Statistical Methods and Neural Network Approaches for Classification of Data from Multiple Sources

Jon Atli Benediktsson
Philip H. Swain

TR-EE 90-64
December 1990

Laboratory for Applications of
Remote Sensing
and

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

Statistical methods and neural network approaches for classification of data from multiple sources

1-425

Statistical Methods and Neural Network Approaches for Classification of Data from Multiple Sources

Jon Atli Benediktsson
Philip H. Swain

TR-EE 90-64
December 1990

Laboratory for Applications of
Remote Sensing
and

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

STATISTICAL METHODS AND NEURAL NETWORK APPROACHES FOR CLASSIFICATION OF DATA FROM MULTIPLE SOURCES

Jon Atli Benediktsson and Philip H. Swain

TR-EE 90-64
December 1990

Laboratory for Applications of Remote Sensing
and
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

This work was supported in part by NASA Grant No. NAGW-925 "Earth Observation Research - Using Multistage EOS-like Data" (Principal Investigators: David A. Landgrebe and Chris Johannsen)

ACKNOWLEDGMENTS

The authors would like to thank Dr. Okan K. Ersoy for his contributions to this research.

The Anderson River SAR/MSS data set was acquired, preprocessed, and loaned by the Canada Centre for Remote Sensing, Department of Energy Mines, and Resources, of the Government of Canada. The Colorado data set was originally acquired, preprocessed, and loaned by Dr. Roger Hoffer, who is now at Colorado State University. Access to both data sets is gratefully acknowledged.

The research was supported in part by the National Aeronautics and Space Administration through Grant No. NAGW-925.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

Table Page

Table Page

Table Page

Table Page

Table Page

Table Page

LIST OF FIGURES

ABSTRACT

Statistical methods for classification of data from multiple data sources (e.g., Landsat MSS data, radar data and topographic data) are investigated and compared to neural network models. A problem with using conventional multivariate statistical approaches for classification of data of multiple types is in general that a multivariate distribution cannot be assumed for the classes in the data sources. Another common problem with statistical classification methods is that the data sources are not equally reliable. This means that the data sources need to be weighted according to their reliability but most statistical classification methods do not have a mechanism for this. This research focuses on statistical methods which can overcome these problems: a method of statistical multisource analysis and consensus theory. Reliability measures for weighting the data sources in these methods are suggested and investigated. Secondly, this research focuses on neural network models. The neural networks are distribution-free since no prior knowledge of the statistical distribution of the data is needed. This is an obvious advantage over most statistical classification methods. The neural networks also automatically take care of the problem involving how much weight each data source should have. On the other hand, their training process is iterative and can take a very long time. Methods to speed up the training procedure are introduced and investigated. Experimental results of classification using both

neural network models and statistical methods are given, and the approaches are compared based on these results.

CHAPTER 1
INTRODUCTION

1.1 The Research Problem

Computerized information extraction from remotely sensed imagery has been applied successfully over the last two decades. The data used in the processing have mostly been multispectral data and the statistical pattern recognition (multivariate classification) methods are now widely known. Within the last decade advances in space and computer technologies have made it possible to amass large amounts of data about the Earth and its environment. The data are now more and more typically not only spectral data but include, for example, forest maps, ground cover maps, radar data and topographic information such as elevation and slope data. For this reason there may be available many kinds of data from different sources regarding the same scene. These are collectively called multisource data.

It is desirable to use all these data to extract more information and get higher accuracy in classification. However, the conventional multivariate classification methods cannot be used satisfactorily in processing multisource data. This is due to several reasons. One is that the multisource data cannot be modeled by a convenient multivariate statistical model since the data are multitype. They can for example be spectral data, elevation ranges and even non-numerical data such as ground cover classes or soil types. The data are also not necessarily in common units and therefore scaling problems may arise. Another problem with statistical classification methods is that the data sources may not be equally reliable. This means that the data sources need to be weighted according to their reliability, but most statistical classification methods do not have such a mechanism. This all implies that methods other than the conventional multivariate classification have to be used to classify multisource data.

1.2 Two Different Classification Approaches

Various heuristic and problem-specific methods have been proposed to classify multisource data. However, this report concentrates on developing more general methods which can be applied to classify any type of data. In this respect two approaches will be considered: a statistical approach and a neural network approach.

In the statistical case, general methods will be investigated: consensus theory and statistical multisource analysis. In particular, attention is focused on statistical multisource analysis by means of a method based on Bayesian classification theory which was proposed by Swain, Richards and Lee [1,2]. This method will be extended to take into account the relative reliabilities of the sources of data involved in the classification. This requires a way to characterize and quantify the reliability of a data source, which becomes important when the combination of information is being looked at. Methods to determine the reliabilities and to translate them into weights to be used in the classification process will be investigated.

Another important problem that needs to be worked on in statistical multisource analysis is how to model effectively non-Gaussian data. In general, the classes in the data sources cannot be assumed to be Gaussianly distributed. In this research, methods to model non-Gaussian data will be considered.

Neural network methods to classify multisource data will also be investigated. Neural network models have as an advantage over the statistical methods that they are distribution-free and thus no prior knowledge is needed about the statistical distributions of the classes in the data sources in order to apply these methods for classification. The neural network methods also take care of determining how much weight each data source should have in the classification. A set of weights describe the neural network, and these weights are computed in an iterative training procedure. On the other hand, neural network models can be very complex computationally, need a lot of training samples to be applied successfully, and their iterative training procedures usually are slow to converge. The time consumption of the training process can be a major problem in application of neural networks in classification of multisource remote sensing data. In this report methods to speed up the training in conventional neural networks will be discussed.

Neural network models have more difficulty than do statistical methods in classifying patterns which are not identical to one or more of the training patterns. The performance of the neural network models in classification is therefore more dependent on having representative training samples whereas the statistical approaches need to have an appropriate model of each class. In this report experimental results of classification using both neural network models and statistical methods will be given, and the approaches will be compared based on these results.

1.3 Report Organization

Statistical methods for multisource classification are addressed in Chapter 2. The two statistical methods focused on in this report can be cast in two different groups of pooling methods: the linear opinion pool and the logarithmic opinion pool. Both pooling methods are discussed in detail and several methods are suggested to weight the different data sources for these methods. Since non-Gaussian modeling is a very important part of designing a statistical multisource classifier, non-Gaussian modeling methods are also addressed in Chapter 2.

The neural network approach for multisource classification is discussed in Chapter 3. Both two-layer (input and output layers) and multi-layer (input, hidden and output layers) are considered. Methods to speed up the training of the neural networks are also discussed in Chapter 3.

Experimental results are given in Chapter 4. Three data sets were used in experiments. Two of them consisted of multisource remote sensing and geographic data; the third data set was very-high-dimensional multispectral data. Both the linear opinion pool and the statistical multisource classifier were used in experiments in conjunction with several non-Gaussian modeling methods. The minimum Euclidean distance and the maximum likelihood method for Gaussian data were also used when appropriate. Both two-layer and three-layer neural network models were used in experiments to classify the different data sets. The results of the different approaches in Chapter 4 are compared in terms of different sample sizes and dimensionalities of input data. The statistical and neural network approaches showed some striking differences. Conclusions based on the experimental results are drawn in Chapter 5 where directions for future research are also suggested.

CHAPTER 2
STATISTICAL METHODS

In this chapter statistical methods for classification of multisource data will be discussed. The chapter begins with a survey of previous approaches to the classification of multisource remote sensing and geographic data. Most of these approaches are problem-specific. General multisource classification methods are discussed in detail. These general methods are consensus theory and statistical multisource analysis. Most consensus theory and statistical multisource analysis methods need source-specific weights (reliability factors) to control the influence of the of the data sources. Methods to select the weights are introduced and discussed. Finally, approaches to model non-Gaussian data sources are addressed.

2.1 A Survey of Previous Work

Several statistical methods have been used in the past to classify multisource data. For instance, topographic data have been combined with remotely sensed data in land cover analysis. One such approach is to subdivide the data into subsets of the data sources and then analyze each subdivision as reported in Strahler et al. [3]. In this method the data are subdivided in such a way that variation within each subdivision is minimized or eliminated based on some of the subdividing variables. Other examples of similar methods can be found in Franklin et al. [4] and Jones et al. [5]

A second method is "ambiguity reduction," where the data are classified based on one or more of the data sources, the results from the classification are assessed, and other sources are then used in order to resolve the remaining ambiguities. The ambiguity reduction can be achieved by logical sorting methods. Hutchinson has used this method successfully [6]. A method related to ambiguity reduction is the layered classifier (tree classifier) applied by Hoffer et al. [7] This particular approach has the advantage that it treats the data sources separately but has the shortcoming that it is very dependent on the analyst's knowledge of the data. Also, as in ambiguity reduction, different groupings or orderings of the sources produce different results [8].

Still another method is supervised relaxation labeling derived by Richards et al. [9] in order to merge data from multiple sources. This method, like other relaxation methods, tries to develop consistency among a collection of observations by means of an iterative numerical "diffusion" process. So far this method has not been fully investigated on multiple sources, but its iterative nature makes it computationally very expensive.

