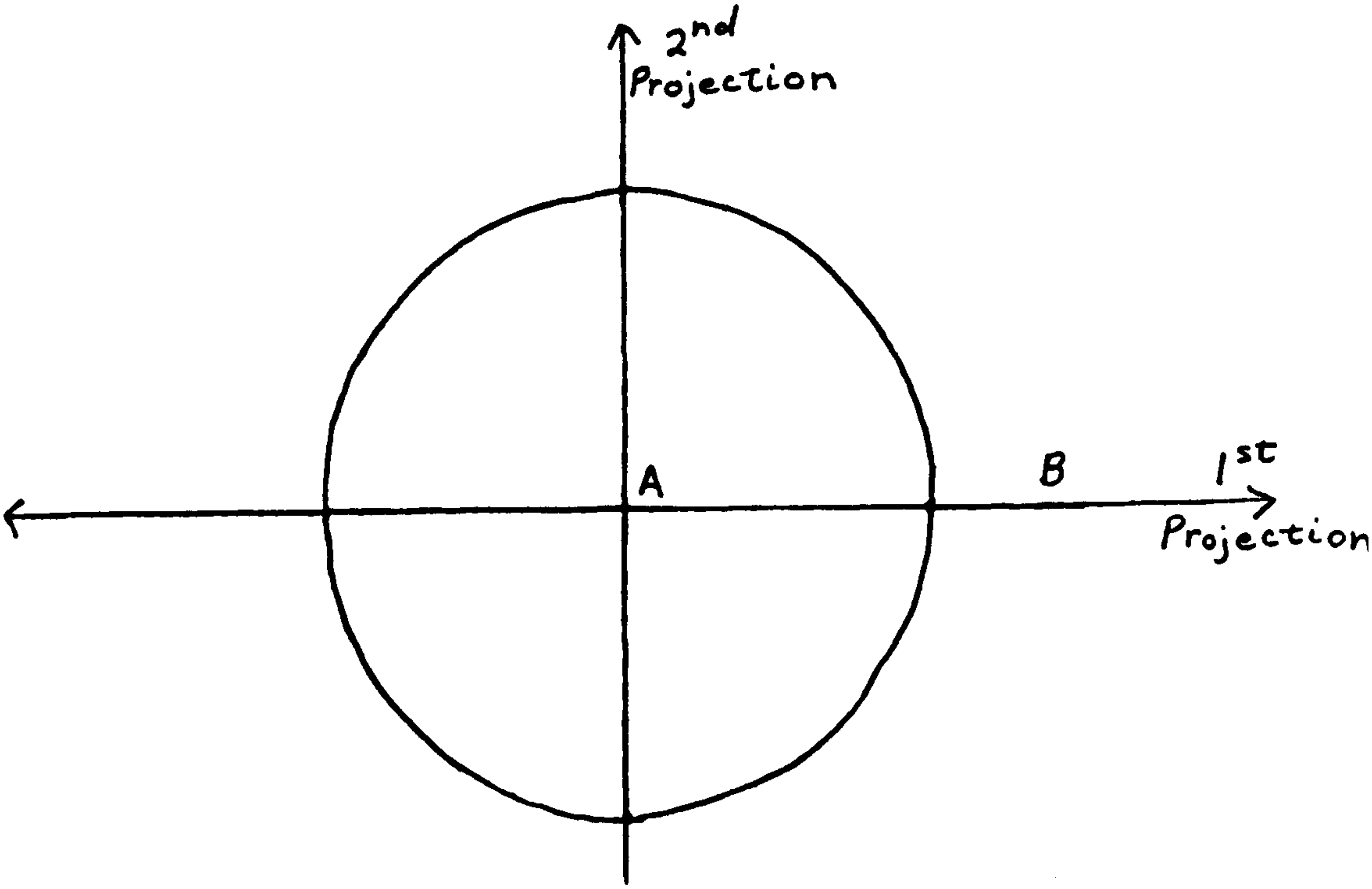

Figure 3.6. Cross–section of Sphere/Annulus Example

CHAPTER FOUR
MULTIPLICATIVE REGRESSION MODELS

§4.1 Description of the Models.

The next situation to be considered is that of regression. N random vectors are observed:

$$(Y_1, \mathbf{x}_1), (Y_2, \mathbf{x}_2), \ldots, (Y_N, \mathbf{x}_N), \tag{4.1}$$

where $\mathbf{x}_i$ is a p–dimensional vector of variables (called predictor variables) and Y_n is the univariate response. Regression analysis seeks to estimate the conditional expectation of Y given $\mathbf{x}$ (called the response surface) based on the observed sample.

Various methods of regression are in use. Most assume that the response surface is a member of some specified parametric family. The parameters of the family are then estimated by the regression procedure, either algebraically or numerically. Most common is linear regression, where the conditional expectation is modelled as linear in some combination of the predictors:

$$E(Y \mid \mathbf{x}) = \mathbf{a}'\mathbf{x} + b. \tag{4.2}$$

When correctly specifiying the underlying parametric family, such methods perform very well. When the family is incorrectly chosen, erroneous conclusions can result.

To avoid the dangers caused by misspecifying the parametric family, various nonparametric methods have beeen developed. These techniques relax the assumptions about the response surface. One of these methods is projection pursuit regression (Friedman and Stuetzle, 1981). It models the conditional expectation as the sum of functions of linear combinations of the predictors:

$$E(Y \mid \mathbf{x}) = \sum_{m=1}^{M} f_m(\mathbf{a}_m'\mathbf{x}). \tag{4.3}$$

The procedure is similar to that described in chapter 1 for categorical regression. For any projection $\mathbf{a}$, an unweighted local linear smooth of Y_n versus $\mathbf{a}'\mathbf{x}_n$ gives the estimate of the smooth function for that projection. A mean squared error criterion

$$S(\mathbf{a}) = \frac{1}{N} \sum_{n=1}^{N} (Y_n - f(\mathbf{a}'\mathbf{x}_n))^2 \tag{4.4}$$

is evaluated. The projection that minimizes $S(\mathbf{a})$ is found numerically. That projection and the corresponding function are taken to be $\mathbf{a}_1$ and f_1. The residuals $Y_n - f_1(\mathbf{a}_1'\mathbf{x}_n)$ are then subjected to the same procedure. In this way a new projection $\mathbf{a}_2$ and function f_2 are found. The new residuals are then processed, and the procedure repeats until no substantial decrease in S is obtained. In this manner the regression surface is estimated by a sum of smooth functions.

Such an additive model can work quite effectively for many response surfaces. For others, the structure can be better approximated in other ways. In seeking to most accurately estimate the surface, various other possible models must be explored.

One possibility is a multiplicative model. Y is assumed to depend not on the sum of smooth functions, but on their product:

$$Y = \prod_{m=1}^{M} f_m(\mathbf{a}_m'\mathbf{x}) + \epsilon. \tag{4.5}$$

There are now several approaches. First, logarithms may be taken and additive regression applied to $\log Y$. This makes the implicit assumption that the magnitude of the error distribution given $\mathbf{x}$ is proportional to $\prod_{m=1}^{M} f_m(\mathbf{a}_m'\mathbf{x})$:

$$\epsilon = \prod_{m=1}^{M} f_m(\mathbf{a}_m'\mathbf{x})\ \gamma, \tag{4.6}$$

where the conditional distribution of the variable γ does not depend on $\mathbf{x}$. Then

$$\log Y = \sum_{m=1}^{M} \log f_m(\mathbf{a}_m'\mathbf{x}) + \log(1+\gamma)\,. \tag{4.7}$$

So the additive procedure can be applied to estimate $\mathbf{a}_m$ and $\log f_m$. The minimization criterion has become

$$\frac{1}{N}\sum_{n=1}^{N}\left(\log Y_n - \sum_{m=1}^{M}\log f_m(\mathbf{a}_m'\mathbf{x}_n)\right)^2, \tag{4.8}$$

rather than the least squares criterion $\frac{1}{N}\sum_{n=1}^{N}(Y_n - \hat{Y}_n)^2$. This accentuates the importance of differences in the smaller values of Y and deflates that of those in the larger values. Such an approach is reasonable, however, if the error is proportional to Y.