None of the methods described above is a general approach to multisource classification and all of them depend heavily on the user. They all deal with the various sources of data independently. In contrast a fourth method is a general approach which does not deal with the data sources independently. This method is the stacked-vector approach, i.e., formation of an extended vector with components from all of the data sources and handling the compound vector in the same manner as data from a single source. This method is the most straightforward and conceptually the simplest of the methods. It works very well if the data sources are similar and the relations between the variables are easily modeled [10]. However, the method is not applicable when the various sources cannot be described by a common model, e.g., the multivariate Gaussian model. Another drawback is that when the multivariate Gaussian model is used, the computational cost grows as the square of the total number of variables, which becomes prohibitive if the total number of variables is large.

All of the methods discussed up to this point have significant limitations as general approaches for multisource classification. Our goal is to develop a general method which can be used to classify complex data sets containing multispectral, topographic and other forms of geographic data. In this chapter consensus theoretic approaches are discussed, where the goal of consensus theory is to get a consensus among experts. In multisource classification the group of "experts" is the collection of data sources used in the classification. Related to consensus theory is a method of statistical multisource analysis, a probabilistic method based on Bayesian decision theory which was developed by Swain, Richards and Lee [1,2]. The method of statistical multisource analysis will be augmented to include mechanisms to weight the influence of the data sources in the classification. Two other important additions to the method will also be addressed: 1) how to select the weights for the data sources and 2) classification of non-Gaussian data.

2.2 Consensus Theoretic Approaches

Here we consider the formulation of the problem of combining expert opinions in which each expert (data source) estimates the probability of certain events in a particular σ-field [11]. The goal is to produce a single probability distribution which summarizes the various estimates with the assumption that the experts are Bayesian. The study of such combination procedures is called consensus theory.

French [12] has stated the following three reasons why a summarized opinion is needed:

i) *The expert problem*: The group of experts has been asked for advice by a decision maker. The decision maker is outside the group.

ii) *The group decision problem*: The group itself may be jointly responsible for a decision.

iii) *The text-book problem*: The members of the group may simply be required to give their opinions for others to use at some time in the future in as yet undefined circumstances. There is no predefined decision problem.

In the following discussion we will concentrate on the *expert problem* since we are interested in getting the information from the experts (data sources) and acting as the decision maker outside the group.

2.2.1 Linear Opinion Pools

Here the combination of probability density functions is discussed without any assumptions concerning their form. The combination formula is called a ***consensus rule.*** In his work McConway [13] shows that if the consensus rules are required to have too many pre-specified properties then flexibility in the combination is lost, as discussed below.

Consider the case where there is a possibly infinite set Ω with a number of elements at least greater than or equal to 3 and a collection of consensus rules for n data sources that depend only on the σ-algebra [11] of events considered, i.e., for each σ-algebra S of Ω there is a function C_S (a consensus rule):

$$C_S:[P(\Omega,S)]^n \rightarrow P(\Omega,S) \tag{2.1}$$

where $P(\Omega,S)$ is the space of all probability measures with σ-algebra S. This implies that if the data sources have probability measures $p_1,...,p_n$ then $C_S(p_1,...,p_n)$ is a new probability measure on the same σ-algebra of events. Now if T is any sub-σ-algebra of S then the $p_1,...,p_n$ can be restricted to T, namely

$$(p_i \mid T)(X) = p_i(X) \qquad X \in T \tag{2.2}$$

One property McConway lists as desirable for a consensus rule is the property of marginalization (MP), which is stated as follows:

$$C_S((p_1,...,p_n) \mid T) = C_T((p_1 \mid T),...,(p_n \mid T)) \tag{2.3}$$

This says that for events in T, the rules C_S and C_T coincide.

Another reasonable property for a consensus rule is the null set property (NSP), i.e., if an event is considered impossible by all the sources then its assigned probability is zero:

$$p_1(X) = \cdots = p_n(X) = 0 \rightarrow C_S(p_1,\ldots,p_n)(X) = 0 \qquad (2.4)$$

Two other properties (constraints) that could be considered are the following. One property is that the consensus depends just on the event and the values of the assessment of the sources (weak setwise function property (WSFP)):

$$C_S(p_1,\ldots,p_n)(X) = F(X,p_1(X),\ldots,p_n(X)) \qquad (2.5)$$

where F: $Q \rightarrow [0,1]$ $(Q = \{(2^{\Omega} - \{\phi,\Omega\}) \text{ x } [0,1]^n\} \cup \{(\phi,0,\ldots,0),(\Omega,1,\ldots,1)\})$, $F(\phi,\ldots)=0$, and $F(\Omega,\ldots)=1$. A stronger restriction is that the consensus depends only on the values of the assessment of the sources (strong setwise function property (SSFP)):

$$C_S(p_1,\ldots,p_n)(X) = G(p_1(X),\ldots,p_n(X)) \qquad (2.6)$$

where G: $[0,1]^n \rightarrow [0,1]$, $G(0,0,\ldots,0)=0$ and $G(1,1,\ldots,1)=1$. (SSFP is also called "strong label neutrality" by Wagner [14] and "context-free assumption" by Bordley and Wolff [15].)

McConway [13,16] investigated the relationship between the properties above and proved the results in Theorem 2.1 [17]:

Theorem 2.1: Suppose there is a family of consensus rules $\{C_S\}$ in Ω. Then

(a) MP is equivalent to WSFP

(b) (MP and NSP) is equivalent to SSFP

(c) SSFP is achieved if and only if there exist nonnegative numbers (weights)

$\alpha_1, \ldots, \alpha_n, \sum_i \alpha_i = 1$ such that for all σ-algebras S, with $X \in S$, and all $p_i \in P(\Omega,S)$ then

$$C_S(p_1,\ldots,p_n)(X) = \sum_{i=1}^{n} \alpha_i p_i(X) \tag{2.7}$$

The sum on the right side of equation (2.7) is called a *linear opinion pool.* The linear opinion pool is probably the most commonly used consensus rule. Its origins date back at least to Laplace [12]. Stone [18] seems to be have been the first to discuss this rule in some detail and he named it the opinion pool.

Part (c) of Theorem 2.1 shows the consequence of imposing too many conditions on the consensus rules. That is, if the SSFP property is imposed then the linear opinion pool becomes the combination function. A very important point here is that the MP and the NSP are not only imposed but also that the consensus rules are defined for all σ-algebras which implies a probability measure is achieved [17 .

The linear opinion pool has a number of appealing properties: It is simple, it yields a probability distribution (or a probability density if densities are used), it has the MP and the NSP, and its weights α_i reflect in some way the relative expertise of the ith expert. Also, if the data sources have absolutely continuous probability distributions, the linear opinion pool gives an absolutely continuous distribution. However, it also has several shortcomings. First of all the linear opinion pool is not externally Bayesian, i.e., the decision maker will not be Bayesian. The reason for this lack of external Bayesianity is that the linear opinion pool is not derived from the joint probabilities using Bayes' rule. Second, Dalkey [19] with the *impossibility theorem* has shown that by imposing not only the SSFP but also requiring the consensus rule to hold for conditional probabilities (($C(\omega_j|X) = C(\omega_j,X)/C(X)$ where ω_j and X are events), then a "dictatorship" results, which implies that only one of the experts (sources) counts. A simple example shows the dictatorship for a two expert problem [20]. If both the SSFP and the conditional probability rule hold, then

$$C(\omega_j|X) = \frac{C(\omega_j,X)}{C(X)} \tag{2.8}$$

Also, by applying equation (2.7), the equation for the conditional linear opinion pool becomes:

$$C(\omega_j|X) = \alpha p_1(\omega_j|X) + (1-\alpha)p_2(\omega_j|X) \tag{2.9}$$

By using elementary arguments on equations (2.8) and (2.9) the following equation is derived:

$$0 = \alpha(1-\alpha)[p_1(\omega_j|X) - p_2(\omega_j|X)][p_2(X) - p_1(X)]$$

where it is clear that the only acceptable alternatives for α are $\alpha = 0$ or $\alpha = 1$ if the domain for C is not limited. To avoid this dictatorship and be able nevertheless to apply some Bayesian updating, it is necessary to limit the possible probability density functions and the consensus rules considered.

2.2.2 Choice of Weights for Linear Opinion Pools

If a linear opinion pool is used as the consensus rule, the problem is how to select the weights assigned to each data source. There is no clear cut method of doing this. A few approaches considered in consensus theory are discussed below.

Winkler [21] suggested four ways of assessing weights:

1. *Equal weights.* $\alpha_i = 1/n$, $i = 1,2,...,n$. In this case the decision maker has no knowledge to allow him to believe that one source is more reliable than another. Therefore, the decision maker is willing to assign equal weights, which implies taking the average of the probability density functions.

2. *Weights proportional to a ranking.* Rank the sources from 1 to n according to "goodness," where a higher rank indicates a source is a "better" assessor. Then assign weight $r/\sum_{r=1}^{n} r$ to the source with rank r (r $= 1,2,...,n$). This rule presumes that the decision maker feels that the sources can be meaningfully ranked. It is used below in statistical multisource analysis.

3. *Weights proportional to a self-rating.* Have each source rate itself on a scale from 1 to c, where c is the highest rating and 1 the lowest. Then assign each source a weight proportional to its self-rating [21,22]. The rationale behind this rule is that a source may act as an expert in a certain area, but its expertise may vary from one area to another and one ground-cover to another.