A second approach retains the least squares criterion. Rather than using residuals like the additive model, it employs multiplicative residuals. At the M^{th} step, $\{\mathbf{a}_m\}_{m=1}^{M-1}$ and $\{f_m\}_{m=1}^{M-1}$ have been previously determined. For any given choice of $\mathbf{a}_M$, that f_M is sought that will minimize

$$\begin{aligned} S &= \frac{1}{N}\sum_{n=1}^{N}\left(Y_n - \prod_{m=1}^{M} f_m(\mathbf{a}_m'\mathbf{x}_n)\right)^2 \\ &= \frac{1}{N}\sum_{n=1}^{N}\left(\prod_{m=1}^{M-1} f_m(\mathbf{a}_m'\mathbf{x}_n)\right)^2\left(\frac{Y_n}{\prod_{m=1}^{M-1} f_m(\mathbf{a}_m'\mathbf{x}_n)} - f_M(\mathbf{a}_M'\mathbf{x}_n)\right)^2. \end{aligned} \tag{4.9}$$

The estimate of f_M is a local linear smooth of $\frac{Y_n}{\prod_{m=1}^{M-1} f_m(\mathbf{a}_m'\mathbf{x}_n)}$ (the multiplicative residual—the ratio of Y_n to the previous step's estimate $\hat{Y}_n$) with weights $(\prod_{m=1}^{M-1} f_m(\mathbf{a}_m'\mathbf{x}_n))^2$. Again, the choice of $\mathbf{a}_M$ is determined by numerical optimization. So the procedure is similar to that used in additive projection pursuit, except that the residuals are obtained through division, and the local linear smooths are weighted.

The observed Y_n's must be positive for this approach to be valid. If they take on both positive and negative values, the estimate of f_m may be zero in places due to averaging. In those places, no further improvement would be possible, since the model always sets the product to be zero. Hence this approach is restricted to positive response values.

In the modelling of response surfaces, the matter of equivariance arises. The additive model is both location and scale equivariant in Y. Altering either or both will not affect the estimates or fit, beyond the appropriate adjustments to location and scale. The multiplicative model, however, is not location equivariant. Changes in location can drastically alter the function estimates and the fit of the model. To take this into account in obtaining the best fit, add a location parameter

$$Y = c + \prod_{m=1}^{M} f_m(\mathbf{a}'_m \mathbf{x}) + \epsilon. \tag{4.10}$$

To maintain positivity of the functions f_m, c will be restricted to the range $(-\infty, \min Y_n)$. This model is location and scale equivariant and allows the modelling of data where the range of Y is not strictly positive.

The estimation procedure is again stepwise. The first step is the same as for additive models: finding the projection $\mathbf{a}_1$ and function f_1 that minimize $\frac{1}{N}\sum_{n=1}^{N}(Y_n - f_1(\mathbf{a}'_1\mathbf{x}_n))^2$. At the second step, the best estimate of c is sought along with those of $\mathbf{a}_2$ and f_2. For any chosen values of c and $\mathbf{a}_2$, the estimate of f_2 is obtained. The choice of c and $\mathbf{a}_2$ are determined through numerical optimization, as described in the appendix.

To obtain the choice of f_2, the procedure mimics what would have taken place had c been known rather than estimated. Had it been known, the previous method (4.9) could have been applied to $\{Y_n - c\}$. The estimate of f_1 would have been c less than the function f_1 obtained in the first step here. Then the residuals would have been $\frac{Y_n - c}{f_1(\mathbf{a}'_1\mathbf{x}_n) - c}$, and the quantity to be minimized would have been

$$\begin{aligned} S_2(\mathbf{a}_2) &= \frac{1}{N}\sum_{n=1}^{N}(Y_n - c - (f_1(\mathbf{a}'_1\mathbf{x}_n) - c)f_2(\mathbf{a}'_2\mathbf{x}_n))^2 \\ &= \frac{1}{N}\sum_{n=1}^{N}(f_1(\mathbf{a}'_1\mathbf{x}_n) - c)^2\left(\frac{Y_n - c}{f_1(\mathbf{a}'_1\mathbf{x}_n) - c} - f_2(\mathbf{a}'_2\mathbf{x}_n)\right)^2 \end{aligned} \tag{4.11}$$

So the estimate of f_2 would be the local linear smooth of $\frac{Y_n - c}{f_1(\mathbf{a}'_1\mathbf{x}_n) - c}$ with weights $(f_1(\mathbf{a}'_1\mathbf{x}_n) - c)^2$.

At succeding steps (say, the M^{th} step), $\mathbf{a}_M$ and f_M are estimated, and c is reoptimized as explained in the appendix. For specified $\mathbf{a}_M$ and c, f_M is selected to minimize

$$S_M(\mathbf{a}_M) = \frac{1}{N}\sum_{n=1}^{N}\left(Y_n - c - (f_1 - c)\prod_{m=2}^{M-1} f_m \; f_M(\mathbf{a}'_M\mathbf{x}_n)\right)^2 , \tag{4.12}$$

$$= \frac{1}{N}\sum_{n=1}^{N}\left((f_1 - c)\prod_{m=2}^{M-1} f_m\right)^2 \left(\frac{Y_n - c}{(f_1 - c)\prod_{m=2}^{M-1} f_m} - f_M(\mathbf{a}'_M\mathbf{x}_n)\right)^2 \tag{4.13}$$

(The argument of $f_m(\mathbf{a}'_m\mathbf{x}_n)$ has been omitted.) So f_M is estimated by a local linear smooth of $\frac{Y_n - c}{(f_1(\mathbf{a}'_1\mathbf{x}_n)-c)\prod_{m=2}^{M-1} f_m(\mathbf{a}'_m\mathbf{x}_n)}$ with weights $((f_1-c)\prod_{m=2}^{M-1} f_m)$. The estimates of $\mathbf{a}_M$ and c are selected through numerical optimization.

§4.2 Numerical Examples.

As an example, consider the following simulated data set. The four explanatory variables are uniformly distributed on (0,1) and $Y = 12 + e^{5x_2}\sin\pi x_1 + x_3 + (1+x_4)^2 + \epsilon$, where ϵ is distributed as a normal with mean 0 and variance 4. This is a mixture of additive and multiplicative factors, with the multiplicative one ($e^{5x_2}\sin\pi x_1$) being the dominant factor. 600 observations are generated, and a smoother using 30% of them is employed.

The first projection chosen is (0,1.0,.1,0). The function (Figure 4.1) resembles the true underlying exponential (with a constant added), except that the upper tail does not rise as quickly as it should. This is the same projection and function as found by the additive model. The mean squared error is reduced 75%.

For the second projection, the procedure selects (1,0,0,0), along with a value of $c = 9.29$. (It does not choose the correct value of 12, since c can not exceed $min\,Y_n \approx 9.5$.) The function (Figure 4.2) resembles the true sine function that it is estimating. It does not approach zero at the extremes due to linearization by the smoother. The scale is larger that that of a sine curve, but this does not affect the fit. The first term was scaled down by the same factor, making the product correct. The present model is

$$Y = c + (f_1 - c)f_2.$$

Subtracting c from f_1 causes it to resemble more closely the true exponential than does Figure 4.1. The mean squared error drops drastically, accounting for 89% of the remaining variance.

The true multiplicative portion of the data's structure has now been modelled. What remains is essentially additive. Despite this, the procedure is still able to improve the fit by adding another multiplicative term. It selects (.24,.10,.96,.11). This captures most of the remaining variance due to x_3. The additive effect (a straight line) is reproduced rather well as a multiplicative effect here (Figure 4.3). The value of c is tuned slightly to 8.39. The squared error drops to 14.3, this term accounting for 39% of the remaining variance.