4. *Weights based on some comparison of previously assessed distributions with actual outcomes.* "Scoring rules" [13,21,23] can be used to make the comparisons to apply this method successfully. A scoring rule is a function on the real line. Scoring rules involve the computations of a score according to a scoring rule which is designed to lead the assessor to reveal his true beliefs. The scoring rules can be thought of in the sense

that each assessor should attempt to maximize his expected score. The idea on which the theory of scoring rules is based is that, if an assessor (data source) indicates that his distribution for $X \in \{X_1, X_2, ..., X_N\}$ is G( ), and it is then observed that $X = X_k$, the assessor gets a score $S(X_k;G(\ ))$. A special case of scoring rules, called *strictly proper* scoring rules, promotes "honest" probability assessment in the sense that if the assessor wants to maximize his expected score, and his true distribution is G( ), he will actually state that his distribution is G( ) [13]. Three proper scoring rules are the following:

i) Quadratic score [13,23]:

$$S(X_k, G(\ )) = 2G(X_k) - \sum_{l=1}^{N}[G(X_l)]^2$$

ii) Spherical score [13]:

$$S(X_k, G(\ )) = \frac{G(X_k)}{\sum_{l=1}^{N}[G(X_l)]^2}$$

iii) Logarithmic score [13]:

$$S(X_k, G(\ )) = \log G(X_k)$$

It is intuitive that the scoring rules above measure the "goodness" of the probability assessments. Winkler [24] shows that they measure normative[1] and substantive[2] goodness simultaneously. McConway [13] proves that they

[1] An assessor is normatively good if he obeys closely the subjectivist postulates of coherence and produces assessment which corresponds closely to his "best judgements."

[2] An assessor is substantively good if he knows a lot about the background and details of the problem in which he is making an assessment.

measure *predictive* goodness also. The predictive goodness indicates that the assessors which give high probability to later observed data will get high scores. An example of weight revision using scoring rules is given later in this section.

Still another possible method of choosing weights is Bayesian weight revision which is based on previously assessed distributions and described in detail in [13]. Whatever the initial weights α_i are in a linear opinion pool, the consensus for the event ω_j is

$$C(\omega_j) = \sum_{i=1}^{n} \alpha_i p_i(\omega_j) \tag{2.10}$$

The weights can be revised through what McConway calls *Bayesian weight revision* if *all* the sources find out that an event X is true, assuming that C satisfies

$$C(\omega_j \mid X) = \frac{C(\omega_j, X)}{C(X)} \tag{2.11}$$

If the event X has occurred then:

$$C(X) = \sum_{i=1}^{n} \alpha_i p_i(X) \tag{2.12}$$

and

$$C(\omega_j, X) = \sum_{i=1}^{n} \alpha_i p_i(\omega_j \mid X) p_i(X) \tag{2.13}$$

Thus the consensus probability of ω_j given that X has occurred is

$$C(\omega_j \mid X) = \frac{C(\omega_j, X)}{C(X)} = \sum_{i=1}^{n} \frac{\alpha_i p_i(\omega_j \mid X) p_i(X)}{\sum_{k=1}^{n} \alpha_k p_k(X)}$$

$$= \sum_{i=1}^{n} \left(\frac{\alpha_i p_i(X)}{\sum\limits_{k=1}^{n} \alpha_k p_k(X)} \right) p_i(\omega_j \mid X) \qquad (2.14)$$

(provided that there exists i with $p_i(X) > 0$). That is, $C(\omega_j | X)$ is a weighted average of the $p_i(\omega_j \mid X)$'s with weights $\beta_1, \ldots, \beta_n$ (the revised weights) given by

$$\beta_i = \frac{\alpha_i p_i(X)}{\sum\limits_{j=1}^{n} \alpha_j p_j(X)} \qquad i = 1,...,n \qquad (2.15)$$

and the new weights β_i are proportional to $\alpha_i p_i(X)$. If there is a sequence of updatings, it is possible to proceed in this manner or use a scoring rule as mentioned above if that reflects the goodness of the fit of the source. Nevertheless the final weights are dependent on the initial weights. The initial score could be chosen by giving all the sources the same weight (or by some of the other weight selection schemes suggested by Winkler [20]) and then having a "trial run" and updating them by the rules discussed above. McConway [13] also extends this rule to the cases were only *some* of the sources agree that a certain event has occurred. He calls that revision method a ***generalized Bayesian revision.***

The Bayesian revision approach can be used in processing multisource remote sensing data since equation (2.14) can be applied as a global membership function with the preassessed density functions $p_i(\omega_j \mid X)$ for each source i. The weights α_i can then be updated by making a run through the training data because each training sample is a true event (ω_j, X) where ω_j is the information class and X is the observation vector, using the language above. The main problem this approach has is dictatorship. Bayesian weight revision can lead to dictatorship for one source according to the impossibility theorem [19] because this weight revision scheme extends the consensus rule to obey Bayes' rule. The dictatorship for such an extension was evident in the short example in equation (2.9). Different consensus rules might be needed to compute $C(\omega_j,X)$ and $C(X)$ in order to avoid dictatorship in Bayesian weight revision.

McConway [13] also describes a method of using scoring rules for weight revision: Let us assume that we have n data sources and before any data are observed their distributions are combined using a linear opinion pool with initial weights $\alpha_1,\alpha_2,...,\alpha_n$. The data are then observed from $X \in \{X_1,...,X_N\}$. Each source gives a distribution G_i for X. Now if $X = X_k$ is observed, a revised set of weights is computed using a strictly proper scoring rule S. The range for S is non-negative and it gives the score $S(X_k,G_i(\ ))$ to each source. The revised weight of the i-th data source, α'_i, is then proportional to $\alpha_i S(X_k,G_i(\ ))$ where $\sum_{i=1}^{n}\alpha'_i=1$.

The relationship between scoring rule weight revision and Bayesian weight revision is the following: Bayesian weight revision can be formalized as scoring rule weight revision with:

$$S(X_k,G_i(\ )) = g_i(X_k) \tag{2.16}$$

where $g_i(X)$ is the density corresponding to the distribution $G_i(\ )$. Therefore, Bayesian weight revision is a special case of scoring rule weight revision. The scoring rule weight revision has an advantage over Bayesian weight revision in the case when a *natural order* exists on X. Then an account of closeness of the assessors' distribution to the true event can be taken using a scoring rule which is sensitive to distance. A scoring rule is said to be sensitive to distance if $S(X_k,G(\ )) > S(X_{k,}G'(\ ))$ whenever $X = X_k$ is the true event and $G'(\ )$ is in some sense *more distant* from the true event than $G(\ )$. However, the scoring rule weight revision also has a disadvantage, namely Bayes' rule does not apply in general. Anyhow, this approach can readily be applied for determining weights in multisource classification. Its success depends on the scoring rule used. Which scoring rule gives the best performance has to be determined empirically.

The final weight selection method mentioned in this section has been proposed by Bordley and Wolff [15]. They suggest selecting weights which minimize the variance of the consensus rule $C(\omega_j\,|X)$:

$$C(\omega_j\ |\,X) = \sum_{i=1}^{n}\alpha_i(\omega_j)p_i(\omega_j\ |\,X) \tag{2.17}$$

By their method, if the data sources are independent, the weights $\alpha_i(\omega_j)$ should be inversely proportional to the variance of the event (ω_j,X). This approach works for a single event but it has its shortcomings for multiple events, especially in decision problems where it is undesirable to let the weights depend on the events. That is undesirable in such problems because the weights could have too much influence in discrimination whereas probability modeling of the events should be most important in discrimination.

2.2.3 Linear Opinion Pools for Multisource Classification

In the consensus theoretic literature, the linear opinion pool rule is used to combine probability distributions. It is assumed that all the experts observe the event X. Therefore, equation (2.7) is simply a weighted average of the probability distributions (or densities) from all the experts and the result is a combined probability distribution. However, in this research the linear opinion pool is considered for decision theoretic purposes rather than simply probability modeling. In this application the event $X = [x_1, x_2, \ldots, x_n]$ is a compound vector consisting of observations from *all* the data sources. Since x_i is the observation from the i-th data source, we can write $p_i(X) = p(x_i)$ when the notation from equation (2.7) is used. Thus, in the decision theoretic case equation (2.7) is extended to:

$$C_S(p_1, p_2, \ldots, p_n)(X) = \sum_{i=1}^{n} \alpha_i p(x_i) \tag{2.18}$$

and more specifically in a decision problem:

$$C_j(\omega_j \,|\, X) = \sum_{i=1}^{n} \alpha_i p(\omega_j \,|\, x_i) \tag{2.19}$$

where $j = 1, \ldots, M$ are the indices for the information classes.

The condition of the weight-sum being 1 is not necessary in equation (2.19). Equation (2.19) does not need to yield a probability distribution but only give a maximum value to the desired class. By including the modifications above for the linear opinion pool, the theory discussed in Section 2.2.1 can be used in the multisource classification problem. Other consensus theoretic rules, discussed later in this chapter, can be extended towards decision theory in a similar way to equation (2.7), i.e., by using $p_i(X) = p(x_i)$.

The linear opinion pool, which is a very simple pooling method, has been discussed up to this point. The linear opinion pool has several weaknesses; e.g., it shows dictatorship when Bayes theorem is applied and it is not externally Bayesian. Another consensus rule, the logarithmic pool, has been proposed to overcome some of the problems with the linear opinion pool. The logarithmic opinion pool is discussed below.