Two additional projections, dealing mainly with x_1 and x_2 reduce the squared error to 12.3 and to 8.3 (Figures 4.4 and 4.5). The true variance when the underlying structure is exactly known is 4.00.

Table 4.1 gives the mean squared errors for the multiplicative and additive models, and also for a strict multiplicative model with no offset c. The additive model never does as well as the multiplicative, with mean squared errors four times as large. The strict multiplicative model performs somewhat better than the additive. It falls far short of the multiplicative with offset, however. To confirm these results, Table 4.2 provides the mean squared errors for the prediction of 10,000 additional observations which were predicted using the model obtained above. Results agree quite well with the previous ones.

The next example comes from the manufacture of semiconductors. Data from 262 tested chips are being analyzed to determine the effect of various parameters on the electrical properties of the semiconductors. The independent variables, all rather technical to the field, are:

x_1 — gain measurement for an enhancement device.

x_2 — current flow from drain to source in an N–depletion device.

x_3 — gate width (length between the N^+ regions on a die).

x_4 — ohm per square area on a four micron device.

x_5 — ohm per square area on a N^+ phosphorous deposition device.

x_6 — ohm per square area on a polycrystalline device.

x_7 — threshhold voltage of an enhancement device.

The response variable is the time required to turn on the chip select[1] .

The data is initially well fit by by a linear regression, as the first step of projection pursuit indicates (Figure 4.6). The mean squared error drops to 4.46 with this projection, down from 272.78. Somewhat further reductions are possible by adding additional projections. In the multiplicative model, a second projection (Figure 4.7) reduces the mean squared error to 3.98. The projection is (0,0,0,0,0,0,1) with c chosen to be 48.85. When the enhancement device had a low threshhold voltage, the function takes smaller values. It climbs sharply as the threshhold increases, and then begins a gradual descent. (When observing Figure 4.7, it appears as though the variation in f_2 is quite small. However, a change of .05 in the value of f_2 brings about a change of $.05(f_1 - c)$ or up to 3.5 in the fit of Y.)

The third projection chosen is x_3—the length between N^+ regions on the die (Figure 4.8). As that length increases, the function climbs, levels out and climbs again. The value of c changes slightly to 48.71. The mean squared error drops to 3.41, a decline of 13%.

Table 4.3 gives the training sample mean squared errors for both the additive and multiplicative models. It also gives the crossvalidated results when the sample was randomly divided into ten groups, with each group's fit predicted by the remaining nine. The multiplicative model performs somewhat better here, though not nearly as spectacularly as in the previous example.

[1] Beumer–Browner (1981)

§4.3. Comparison of the Additive and Multiplicative Models

After comparing the methods on various real and simulated data sets, the results are not completely clear. On many data sets, the two methods provide comparable results. When they differ, the additive tends more often to be the superior. The multiplicative, however, performs better in some of the cases. One particular situation where a multiplicative model performs superiorly is when a significant portion of the data lies near the $\min Y_n$ (no long lower tail). In that case, the multiplicative model will frequently set c to be close to that minimum (effectively setting the base to be 0). It can then easily send regions of the data to that minimum by setting the function to be zero in that area. (This is similar to the case in categorical regression, where the data are restricted to (0,1), with areas where $p_1(\mathbf{x})$ is close to 0.) Beyond this situation, no patterns have developed that delineate when either method performs better than the other.

Table 4.1 Training Sample Mean Squared Errors
For Additive, Multiplicative and Strict Multiplicative Models
Example 4

Projection	*Multiplicative*	*Additive*	*Strict Multiplicative*
0	804.37	804.37	804.37
1	204.75	204.75	204.75
2	23.45	125.17	36.97
3	14.34	59.14	31.30
4	12.30	52.29	29.06
5	8.30	38.94	26.92

Table 4.2 Validated Mean Squared Errors
For Additive, Multiplicative and Strict Multiplicative Models
Example 4

Projection	*Multiplicative*	*Additive*	*Strict Multiplicative*
1	225.76	225.76	225.76
2	29.68	148.87	45.08
3	15.72	71.20	36.58
4	14.21	62.42	35.15
5	8.62	46.39	31.98

Table 4.3 Training and Crossvalidated Mean Squared Errors
Example 5 (Semiconductors)

	Training		Crossvalidated	
Projection	*Additive*	*Multiplicative*	*Additive*	*Multiplicative*
1	4.46	4.46	5.47	5.47
2	4.00	3.98	5.40	5.02
3	3.69	3.41	5.26	4.49

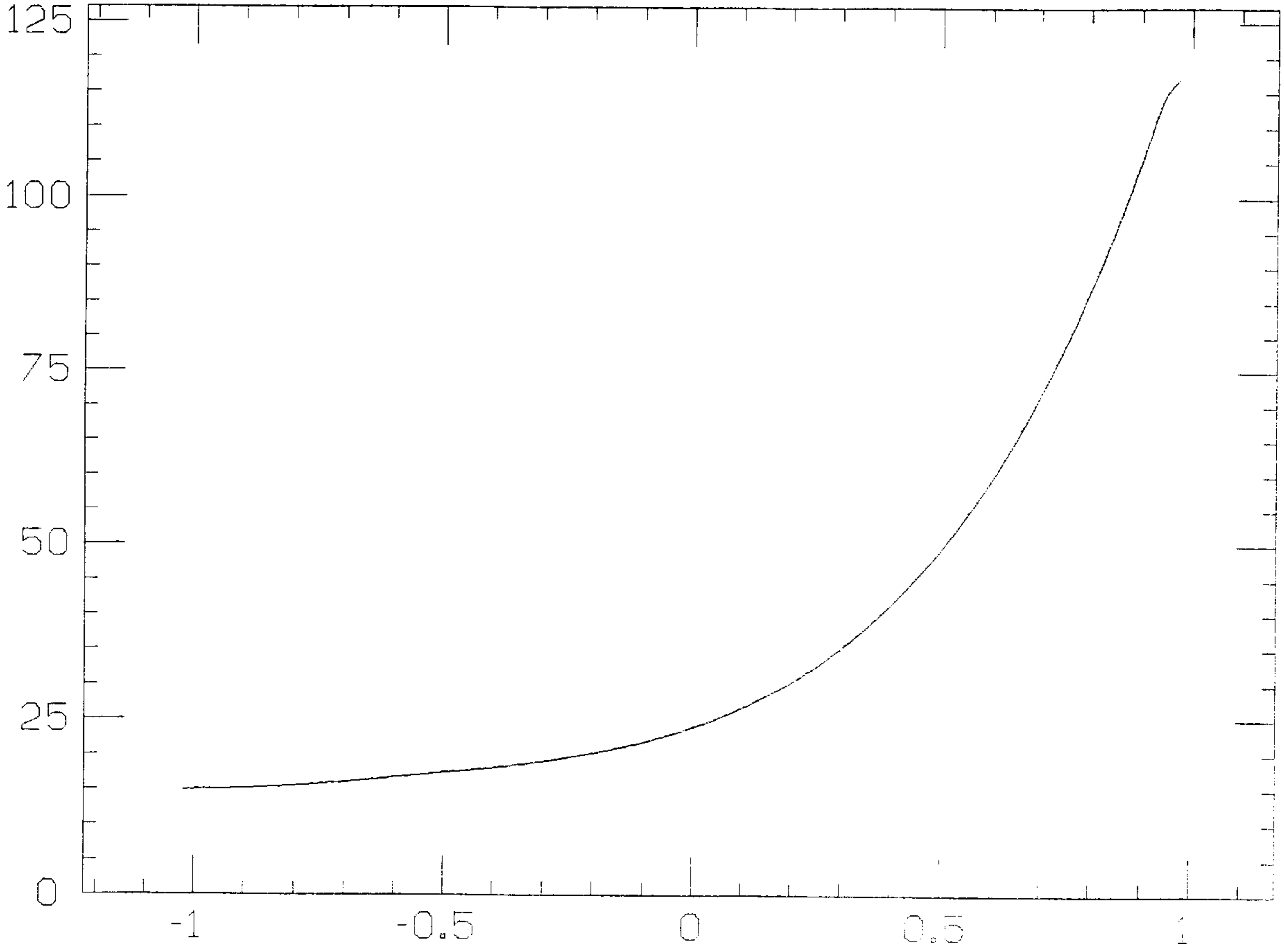

FIGURE 4.1. FUNCTIØN FØR FIRST PRØJECTIØN
EXAMPLE 1. CHAPTER 4
PRØJECTIØN = X2

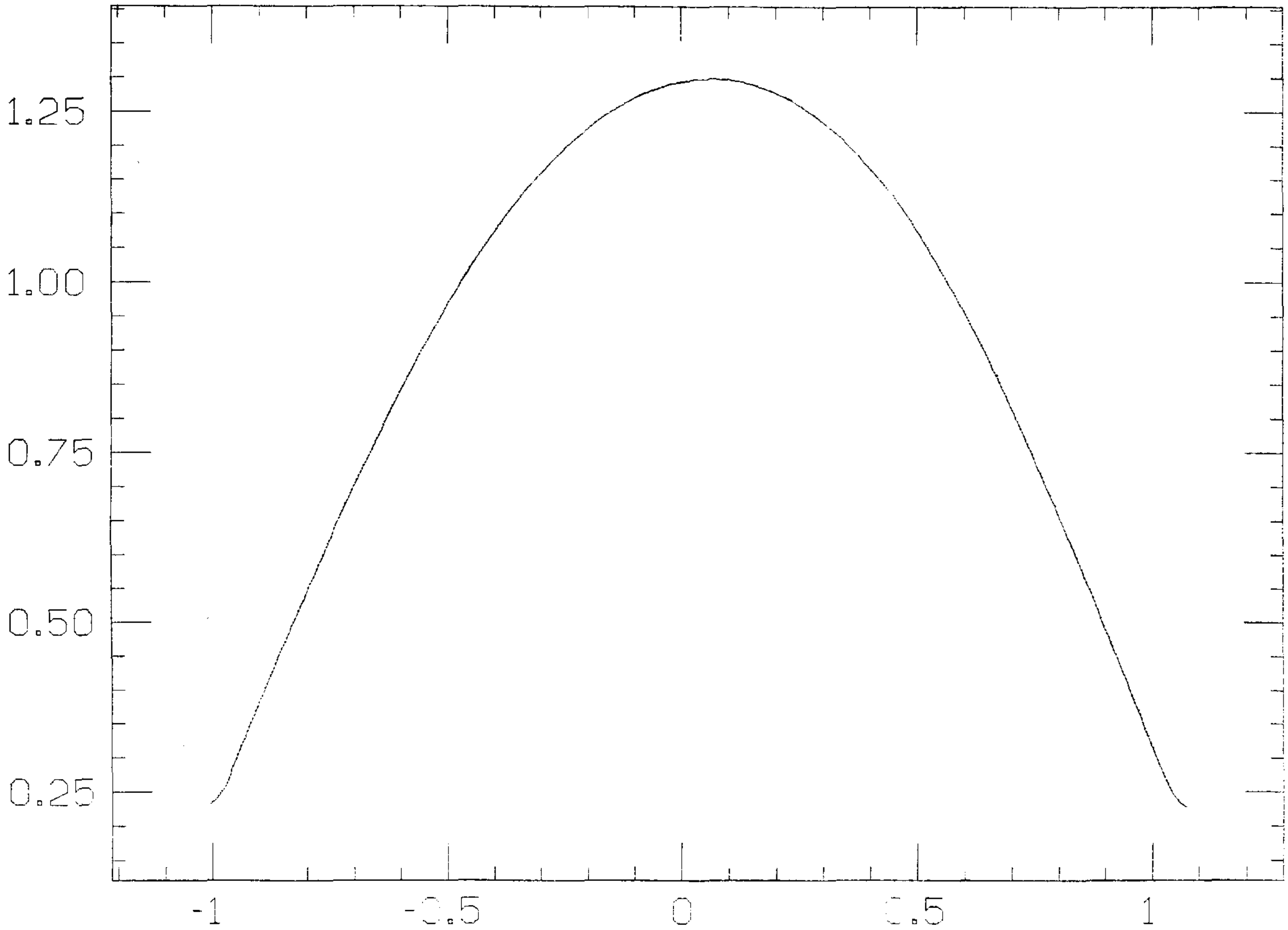

FIGURE 4.2. FUNCTIØN FØR SECØND PRØJECTIØN
EXAMPLE 1. CHAPTER 4
PRØJECTIØN = X1

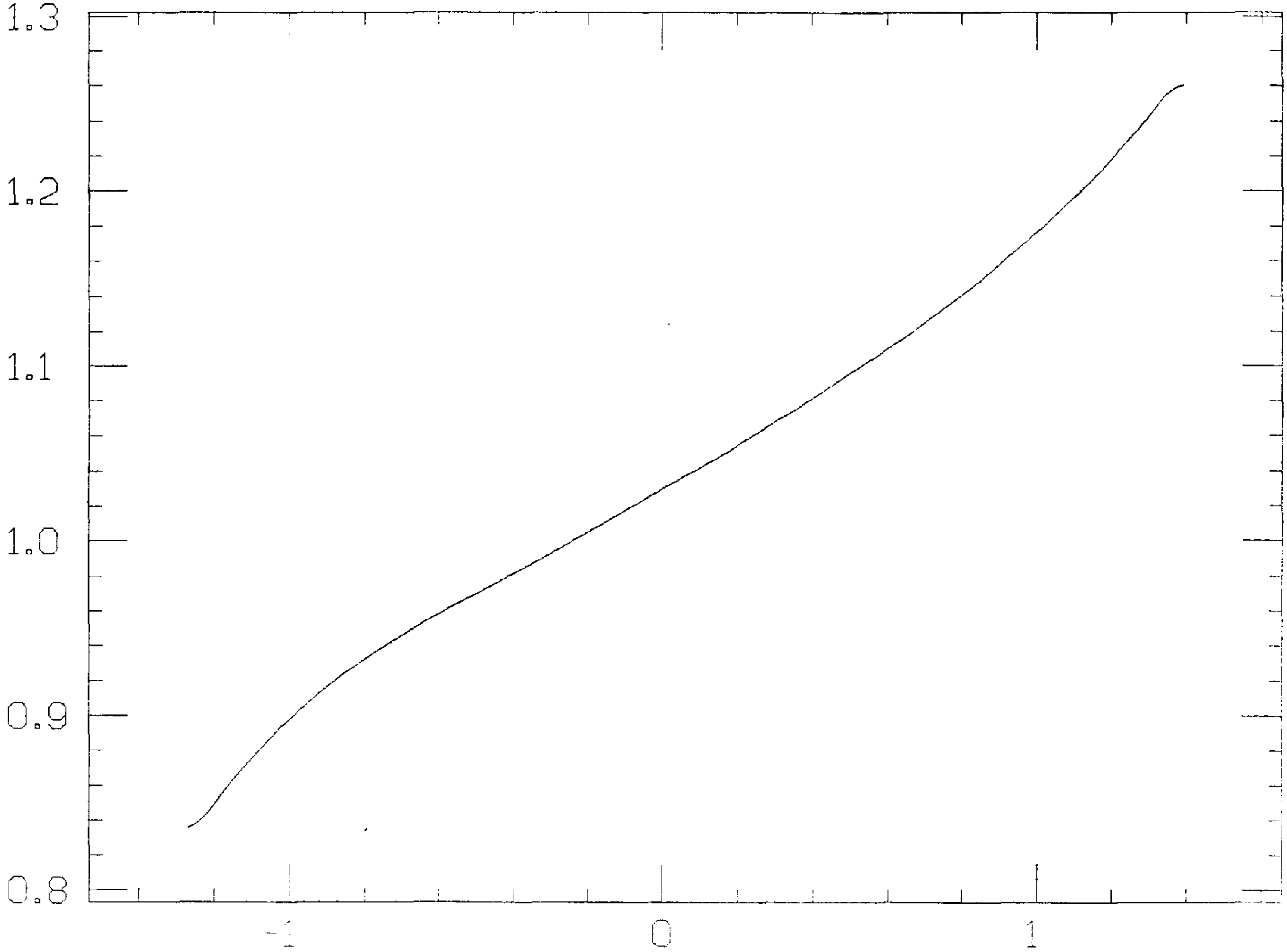