2.2.4 The Logarithmic Opinion Pool

Some authors have discussed the logarithmic opinion pool:

$$C^*(p_1,...,p_n) = \frac{\prod_{i=1}^{n} p_i^{\alpha_i}}{\int \prod_{i=1}^{n} p_i^{\alpha_i} d\mu} \tag{2.20}$$

where $\alpha_1, \ldots, \alpha_n$ are weights such that the integral in the denominator of equation (2.20) is finite [25]. Often it is assumed that $\sum_{i=1}^{n} \alpha_i = 1$. Bacharach [26] attributes the logarithmic opinion pool to Peter Hammond. Winkler [21] has given the logarithmic opinion pool a *natural-conjugate* interpretation. Winkler [21] also showed that the logarithmic opinion pool differs from the linear opinion pool in that it is unimodal and less dispersed.

Genest et al. [27] have extended equation (2.20) by relaxing the SSFP condition to allow the combination function in equation (2.6) to change with the event X. They call the result the *generalized logarithmic opinion pool*:

$$C^*(p_1,...,p_n) = \frac{g\prod_{i=1}^{n} p_i^{\alpha_i}}{\int g \prod_{i=1}^{n} p_i^{\alpha_i} d\mu} \tag{2.21}$$

where g is some ***essentially bounded function*** [11] on the sample space Ω [25,27]. Genest et al. [25] suggest regarding g as a likelihood (the probability of observing the data conditionally). The weights are non-negative except when the underlying σ-field on Ω is finite.

The logarithmic opinion pool treats the data sources independently (data independence property). It has the NSP in a very dramatic way. Zeros in the logarithmic opinion pool are vetoes; i.e., if any expert assigns $p_i(\omega_j) = 0$, then $C^*(p_1, \ldots, p_n) = 0$. This dramatic version of NSP is a drawback if the density functions are not carefully estimated. The logarithmic opinion pool is externally Bayesian. The external Bayesianity makes it a desirable choice in multisource classification along with the data independence property.

The main problem with the logarithmic opinion pool is also evident for the linear opinion pool, i.e., how to select the weights. Only heuristic and ad hoc methods exist in the literature on how to determine the weights. The weights should reflect in some way the relative expertise of the sources. Some of the weight selection methods described above for the linear opinion pool could be used, but the weight selection for the logarithmic opinion pool is less intuitive because of the product form of the pool. Even though the logarithmic opinion pool overcomes some of the problems associated with its linear counterpart (dictatorship and no external Bayesianity), it has the slight drawback that it is mathematically more complicated.

Bordley [28] has derived a version of the logarithmic pool from the conditional probabilities. The derivation is as follows for the event ω_j and X $[x_1,\ldots,x_n]$:

$$p(\omega_j \mid X) = \frac{p(X \mid \omega_j)p(\omega_j)}{p(X \mid \omega_j)p(\omega_j) + p(X \mid \omega_j^c)p(\omega_j^c)}$$

where $\omega_j{}^c$ is the compliment set of ω_j. Also, from Bayes' rule:

$$p(x_i \mid \omega_j) = \frac{p(\omega_j \mid x_i)p(x_i)}{p(\omega_j)}$$

for each i. If the experts are independent then:

$$p(\omega_j \mid X) = \frac{\left[\prod_{i=1}^{n} \frac{p(\omega_j \mid x_i)p(x_i)}{p(\omega_j)}\right] p(\omega_j)}{\left[\prod_{i=1}^{n} \frac{p(\omega_j \mid x_i)p(x_i)}{p(\omega_j)}\right] p(\omega_j) + \left[\prod_{i=1}^{n} \frac{p(\omega_j{}^c \mid x_i)p(x_i)}{p(\omega_j{}^c)}\right] p(\omega_j{}^c)}$$

$$= \frac{\left[\prod_{i=1}^{n} \frac{p(\omega_j \mid x_i)}{p(\omega_j)}\right] p(\omega_j)}{\left[\prod_{i=1}^{n} \frac{p(\omega_j \mid x_i)}{p(\omega_j)}\right] p(\omega_j) + \left[\prod_{i=1}^{n} \frac{p(\omega_j{}^c \mid x_i)}{p(\omega_j{}^c)}\right] p(\omega_j{}^c)} \quad (2.22)$$

Bordley gives some interesting properties for equation (2.22):

1. If $p(\omega_j \mid x_i) > p(\omega_j)$ for all i, then $p(\omega_j \mid X)$ will always be greater than $\max_i p(\omega_j \mid x_i)$ (unless some $p(\omega_j \mid x_i) = 1$), i.e., if all the source-specific posterior probabilities for a class are greater than the prior probability for that class, then the posterior probability of the combined sources will be greater than the posterior probability for every source.

2. If $p(\omega_j \mid x_i) < p(\omega_j)$ for all i, then $p(\omega_j \mid X)$ will always be less than $\min_i p(\omega_j \mid x_i)$ (unless some $p(\omega_j \mid x_i) = 0$), i.e., if all the source-specific posterior probabilities for a class are less than the prior probability for that class, then the posterior probability of the combined sources will be less than the posterior probability for every source.

3. If expert i is ignorant, i.e., if $p(\omega_j \mid x_i) = p(\omega_j)$ his assessment does not say anything about whether ω_j will occur. This implies:

$$p(\omega_j \mid x_1,\ldots,x_n) = p(\omega_j \mid x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)$$

4. Equation (2.22) has the NSP.

5. One expert can nullify the impact of another expert.

6. The formula is associative.

7. Bordley's version of the logarithmic opinion pool is externally Bayesian. Since each expert is externally Bayesian the decision maker will be Bayesian.

8. The group probability, $p(\omega_j \mid X)$, is always "better" in terms of minimized mean squared error loss than for any individual. To show this is the case, an indicator function, I_{ω_j}, can be defined:

$$I_{\omega_j} = \begin{cases} 1 & \text{if } \omega_j \text{ occurs} \\ 0 & \text{if } \omega_j \text{ does not occur} \end{cases}$$

It is needed to minimize $(r - I_{\omega_j})^2$ which is minimized by the r that minimizes

$$\sum_X (r - I_{\omega_j} \mid x)^2 p(X)$$

The r which minimizes the equation above is $r = p(\omega_j \mid X)$ which shows that the group probability is "better" in terms of mean squared error loss than the probability for any individual source.

Another method which has similar characteristics to the Bordley approach was developed by Swain, Richards and Lee [1,2]. This method is discussed in the next section.

2.3 Statistical Multisource Analysis

The method proposed in [1,2] is a general method which extends well-known concepts used for classification of multispectral images involving a single data source. This method is similar to Bordley's version of the logarithmic opinion pool: the various data sources are handled independently and each data source can be characterized by any appropriate model. However, these methods were developed independently. Also, the Swain, Richards and Lee method was specificly developed for combination of multisource remote sensing and geographic data. The main concepts in the method of Swain, Richards and Lee are addressed below.

Assume there are n distinct data sources, each providing a measurement x_i $(i = 1,...,n)$ for each of the pixels of interest. If any of the sources is multidimensional, the corresponding x_i will be a measurement vector. Let there be M user-specified information classes in the scene (not necessarily a property of the data) denoted ω_j $(j = 1,...,M)$. The pixels are to be classified into these classes.

Each data source is at first considered separately. For a given source, an appropriate training procedure can be used to segment or classify the data into a set of classes that will characterize that source. For example clustering could be used for this purpose. The data types are assumed to be very general, e.g., both topographic and multispectral data. The source-specific classes or clusters are therefore referred to as data classes, since they are defined from relationships in a particular data space. The data classes are for instance spectral classes in the case of spectral data and topographic classes in the case of topographic data. In general there may not be a simple one-to-one relation between the user-desired information classes and the set of data classes available. It is one of the requirements of a multisource analytical procedure to devise a method by which inferences about information classes can be drawn from the collection of data classes.

The k-th data class from the i-th source is denoted by d_{ik} ($k = 1,2,\ldots, m_i$), where m_i is the number of data classes for source i. The measurement vectors are associated with data classes according to a set of data-specific membership functions, $f(d_{ik}|x_i)$. This means that for a given measurement from the i-th source, $f(d_{ik}|x_i)$ gives the strength of association of x_i with data class d_{ik} defined for that source.

The information classes ω_j are related to the data classes from a single source by means of a set of source-specific membership functions $f(\omega_j|d_{ik}(x_i))$, for all i, j, k, where $f(\omega_j|d_{ik}(x_i))$ is the strength of association of data class d_{ik} with information class ω_j, possibly influenced by the value of x_i. This expression is different from previous approaches for single source classification, where it is often assumed in the analysis that there is a unique correspondence between spectral and information classes, once prior probabilities have been determined.

A set of global membership functions is defined, that collect together the inferences concerning a single information class from all of the data sources (as represented by their data classes). The membership function F_j for class ω_j is of the general form:

$$F_j = F_j[f(\omega_j | d_{ik}(x_i)), \alpha_i] \qquad (k=1,2,\dots,m_i \quad i=1,2,\dots,n) \qquad (2.23)$$

where α_i is the quality or reliability factor of the i-th source and is defined to weight the various sources, reflecting the perceived or measured reliabilities of the various sources of data. This is very important because it may be known that all the sources are not equally reliable and therefore the analyst is allowed to take into account his confidence in the recommendation of each of the individual sources of data available.