FIGURE 4.3. FUNCTIØN FØR THIRD PRØJECTIØN
EXAMPLE 1, CHAPTER 4
PRØJECTIØN = .24 * X1 + .10 * X2 + .96 * X3 + .11 * X4

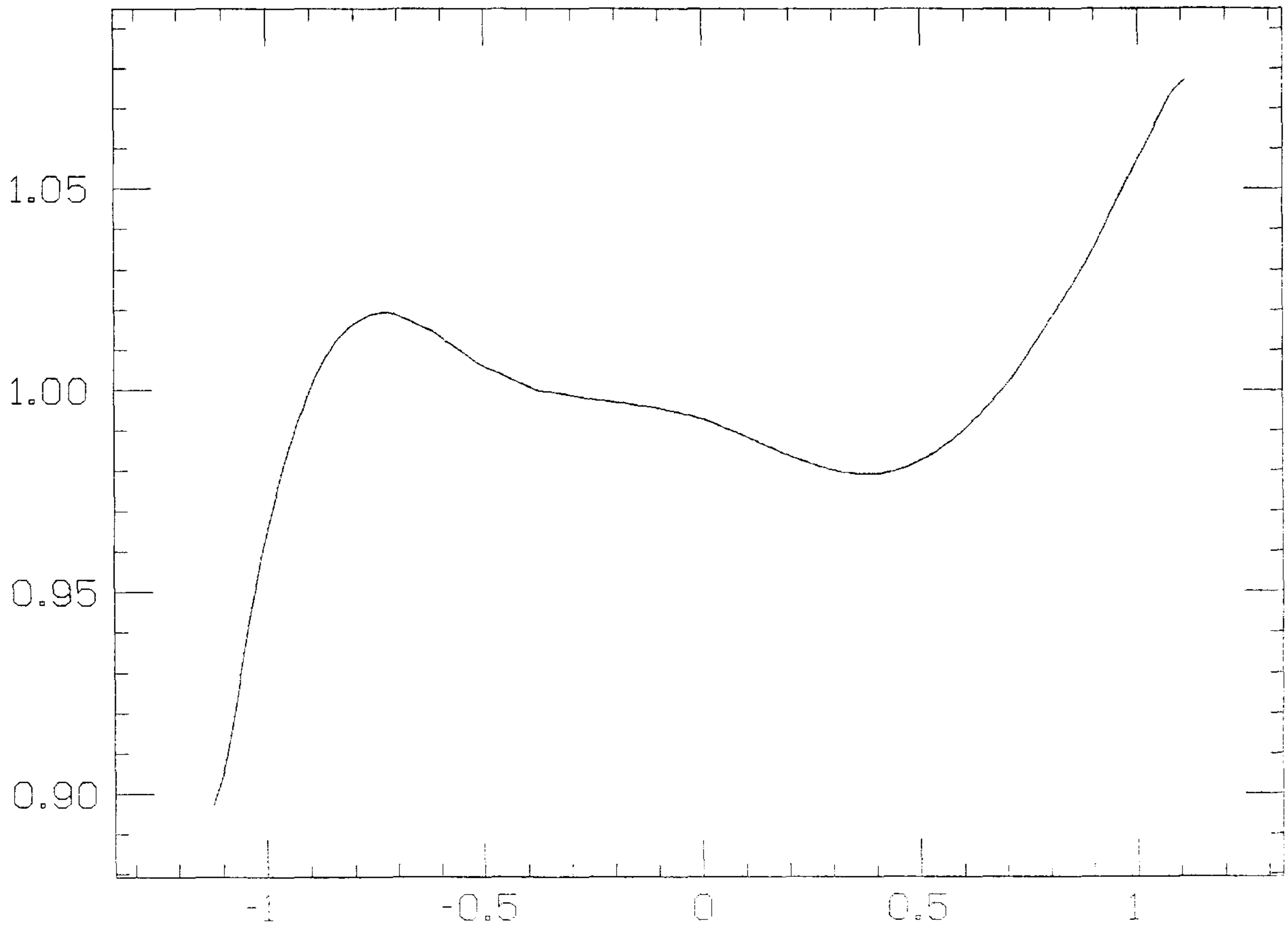

FIGURE 4.4. FUNCTIØN FØR FØURTH PRØJECTIØN
EXAMPLE 1, CHAPTER 4
PRØJECTIØN = -.10 * X1 + .98 * X2 + .10 * X3 + .20 * X4

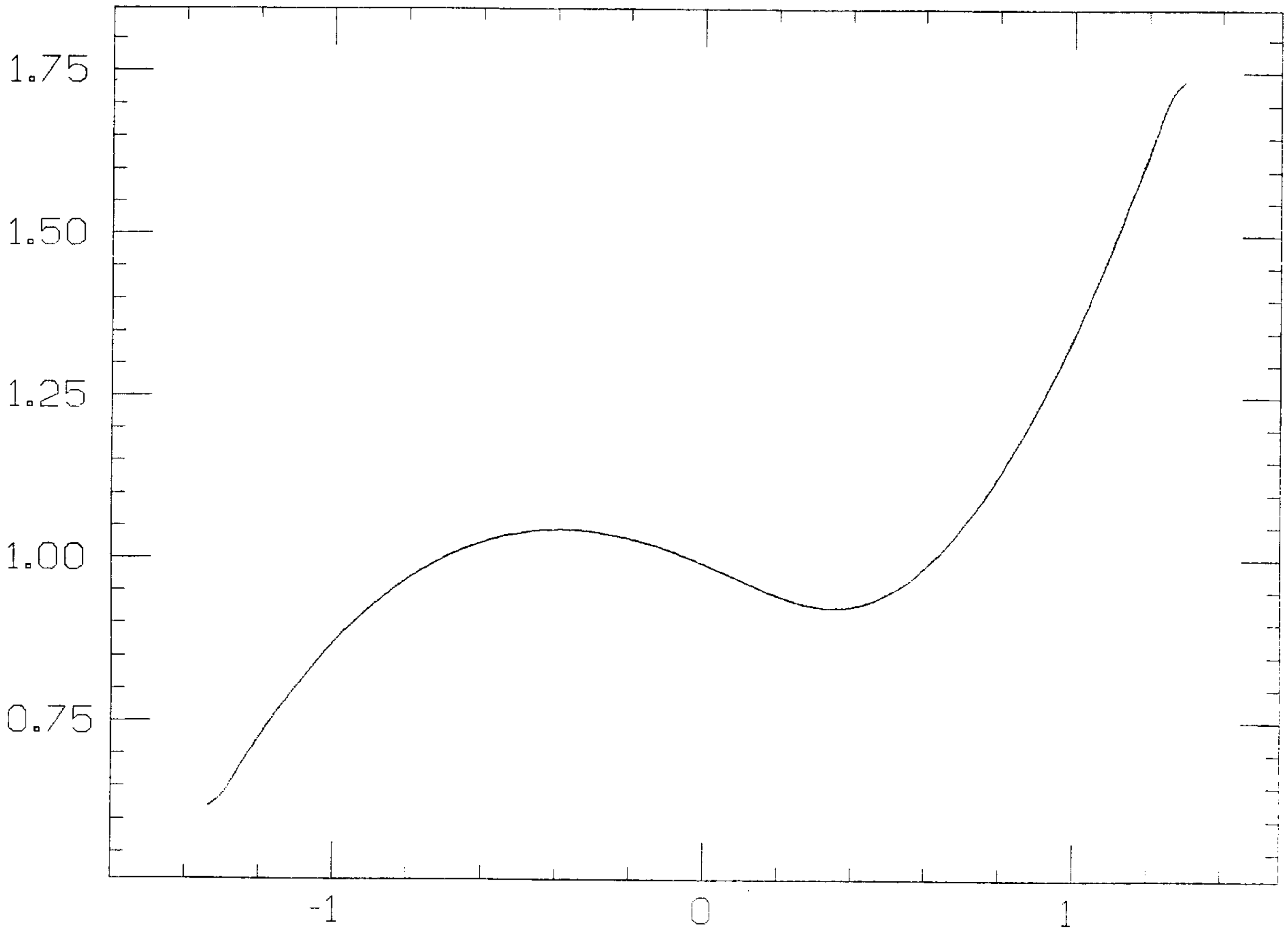

FIGURE 4.5. FUNCTIØN FØR FIFTH PRØJECTIØN
EXAMPLE 1, CHAPTER 4
PRØJECTIØN = -.8 * X1 - .5 * X2 + .1 * X3

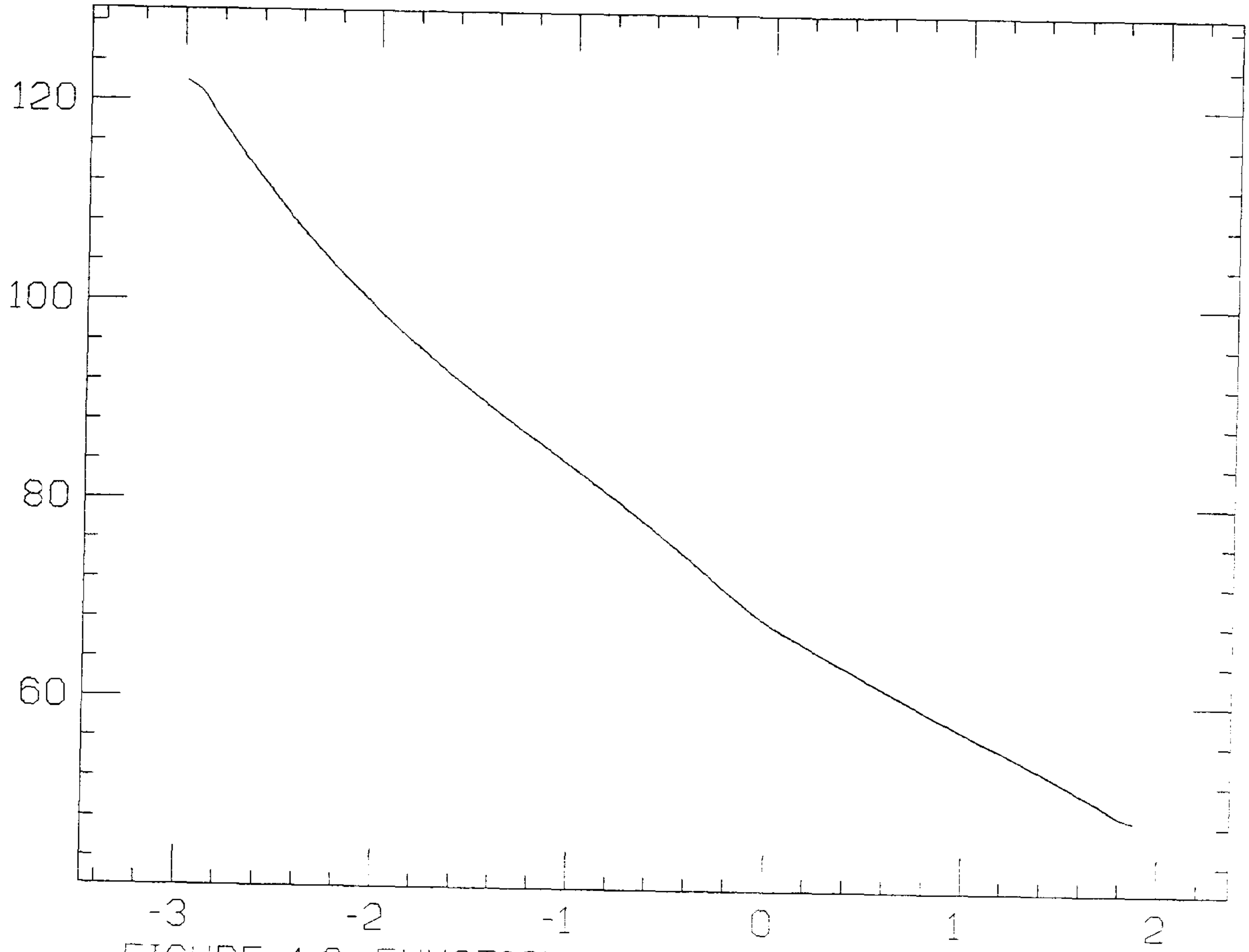

FIGURE 4.6. FUNCTIØN FØR FIRST PRØJECTIØN.
EXAMPLE 2, CHAPTER 4 (SEMICØNDUCTØRS)
PRØJECTIØN = (.18, .65, -.72, -.06, -.03, -.02, .16)