Finally a pixel $X = [x_1,\dots,x_n]^T$ is classified according to the usual maximum selection rule, i.e., it is decided that X is in class ω^* for which

$$F^* = \max_j F_j \qquad (2.24)$$

Now the membership functions are defined specifically. The reliability factor α_i will be disregarded for now but it will be included in Section 2.3.1. From experience with Bayesian classification theory a natural choice for the global membership function is the joint-source posterior probabilities.

$$F_j(X) = p(\omega_j | X) = p(\omega_j | x_1, x_2, \dots, x_n) \qquad (2.25)$$

If the assumption is made that class conditional independence exists between the data sources, the global membership function may be written [1,2]:

$$F_j(X) = [p(\omega_j)]^{1-n} \prod_{i=1}^{n} p(\omega_j | x_i) \qquad (2.26)$$

It may be argued that class-conditional independence between two unrelated sources is unlikely and the independence assumption may therefore introduce errors. On the other hand there are mainly two reasons why use of the independence assumption is desirable in this case. First, it is clear that interactions between two data sources can be very complex and consequently hard to model. However, to make use of dependence between sources these interactions have to be modeled. Also, analysts are in most cases unable to model the dependence because of the complexity of the interactions. Secondly, there is a trade-off between taking dependence into account and the computational complexity of the classification procedure, i.e., taking dependence into account may impose an unrealistic burden on the computer resources available. Using this reasoning, the independence assumption is justified in the global membership function.

Now consider the individual source-specific membership functions which appear here explicitly as source-specific posterior probabilities. These can be expressed as:

$$p(\omega_j|x_i) = \sum_{k=1}^{m_i} p(\omega_j|d_{ik},x_i)p(d_{ik}|x_i) \tag{2.27}$$

where the source-specific membership functions appear explicitly as $p(\omega_j|d_{ik},x_i)$ and the data-specific membership functions as $p(d_{ik}|x_i)$. Another way to write equation (2.27) is:

$$p(\omega_j|x_i) = \sum_{k=1}^{m_i} p(x_i|\omega_{j,}\, d_{ik})p(d_{ik}|\omega_j)p(\omega_j)/p(x_i) \tag{2.28}$$

Implementation of the classification technique involves using either equation (2.27) or equation (2.28) to determine the posterior probabilities in equation

(2.26). Then equation (2.24) is used for the decision. Equations (2.27) and (2.28) just look at one source at a time. There the relation between the data vectors and the data classes and the information classes is seen explicitly, demonstrating the role of data classes as intermediaries. Equation (2.26) then aggregates the information from all the sources of data for each specific information class.

As seen above, statistical multisource analysis is an extension of single-source Bayesian classification. However, this method as presented by Swain, Richards and Lee [1,2] does not provide a mechanism to account for varying degrees of reliability. It is reasonable to assume that this problem can be overcome if reliability factors are associated with each source involved in the classification in a similar way to weights in the linear and logarithmic pools. For this reason a modified version of this method will be investigated by means of which reliability analysis is added to the classification process. The following discussion also applies for Bordley's version of the logarithmic pool, which does not have any weights associated with it.

2.3.1 Controlling the Influence of the Data Sources

We want to associate reliability factors with the sources in the global membership function discussed above, i.e., to express quantitatively our confidence in each source, and use the reliability factor for classification purposes. This is very important because it is desirable to increase the influence of the "more reliable" sources, i.e., the sources we have more confidence in, on the global membership function and consequently decrease the influence of the "less reliable" sources in order to improve the classification accuracy. The need for reliability factors becomes apparent by looking at equation (2.26) where the global membership function is a product of probabilities related to each source. Each probability has value in the interval from 0 to 1. If any one of them is near zero it will carry the value of the membership function close to zero and therefore downgrade drastically the contribution of information from other sources, even though the particular source involved may have little or no reliability.

From above it is clear that it is necessary to put weights (reliability factors) on the sources which will influence their contributions to classification. Since the global membership function is a product of probabilities this weight has to be involved in such a way that when the reliability of a source is low it must discount the influence of that source and when the reliability of a source is high it must give the source relatively high influence. One possible choice for this kind of analysis is to put reliability factors as exponents on the contribution from each source in the global membership function, i.e., to weight the sources as in the logarithmic pool in equation (2.20).

Let us now determine the contribution from a single source in the global membership function. The global membership function for n sources is shown in equation (2.24). If one source is added, the global membership function for n+1 sources could be written in the following form:

$$F_j(X) = [p(\omega_j)]^{-n} \prod_{i=1}^{n+1} p(\omega_j | x_i) \tag{2.29}$$

If equation (2.29) is divided by equation (2.26) we get the contribution from source number n+1 which is $p(\omega_j | x_{n+1})/p(\omega_j)$. This motivates us to rewrite equation (2.26) in the following form:

$$F_j(X) = p(\omega_j)\prod_{i=1}^{n}\{p(\omega_j|x_i)/p(\omega_j)\} \tag{2.30}$$

Now to control the influence of each source, reliability factors α_i are assigned as exponents on the contribution from each source. Therefore equation (2.30) with reliability factors is written as:

$$F_j(X) = p(\omega_j)\prod_{i=1}^{n}\{p(\omega_j|x_i)/p(\omega_j)\}^{\alpha_i} \tag{2.31a}$$

where the α_i's (i = 1,...,n) are selected in the interval [0,1] because of the following reasons. If source i is totally unreliable (α_i=0) it will not have any influence on equation (2.31a) because

$$\{p(\omega_j|x_i)/p(\omega_j)\}^0 = 1$$

regardless of the value of $p(\omega_j|x_i)$. And if source i has the highest reliability (α_i=1) then it will give a full contribution to equation (2.31a) because

$$\{p(\omega_j|x_i)/p(\omega_j)\}^1 = p(\omega_j|x_i)/p(\omega_j)$$

It is also worthwhile to note that this method of putting exponents on the probabilities does not change the decision for a single-source classification because the exponential function p^α is a monotonic function of p. Also, equation (2.31a) looks similar to a logarithmic opinion pool, especially Bordley's version [28]. The difference is that equation (2.31a) has variable weights where Bordley's method has equal weights. A schematic diagram of the classification process associated with equation (2.31a) is shown in Figure 2.1.

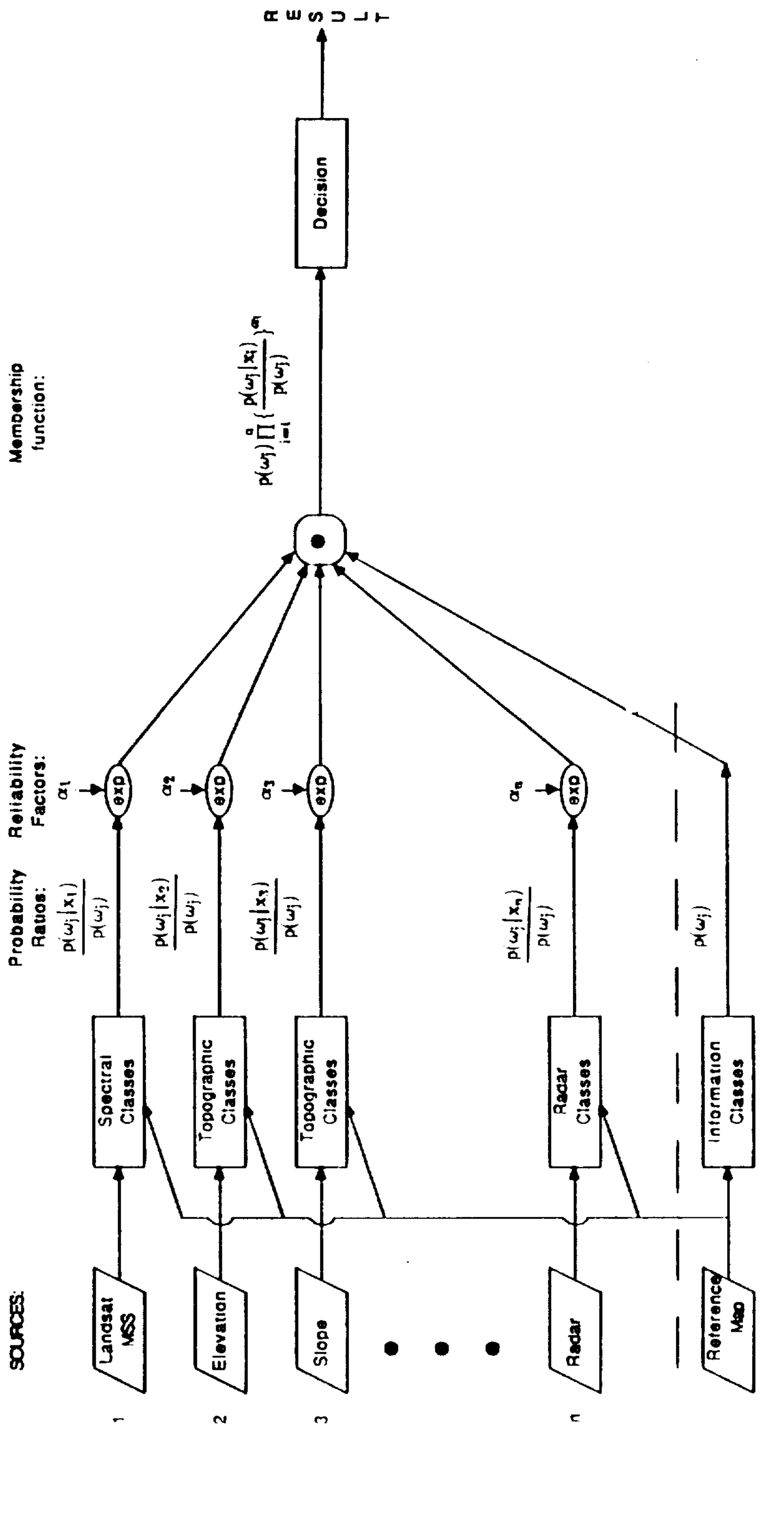


Figure 2.1 Schematic Diagram of Statistical Multisource Classifier

Equation (2.31a) can also be written in a logarithmic form as:

$$\log F_j(X) = \log p(\omega_j) + \sum_{i=1}^{n} \alpha_i \log \{p(\omega_j | x_i)/p(\omega_j)\} \tag{2.31b}$$

where the reliability factors are expressed as the coefficients in the sum. These coefficients control the influence of each source on the global membership function. If a coefficient is large compared to the other coefficients, the source it represents will have greater influence on the global membership function. If on the other hand a coefficient is low compared to other coefficients, it will decrease the influence of its source. Another way to see this is to look at the sensitivity of the global membership function to changes in one of the probability ratios. This can be expressed as:

$$\frac{\delta F_j(X)}{F_j(X)} = \alpha_i \frac{\delta p(\omega_j | x_i)/p(\omega_j)}{p(\omega_j | x_i)/p(\omega_j)} \tag{2.32}$$

which implies that the value of α_i will control the influence of source number i on the global membership function; a percentage change in the posterior probability leads to the same percentage change in the global membership function, multiplied by α_i.

The problem is to determine and quantify the reliability of the sources and to define the reliability factors, $\{\alpha_i\}$, based on the reliability of the sources. We think of a source as being reliable if its contribution to the combination of information from various sources is "good," i.e., if the classification accuracy is increased substantially or more information is extracted by using this particular source.

The process of determining the reliability factors is a two stage process. First the reliabilities of the sources have to be measured by some appropriate "reliability measure" and then the values of the reliability measures must be associated with the reliability factors in the global membership function.

2.3.2 Reliability Measures

Using the above understanding of a reliable source, three measures are proposed to determine the reliability of a source: weighted average separability, overall classification accuracy and equivocation. All of these measures are related to the classification accuracy of the source and can be considered to possess both normative and substantive goodness as defined for scoring rules. Also, the reliability measures are in some ways similar to scoring rules since they try to quantify the goodness of a data source. However, the reliability measures estimate how good the source is for classification in contrast to the scoring rules which only estimate the goodness of a specific probability distribution in a particular data source. To measure the goodness of the sources using the scoring rules, a weighted average of the goodness of class-specific probability distributions can be computed. Weighted average of the scoring rules can thus be used as a reliability measure.

a) Separability of Information Classes

We consider a source reliable if the separability of the information classes is high for the source. If on the other hand the separability of the information classes is low, the source is less reliable. Therefore one possibility for reliability evaluation is to use the average separability of the information classes in each source, e.g., average Bhattacharyya distance [29], average Jeffries-Matusita (JM) distance, average transformed divergence or any other separability function [30,31,32]. What kind of average is used depends on what we are after in the multisource classification. For instance if it is desired to improve the overall classification accuracy, the arithmetic average is used. If, however, we are concentrating just on specific classes, a weighted average separability of those information classes may be used. Calculation of separability involves computing volume integrals when the measurement space is multidimensional [30]. However, when the classes are assumed to have Gaussian probability density functions, the JM distance, the Bhattacharyya distance and the transformed divergence can be written as expressions involving the means and covariance matrices but no integrals. On the other hand, no similar expressions are available for non-Gaussian data. In multisource classification not all of the data sources can be modeled by the Gaussian model. To avoid computing volume integrals, the separability measure will only be used in our experiments when all the sources are Gaussian.

b) Classification Accuracy of a Data Source

Another way to measure reliability of a data source is to use the classification accuracy of the source. In this case a source is considered reliable if the classification accuracy for the source is high, but if the accuracy is low the source is considered unreliable. This approach is related to the method of using separability measures in that increased separability is consistent with higher accuracy. On the other hand there is no need estimate covariance matrices to compute the classification accuracy, so this approach is always applicable.

c) Equivocation

Still another way to characterize reliability of a source is to examine how strongly the data classes indicate information classes, i.e., by looking at the conditional probabilities that a specific information class is observed given a data class. All these conditional probabilities can be computed by comparing the reference map to a map of classification results produced from a data source.

Assuming there are M information classes $\{\omega_1,...,\omega_M\}$ and m data classes $\{d_1,...,d_m\}$, all the conditional probabilities can be used to form the m x M correspondence matrix R, where R is:

$$R = \begin{bmatrix} p(\omega_1|d_1) & p(\omega_2|d_1) & \cdots & p(\omega_M|d_1) \\ p(\omega_1|d_2) & p(\omega_2|d_2) & \cdots & p(\omega_M|d_2) \\ \vdots & \vdots & \cdots & \vdots \\ p(\omega_1|d_m) & p(\omega_2|d_m) & \cdots & p(\omega_M|d_m) \end{bmatrix} \tag{2.33}$$

Reliability can now be defined in the following way: If a source were optimal in reliability there would be a unique information class corresponding to each data class. Therefore ideally one conditional probability in each row of R would be 1 and all the others would be zero. If a source were very unreliable, there would be no correspondence between the data classes and the information classes; in the worst case all the probabilities in the matrix would be equal.

Now it is necessary to associate a number with the matrix R to characterize the reliability. Using information theoretic measures [33] the information classes can be thought of as transmitted signals and the data classes as received signals which must be used to estimate the transmitted signals. Using this approach it can be stated that there is an uncertainty of $\log[1/p(\omega_i|d_j)]$ about the information class ω_i when data class d_j is observed in a data source.

The average loss of information can be calculated when the data class d_j is observed, which is given by [33,34]:

$$H(\omega|d_j) = \sum_i p(\omega_i|d_j)\log\frac{1}{p(\omega_i|d_j)} \tag{2.34}$$

Now we want to average the information loss over all observed data classes d_j. This is the equivocation of ω with respect to d and is denoted by $H(\omega|d)$:

$$\begin{aligned} H(\omega|d) &= \sum_j p(d_j)H(\omega|d_j) \\ &= \sum_i\sum_j p(d_j)p(\omega_i|d_j)\{\log\frac{1}{p(\omega_i|d_j)}\} \\ &= \sum_i\sum_j p(\omega_i, d_j)\{\log\frac{1}{p(\omega_i|d_j)}\} \end{aligned} \tag{2.35}$$

$H(\omega|d)$ represents the average uncertainty about an information class over all the data classes. Evidently, $H(\omega|d)$ is the average loss of information per data class and therefore would seem to be a reasonable term to associate with the reliability of a source. Since $H(\omega|d)$ measures uncertainty, the lower the value it has the more reliable a source is. Therefore, the equivocation is called an uncertainty measure rather than a reliability measure. To be able to transform this uncertainty measure into a reliability factor, it first has to be mapped into a reliability measure and then associated with a reliability factor.

2.3.3 Association

The values of the reliability (uncertainty) measures must be associated with the reliability factors in order to improve the classification accuracy. It is worthwhile to note that we only want to include a source in the global membership function if the presence of that source improves the classification accuracy, i.e., we want the classification accuracy to be an increasing function of the number of sources. This is similar to feature selection but the difference here is that the sources (features) are not only selected but also the contribution of each source to the global membership function is quantified.

Using any of the measures discussed in Section 2.3.2 gives a specific value for each source. This value should be mapped into a reliability factor on the basis of our belief in the contribution of the source to the classification accuracy. The reliability (or uncertainty) measures take values in some particular interval and it is necessary to know the (functional) mapping between the values of the measures and the values of the reliability factors. In fact it is desirable to assign reliability factors to the sources in such a way as to improve the classification accuracy the most. It is very difficult to find an explicit association function between the values of the reliability and uncertainty measures on one hand and the reliability factors on the other. The measures can easily be used to rank the sources from "best" to "worst" but it is very difficult to determine the *optimal* value of the reliability factors. Ranking measures have previously been used in *consensus theory* for linear opinion pools as discussed in Section 2.2.2, whereas in contrast the global membership function in equation (2.31a) can be considered a ***logarithmic opinion pool*** problem. A possibility is to use optimization techniques to determine the reliability factors. That approach is discussed next.

2.3.4 Linear Programming Approach

The weight selection approaches described in Sections 2.3.2 and 2.3.3 are all relatively simple but somewhat ad hoc. In this section we describe an automatic method to determine the reliability factors of the sources. To accomplish this we apply linear programming to optimize the values of the global membership function using the training samples. From equation (2.31b) the global membership function in logarithmic form is:

$$\log F_j(X) = \log p(\omega_j) + \sum_{i=1}^{n} \alpha_i \log\{\frac{p(\omega_j \mid x_i)}{p(\omega_j)}\} \tag{2.38}$$

This equation must be optimized with respect to classification accuracy. Since training data are available, it is known for which classes the global membership function should be maximum for specific ground-cover elements. Therefore, optimizing equation (2.38) can be cast as a linear programming problem for *each* training sample selected. If there are M information classes, there will be M equations of the form (2.25) for each training sample.