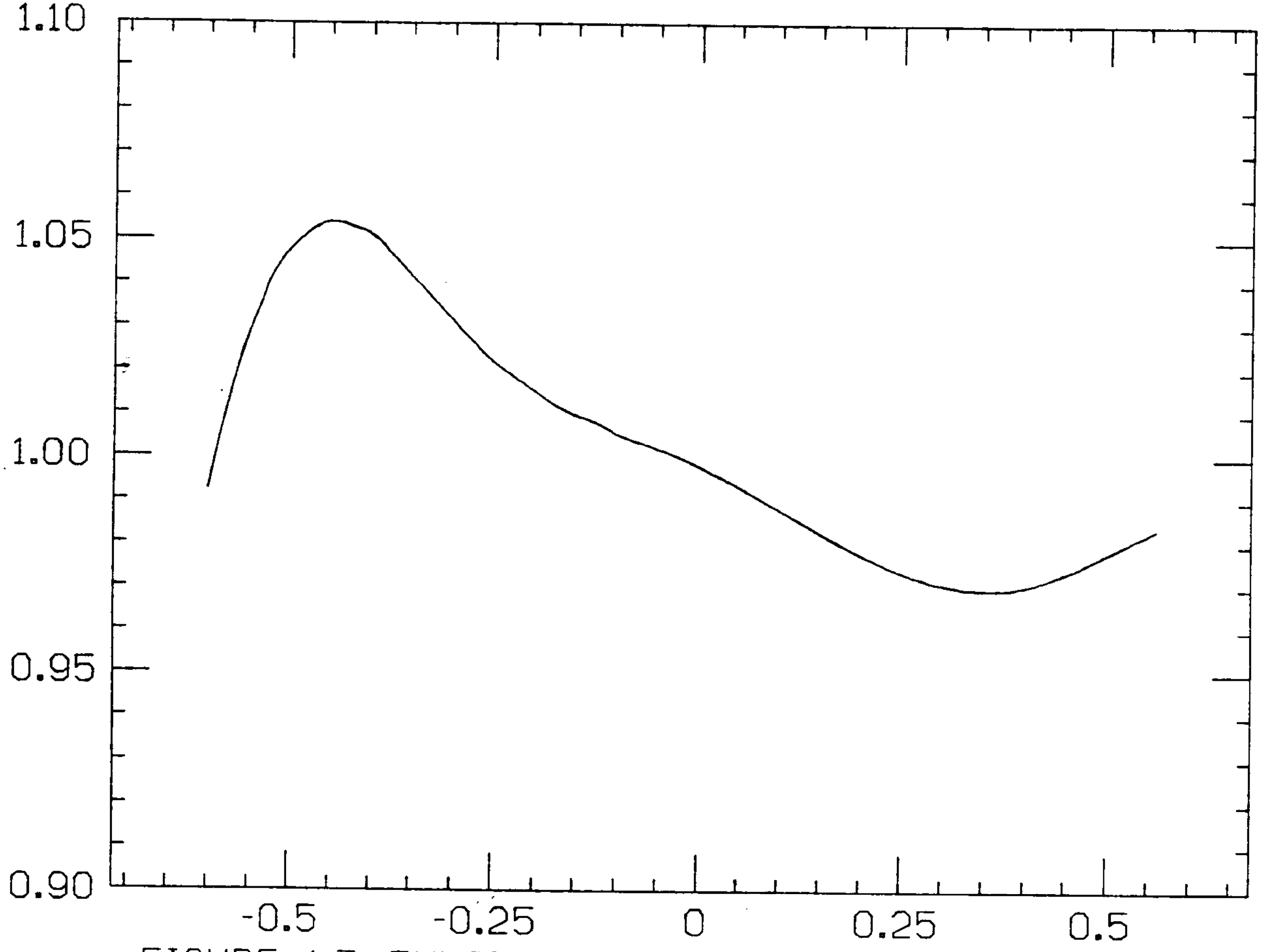

FIGURE 4.7. FUNCTIØN FØR SECØND PRØJECTIØN.
EXAMPLE 2, CHAPTER 4 (SEMICØNDUCTØRS)
PRØJECTIØN = X7

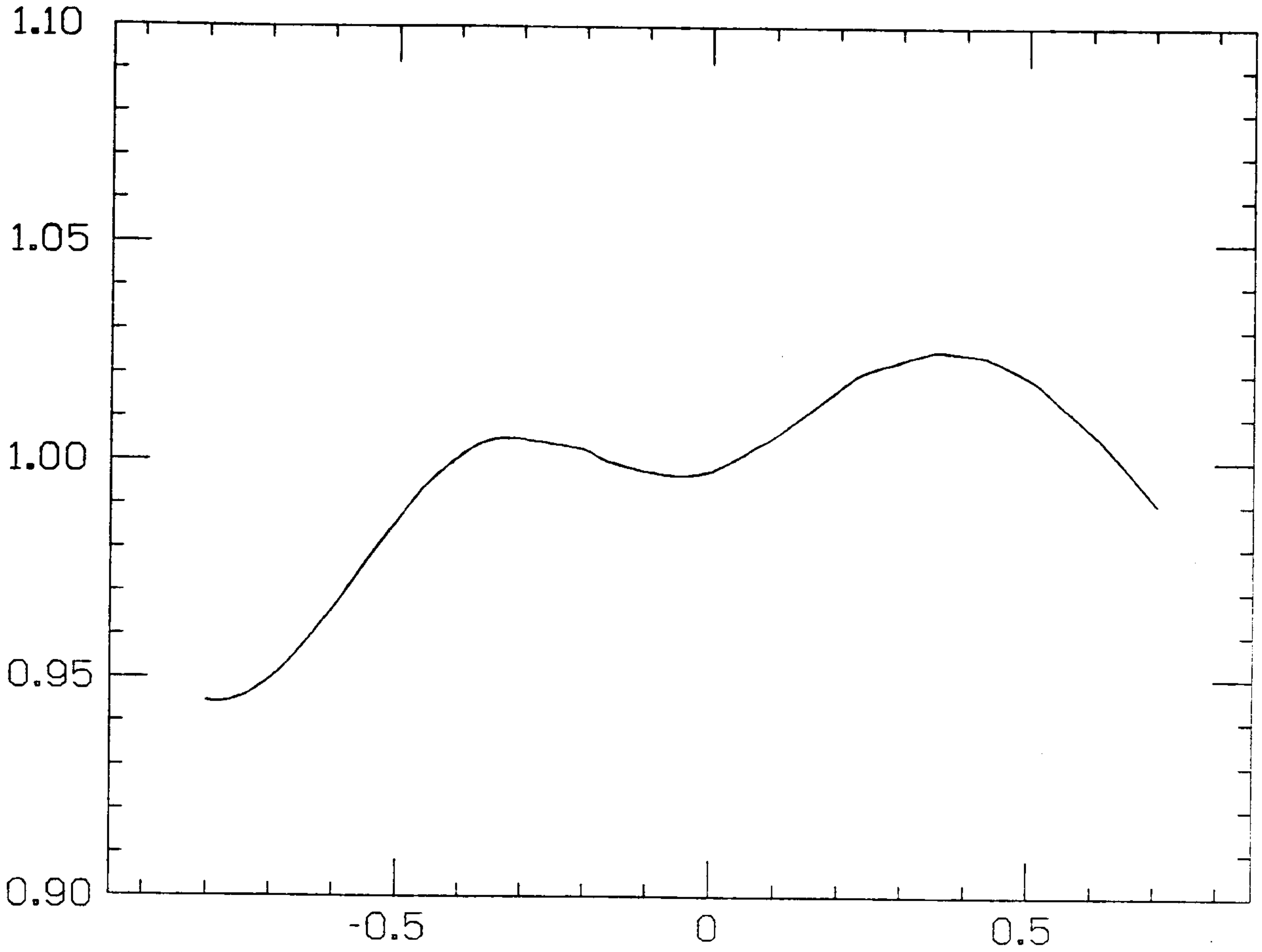

FIGURE 4.8. FUNCTIØN FØR THIRD PRØJECTIØN.
EXAMPLE 2, CHAPTER 4 (SEMICØNDUCTØRS)
PRØJECTIØN = X3

APPENDIX
NUMERICAL OPTIMIZATION TECHNIQUES

In this appendix the numerical optimization techniques used in the regression and categorical regression are discussed. The method used is a version of the Davidon–Fletcher–Powell algorithm. This method is a quasi–Newton method and requires the first derivatives of the function being minimized. The procedure approximates second derivatives and uses the approximations as in a Newton optimization.

The method is iterative. Let $\mathbf{g}_i$ denote the gradient vector at the i^{th} step, and $\mathbf{G}_i$ the Hessian matrix at that step. If $\mathbf{G}_i$ were known, the Newton method would seek the optimal $\mathbf{a}$ by iteration, setting

$$\mathbf{a}_{i+1} = \mathbf{a}_i - s_i \mathbf{G}_i^{-1} \mathbf{g}_i, \tag{A.1}$$

where s_i is the step size at step i and $\mathbf{a}_i$ is the estimate of $\mathbf{a}$ at that step. Since $\mathbf{G}_i$ is not known, the method approximates $\mathbf{G}_i^{-1}$ by $\mathbf{H}_i$, where the approximations improve as the minimum is approached. Then

$$\mathbf{a}_{i+1} = \mathbf{a}_i - s_i \mathbf{H}_i \mathbf{g}_i. \tag{A.2}$$

Let the starting estimate of $\mathbf{G}_i^{-1}$ be $\mathbf{H}_0 = I_p$, the identity matrix. At each step the new estimate $\mathbf{a}_{i+1}$ is obtained using (A.2). If the difference between the criteria for $\mathbf{a}_{i+1}$ and $\mathbf{a}_i$ is below a threshhold, the procedure halts. Otherwise, new first derivatives $\mathbf{g}_{i+1}$ are calculated, and an approximation to $\mathbf{G}_{i+1}^{-1}$ is obtained as

$$\mathbf{H}_{i+1} = \mathbf{H}_i - \frac{s_i \mathbf{H}_i \mathbf{g}_i \mathbf{g}_i'}{\mathbf{g}_i' \mathbf{H}_i (\mathbf{g}_{i+1} - \mathbf{g}_i)} - \frac{\mathbf{H}_i (\mathbf{g}_{i+1} - \mathbf{g}_i)(\mathbf{g}_{i+1} - \mathbf{g}_i)' \mathbf{H}_i}{(\mathbf{g}_{i+1} - \mathbf{g}_i)' \mathbf{H}_i (\mathbf{g}_{i+1} - \mathbf{g}_i)}. \tag{A.3}$$

From these a new estimate $\mathbf{a}_{i+2}$ is obtained. This procedures repeats until convergence is reached. (Additional details can found in the book by Kennedy and Gentle(1980).)

For this method the first derivatives of the function to be minimized are required. They are not available in exact form, but can be approximated using a method similar to that proposed by Buja and Thisted (1982) for additive projection pursuit regression. For categorical regression, at the M^{th} step, the function to be minimized over $\mathbf{a}_M$ is

$$S(\mathbf{a}_M) = \frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N} v_n(I_{kn} - h_n f_{kM}(\mathbf{a}_M, \mathbf{a}'_M\mathbf{x}_n))^2, \qquad (A.4)$$

where $g(\mathbf{x}) = \Pi_{m=1}^{M-1} f_{km}(amx)$. (Another argument has been added to f_{kM} to emphasize its dependence on $\mathbf{a}_M$ directly as well as through $\mathbf{a}'_M\mathbf{x}_n$.) The derivative with respect to a_{Mi} is

$$\frac{\partial S}{\partial a_{Mi}} = -\frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N} 2h_n v_n(I_{kn} - h_n f_{kM}(\mathbf{a}'_M\mathbf{x}_n)) \left(\frac{\partial f_{kM}(\alpha, z)}{\partial \alpha_i}\bigg|_{\alpha=\mathbf{a}_M} + x_{ni}\frac{\partial f_{kM}(\mathbf{a}_M, z)}{\partial z}\bigg|_{z=\mathbf{a}'_M\mathbf{x}_n}\right). \qquad (A.5)$$

There is no obvious way of evaluating the term $\frac{\partial f_{kM}(\alpha,z)}{\partial \alpha_i}$, which is the change in the smooth at a fixed position z as the projection changes. This term will be ignored. Its effect will be taken to be negligible compared with that of the second term. The second term is the local slope of f_{kM}, which is the slope of the local line fit in the neighborhood of the point. So the i^{th} element of the gradient can be approximated by

$$\frac{\partial S}{\partial a_i} \approx -\frac{2}{K}\sum_{k=1}^{K}\sum_{n=1}^{N} h_n v_n x_{ni}(I_{kn} - \hat{p}_k(\mathbf{x}_n))\frac{\partial f_{kM}(\mathbf{a}_M, z)}{\partial z}\bigg|_{z=\mathbf{a}'_M\mathbf{x}_n}. \qquad (A.6)$$

These approximations are substituted for $\mathbf{g}_i$ in the procedure. The starting value for the iteration to find $\mathbf{a}_M$ is determined by calculating the criterion $S(\mathbf{a}_M)$ for each major axis $(0, \ldots, 0, 1, 0, \ldots, 0)$. The one with the lowest value is selected as the starting point.

For the multiplicative regression model, both the derivatives $\frac{\partial S}{\partial a_i}$ and $\frac{\partial S}{\partial c}$ are needed. Let $h_n = \Pi_{m=2}^{M-1} f_m(\mathbf{a}'_m\mathbf{x}_n)$. Then

$$S = \frac{1}{N}\sum_{n=1}^{N}(Y_n - c - (f_1 - c)h_n f_M(\mathbf{a}_M, c, \mathbf{a}'_M\mathbf{x}_n))^2. \qquad (A.7)$$

(The arguments of f_M have been expanded to emphasize the dependence on $\mathbf{a}_M$ and c.) The derivative with respect to a_i is

$$\frac{\partial S}{\partial a_i} = -\frac{2}{N}\sum_{n=1}^{N}(f_1 - c)h_n\Big(Y_n - c - (f_1 - c)\Big(h_n\Big)f_M(\mathbf{a}'_M\mathbf{x}_n)\Big)$$
$$\left(\frac{\partial f_M(\theta, c, \mathbf{a}'_M\mathbf{x}_n)}{\partial\theta}\bigg|_{\theta=\mathbf{a}_M} + \frac{\partial f_M(\mathbf{a}_M, c, z)}{\partial z}\bigg|_{z=\mathbf{a}'_M\mathbf{x}_n} x_{ni}\right). \quad (A.8)$$

Again, $\frac{\partial f_M(\theta,c,\mathbf{a}'_M\mathbf{x}_n)}{\partial\theta}$ is ignored, and $\frac{\partial f_M(\mathbf{a}_M,c,z)}{\partial z}\big|_{z=\mathbf{a}'_M\mathbf{x}_n}$ is the slope of the local regression line used in the local linear smoother. The derivative with respect to c is

$$\frac{\partial S}{\partial c} = -\frac{2}{N}\sum_{n=1}^{N}(Y_n - c - (f_1 - c)h_n f_M(\mathbf{a}'_M\mathbf{x}_n))$$
$$\left(1 - h_n\left(f_M(\mathbf{a}'_M\mathbf{x}_n) + c\frac{\partial f_M(\mathbf{a}_M, c, \mathbf{a}'_M\mathbf{x}_n)}{\partial c}\right)\right). \quad (A.9)$$

The factor $\frac{\partial f_M(\mathbf{a}_M,c,\mathbf{a}'_m\mathbf{x}_n)}{\partial c}$ is ignored due to computational difficulties. This leaves

$$\frac{\partial S}{\partial c} = -\frac{2}{N}\sum_{n=1}^{N}(Y_n - c - (f_1 - c)h_n f_M(\mathbf{a}'_M\mathbf{x}_n))(1 - h_n f_M(\mathbf{a}'_M\mathbf{x}_n)). \quad (A.10)$$

These approximations are used for the slopes in the gradient search. Starting values are obtained similarly to the categorical regression case. For the second projection, the criterion for each major axis is calculated with each of several selected values of c (for example, $\min Y_n - k(\textit{interquartile range of } Y)$). The combination that provides the smallest value of the criterion is selected as the starting point. For succeeding projections, the starting value of c is taken to be the value obtained at the preceding step. This value is used with each major axis to find the one that gives the smallest squared distance.

The initial implementation of projection pursuit procedures utilized a Rosenbrock optimization (Rosenbrock, 1960). Methods based on the gradient are considerably faster computationally. The methods described here reduce the computational time required by thirty to eighty percent.

References

Anderson, T.W. (1958). *An Introduction to Multivariate Analysis*, New York: Wiley.

Beumer-Browner, K. (1981). Principal Components and Regression Analysis on MOS Final Test Data. Unpublished report.

Cleveland, W.S. (1979). Robust Locally Weighted Regression and Smoothing of Scatterplots. *Journal of the American Statistical Association*, **74**, 828-836.

Cox, D.R. (1970). *Analysis of Binary Data*, London: Chapman and Hall.

Delury, G.E. (ed) (1973). *The 1973 World Almanac and Book of Facts*, New York: Newspaper Enterprise Association.

Friedman, J.H. and Stuetzle, W. (1980). Projection pursuit classification. Unpublished manuscript.

Friedman, J.H. and Stuetzle, W. (1981). Projection Pursuit Regression. *Journal of the American Statistical Association*, **76**, 817–823.

Friedman, J.H. and Stuetzle, W. (1982). Smoothing of scatterplots. Dept. of Statistics, Stanford University, Tech. Report ORION003.

Kennedy, W.J. and Gentle, J.E. (1980). *Statistical Computing*, New York: Marcel Dekker, Inc.

Press, S.J. and Wilson, S. (1978). Choosing between Logistic Regression and Discriminant Analysis. *Journal of the American Statistical Association*, **73**, 699–705.

Rosenbrock, H.H. (1960). An Automatic Method for Finding the Greatest or Least Value of a Function. *Computer Journal*, **3**, 175–184.

Switzer, P. (1970). Numerical classification. In *Geostatistics*, 31–43. Editor: D.F. Merriam. Plenum Press, New York.