The linear programming problem has the following form if a training sample from the class ω^* is selected and $q_{ji} = \log \{\frac{p(\omega_j \mid x_i)}{p(\omega_j)}\}$:

maximize:

$$\alpha_1 q_{*1} + \alpha_2 q_{*2} + \cdots + \alpha_n q_{*n} + \log p(\omega^*) = h$$

subject to the constraints:

$$\alpha_1 q_{11} + \alpha_2 q_{12} + \cdots + \alpha_n q_{1n} + \log p(\omega_1) \leq h$$

$$\vdots$$

$$\alpha_1 q_{M1} + \alpha_2 q_{M2} + \cdots + \alpha_n q_{Mn} + \log p(\omega_M) \leq h$$

$$\alpha_1 \geq 0,\ \ \alpha_2 \geq 0,\ \ , \ldots, \alpha_n \geq 0$$

Above, one equation out of the M equations of the form (2.38) is maximized, i.e., the equation corresponding to the class of the training sample. That leaves M-1 equations to be less or equal to the value (h) of that equation. The M x n matrix Q is known, where Q is

$$Q = \begin{bmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \vdots & \vdots \\ q_{M1} & \cdots & q_{Mn} \end{bmatrix} \tag{2.39}$$

To solve the linear programming problem it is necessary to get rid of the h variable on the right side of the inequalities in the constraints. That can be done simply by subtracting the objective function from each side in the inequalities. This gives the following linear programming problem:

maximize:

$$\alpha_1 q_{*1} + \alpha_2 q_{*2} + \cdots + \alpha_n q_{*n} + \log p(\omega^*) = h$$

subject to the constraints:

$$\alpha_1 (q_{11} - q_{*1}) + \alpha_2 (q_{12} - q_{*2}) + \cdots + \alpha_n (q_{1n} - q_{*n}) + \log \{p(\omega_1)/p(\omega^*)\} \leq 0$$

$$\vdots$$

$$\alpha_1(q_{M1}-q_{*1}) + \alpha_2(q_{M2}-q_{*2}) + \cdots + \alpha_n(q_{Mn}-q_{*n}) + \log\{p(\omega_M)/p(\omega^*)\} \leq 0$$

$$\alpha_1 \geq 0,\ \ \alpha_2 \geq 0,\ \ ,\ldots,\alpha_n \geq 0$$

where everything is known except the reliability factors $\alpha_1, \alpha_2, \ldots, \alpha_n$. If b training samples are selected from each information class ω_j, there will be Mb linear programming problems to solve like the one above. Solving all these linear programming problems gives us an interval estimate for each reliability factor:

$$l_i \leq \alpha_i \leq u_i$$

Using this interval estimate lower and upper bounds for each $\alpha_i q_{ji}$ in equation (2.38) can be computed and then:

$$\alpha_i q_{ij} \ \in \ [l_i q_{ji},\ u_i q_{ji}] \tag{2.40}$$

This leads to an interval estimate for log $F_j(X)$:

$$[\log F_j(X)_l,\ \log F_j(X)_u] =$$

$$[\log p(\omega_j) + \sum_{i=1}^{n} l_i q_{ji},\ \ \log p(\omega_j) + \sum_{i=1}^{n} u_i q_{ji}] \tag{2.41}$$

There will be M interval estimates of this kind for each pixel X. These interval estimates can be used for classification by applying the same decision methods as discussed in [2,35] in conjunction with Dempster-Shafer theory.

Using the optimization technique for the weights, the multisource classification algorithm takes the following form:

1) Train the classifier by using the sources independently.

2) Establish priors and posteriors.

3) Select training samples for computing reliability factors. Apply linear programming and determine intervals for each reliability factor.

4) Classify data using interval methods.

2.3.5 Non-Linear Programming Approach

The problem with the linear programming approach above is that it can give significantly different values of reliability factors for different information classes. Another idea to determine the weights in the global membership function is the following algorithm which uses gradient descent optimization as described below:

1. Select the initial values of the reliability factors by a reliability measure (classification accuracy, separability or equivocation). Select the gain factor η (a low value, e.g., 0.00001).

2. Use gradient descent in the following manner: Define the cost function

$$\mathrm{Cost}(X) = \sum_{j=1}^{N} F_{d(j)}(X) - F_{next(j)}(X) \tag{2.42}$$

where d(j) is the desired class for pixel X, next(j) is the class that has the highest value of the global membership function apart from d(j), and N is the number of training samples used. It is desired to maximize Cost(X) with respect to the weights (or minimize -Cost(X)). We take the gradient of equation (2.42) and the gradient descent equation for the (k+1)th pass follows:

$$\alpha_i(k+1) = \alpha_i(k) + \eta \nabla_{\alpha} \mathrm{Cost}(X) \tag{2.43}$$

where

$$\nabla_\alpha \mathrm{Cost}(X)=\log(p(\omega_d \mid x_i)/p(\omega_d))-\log(p(\omega_{next} \mid x_i)/p(\omega_{next}))$$

is the i-th element of the gradient vector.

3. Continue to update the weights by equation (2.43) until minimum error is reached.

By using equation (2.43) the condition that the weights should be in the interval from 0 to 1 is relaxed. The optimum weight values can be larger than 1 and some weights can become negative. The cost function in equation (2.42) is obviously linear and has no minimum value. A squared cost function is used in most applications of gradient descent optimization but such a function cannot be used here. A squared cost function would continue to decrease until the optimum values of 0 were given to all the weights. The approach in equation (2.42) is somewhat similar to the linear programming approach described in Section 2.3.4. However, equation (2.42) gives reliability factors to sources based on all the classes instead of individual classes.

2.3.6 Bordley's Log Odds Approach

Bordley [11,36] has derived a similar approach to the logarithmic opinion pool for log odds. In his log odds approach the i-th expert's odds on the event X are:

$$o_i(X) = \frac{p_i(X)}{1 - p_i(X)}$$

Let us now consider $\mathbf{o} = (o_1, o_2, \ldots, o_n)$ and let the odds after combination be:

$$o_D = \frac{p(X)}{1 - p(X)}$$

Then Bordley derives a log odds consensus rule of the form:

$$\log(\frac{o_D}{o_o}) = \sum_{i=1}^{n} \alpha_i \log(\frac{o_i}{o_o}) \qquad (2.44)$$

where α_i is the weight of the i-th expert and o_o is a constant which can be determined from fitting an *add.tive conjoint structure* [37] to a decision maker's subjective judgement [11]. By interpreting o_o as prior odds it can be seen that equation (2.44) is a log-odds version of the logarithmic opinion pool. Using that interpretation, equation (2.44) has both the same properties and shortcomings as the logarithmic opinion pool in equation (2.20).

2.3.7 Morris' Axiomatic Approach

Morris [38] has proposed an axiomatic approach to combine the probability judgements of experts. He begins by looking at a single expert which has a distribution $p_1(X)$ and assumes the decision maker has a prior $p(\omega_j)$. Morris then produces a consensus probability distribution C:

$$C(X) = \phi[p_1(X), p(\omega_j)] \qquad (2.45)$$

ϕ is called a processing rule which operates on two functions. Morris defines axioms which characterize desirable properties for the processing rule:

Axiom A:

The outcome should not depend on who observes a given piece of data if there is agreement on the likelihood function.

Axiom B:

A uniform prior of a calibrated expert is noninformative. (A calibrated expert is an expert which is good at encoding his beliefs as probabilities.)

Axiom C:

If the decision maker has a noninformative prior, he should adopt a calibrated expert's prior as his own.

Axiom A places a condition on the processing rule but does not determine it. By applying axiom A in conjunction with axiom B, the form of the processing rule can be completely determined. Axiom C is equivalent in effect to both axioms A and B together. The decision maker must also calibrate the experts' opinions. Sequential application of the axioms results in a multiplicative rule for multiple experts:

$$C(X) = k\ \mathrm{cal}(X)\ p_1(X) \cdots p_n(X)p(\omega_j) \tag{2.46}$$

where k is a normalization constant and cal(X) is a calibration function which is defined empirically. If the experts are all calibrated and independent, then cal(X) = 1.

Lindley [39] has argued that axiom A is unsatisfactory in the extreme case when the decision maker decides to ignore the opinion of an expert (the decision maker makes the outcome be equal to his own prior regardless of what the expert states). Schervish [40] has showed that the axioms are self-contradictory due to the concept of the processing rule (2.45). The issue of calibration is also very important in this approach. The decision maker must calibrate the experts' opinions. This demonstrates that the method is not truly Bayesian in spirit. But it is also worth noting that when the density functions can easily be estimated and the data sources are independent, Morris' axiomatic approach becomes a logarithmic opinion pool with equal weights. In the case of classification of multisource remote sensing and geographic data it can be assumed that the data sources are independent but not equally reliable. Therefore, the logarithmic opinion pool with *variable* weights is a more desirable choice for classification of such data.

2.4 Group Interaction Methods

All the consensus theoretic approaches described so far do not allow the experts to interact. DeGroot [41] has suggested a different approach for choosing weights in consensus theory which consists of giving the weights using the sources' own opinions of each other (group interaction). Although DeGroot's method can be effective in simple expert problems it is hard to implement the method for multisource remote sensing and geographic data, since it is difficult to let each data source evaluate the performance of the other sources in classification. However, the method has some parallels with the neural network methods discussed in Chapter 3. The neural networks use feedback to self-stabilize but are distribution-free. The DeGroot method will thus not be discussed further here.

2.5 The Super Bayesian Approach

Many Bayesians question all the consensus approaches discussed above and describe them as ad hoc. They also point out that expert weights do allow for some discrimination but in vague, somewhat ill-defined ways. They prefer a careful probabilistic modeling of the situation, combined with probabilistic processing. This means obtaining the joint distributions of all unknown parameters of interest. The approach, called the super (supra) Bayesian approach, is natural and is based on the assumption that all the expert opinions are data for the decision maker. Therefore, Bayes' rule should be used to update the belief of the decision maker [11,19,20,42,43,44]. The problem with this approach is that its implementation is very difficult because dependence between all the experts has to be modeled.

French [11,44] is one of the advocates of the super Bayesian approach. He has proposed the following log-odds approach for the event of interest ω_j: Let λ_i be the log-odds for the i-th expert when X is observed:

$$\lambda_i = \log(\frac{p_i(X)}{1 - p_i(X)})$$

Further let $\lambda = (\lambda_1,...,\lambda_n)^T$. French assumes that λ has a jointly normal distribution in the view of the super Bayesian. This density is conditional on ω_j and the super Bayesian's prior, $p(\omega_j)$. The log-odds of the super Bayesian's posterior, $p(\omega_j \,|\lambda\,)$, can be shown to be [19,44]:

$$\log(\frac{p(\omega_j \mid \lambda)}{(1 - p(\omega_j \mid \lambda)}) =$$

$$(m_{\omega_j} - m_{\omega_j^c})^T \Sigma^{-1} (\lambda - 0.5(m_{\omega_j} + m_{\omega_j^c})) + \log(\frac{p(\omega_j)}{(1 - p(\omega_j))})$$

where $m_{\omega_j} = E_p(\lambda \,|\omega_j)$, Σ is the covariance matrix for λ given the event ω_j and $\omega_j{}^c$ is the compliment of ω_j. By writing o_{io} as the antilogarithm of the i-th component of $0.5(m_{\omega_j} + m_{\omega_j^c})$ together with a little manipulation, the equation above can be written as:

$$\log(\frac{p(\omega_j \mid \lambda)}{(1 - p(\omega_j \mid \lambda)}) - \log(\frac{p(\omega_j)}{(1 - p(\omega_j))}) = \sum_{i=1}^{n} \beta_i \log(\frac{o_i}{o_{io}}) \qquad (2.47)$$

This equation is very similar (but not identical) to Bordley's log-odds approach. But it should be noted that this approach is completely equipped with weights as interpretable coefficients where β_i is a function of m_{ω_j}, $m_{\omega_j}{}^c$

and Σ. However, there is very little empirical evidence available to determine the super Bayesian's choice of the jointly normal distribution of λ. The dependence between the data sources has to be modeled and that problem is very difficult especially in classification of multisource remote sensing and geographic data. As noted earlier, we are usually either unable or unwilling to model this dependence. Therefore, the super Bayesian approach is in most cases not applicable to the research problem discussed here.

2.6 Overview of the Consensus Theoretic Approaches

The consensus theoretic approaches discussed above have different characteristics. The linear opinion pool is very simple and has several shortcomings, e.g., it is not externally Bayesian and the impossibility theorem limits its application because of source-specific dictatorship when Bayes' rule is used. The logarithmic opinion pool overcomes these shortcomings and will give unimodal consensus densities whereas the linear opinion pool gives multimodal consensus densities. In the experiments in Chapter 4, the linear opinion pool and the version of statistical multisource classifier introduced in Section 2.3.1 will be used. The reliability measures introduced in Section 2.3.2 will be used for selection of reliability factors in the experiments.

The statistical multisource classifier is a version of the logarithmic opinion pool. Both of the approaches proposed by Bordley are related to the statistical multisource classifier as discussed above. Neither the super Bayesian nor the group interaction methods will be used in experiments because of the implementation difficulty for these methods.

In order to apply the consensus theoretic approaches all the data sources have to be modeled by probability densities. Some data sources can be assumed to have Gaussian data classes. All of the other sources will be non-Gaussian and these sources need to be modeled by density estimation methods. Such methods are discussed in the next section.

2.7 Classification of Non-Gaussian Data

A very important part of designing a statistical multisource classifier is to handle the problem of modeling and classifying non-Gaussian data efficiently. Modeling of non-Gaussian data is a well established research field. In the following three sections the main approaches of modeling will be addressed. First a histogram approach is discussed. The histogram approach is the simplest way to model non-Gaussian data. Two more advanced methods are addressed: Parzen density estimation and the maximum penalized likelihood estimator. Several other approaches have been reviewed in the literature [45], e.g., nearest neighbor density estimation, density estimation using weight functions and orthogonal series estimators. For the research problem addressed here, the three methods discussed below should be sufficient.

2.7.1 Histogram Approach

The simplest way to model non-Gaussian data is to use the histograms of the training data. Here a fixed cells histogram approach [29,45] is described. In this method the data space is partitioned into mutually disjoint cells $\Gamma_1, \Gamma_2, \ldots, \Gamma_N$, whose volumes are equal. The density function is estimated by the proportion of samples which falls into each cell. When the data have been modeled by the histogram approach, they can be classified by, e.g., the maximum likelihood algorithm [45].

The histogram approach is distribution-free and, if regular meshes are used for the Γ's, the selection of cells is straightforward. However, one major disadvantage of this method is that it requires too much storage; for example, N^k cells for k variables with N sections for each variables. Therefore, most modifications which have been proposed are designed to reduce the number of cells. The variable cells method [29] is one such variant.

Although the histogram approach usually does a good job of modeling univariate data, it can be significantly improved upon in terms of accuracy by more advanced methods. It is also desirable to use more general methods which do a good job of modeling multivariate data. Parzen density estimation is one commonly used such method. Another method which improves upon the histogram approach for univariate data is the method of maximum penalized likelihood estimators.

2.7.2 Parzen Density Estimation

The Parzen density estimator with kernel K is defined by [29,45,46]:

$$\hat{p}(X) = \frac{1}{N\sigma^d} \sum_{i=1}^{N} K(\frac{X - X_i}{\sigma}) \tag{2.48}$$

where d is the dimensionality of the data and σ is the window width, also called the smoothing parameter. N is the number of training samples, X_i. The kernel K can be of any shape (rectangular, triangular, Gaussian, etc.) with the condition:

$$\int_{R^d} K(X)\, dX = 1 \tag{2.49}$$

If the kernel K is both everywhere non-negative and satisfies (2.49), then K is a density function. It follows from this that $\hat{p}(X)$ will be a probability density function and $\hat{p}(X)$ will also inherit all the continuity and differentiability properties of the kernel K.

The Parzen density estimator has been widely studied and applied. However, it suffers from a slight drawback when applied to data from long-tailed distributions [45]. The window width is fixed across the entire sample and this often leads to noise appearing in the tails of the estimates. Also, if the estimate is smoothed to avoid this problem, essential detail in the main lobe of the distribution can be lost. Apart from this drawback, the Parzen density estimator is a very desirable choice for modeling non-Gaussian data.

2.7.3 Maximum Penalized Likelihood Estimators

The maximum penalized likelihood estimator [45,47] computes a piecewise linear estimate of a one-dimensional density function for a given random sample of observations. This particular method tries to maximize the likelihood for a particular curve f. As pointed out in [45] it is not possible to use maximum likelihood estimation directly for density estimation without placing restrictions on the class of densities over which the likelihood is to be maximized. However, methods relating to the maximum likelihood can be used, e.g., by applying with the likelihood a term which quantifies the roughness of the curve f. The roughness term can be described by a functional R(f).

The penalized log-likelihood is now defined by:

$$l_\gamma(f) = \sum_{i=1}^{N} \log f(X_i) - \gamma R(f) \tag{2.50}$$

where γ is a positive smoothing parameter and N is the number of samples. The probability density function $\hat{p}$ is found by maximizing $l_\gamma(f)$ [45]. This approach is attractive since it relates curve estimation to density estimation. Also, the approach controls the balance between smoothness and goodness-of-fit. The roughness penalty predefines undesirable effects.

2.7.4 Discussion of Density Estimation Methods

Of the density estimation methods discussed here, the histogram approach is the most straight-forward. However, this method can be improved upon in terms of classification accuracy of test data. The histogram approach has in common with the maximum penalized likelihood method that these methods are most effective for univariate data. The maximum penalized likelihood estimation is attractive since it combines density estimation with curve fitting. Because of its smoothing properties this method should be more accurate in classification of test data than the histogram approach. The Parzen density method is a very well established density estimation method which can be used for multivariate density estimation. Also, Parzen density estimation should generalize better than the histogram method. However, the Parzen method has the drawback that it is very slow and this is a problem if the size of data to be classified is large. To explore the differences of these methods all three will be used in the experiments in Chapter 4.

CHAPTER 3
NEURAL NETWORK APPROACHES

Neural networks for classification of multisource data are addressed in this chapter. The chapter begins with a general discussion of neural networks used for pattern recognition, followed by a discussion of well-known neural network models and previous work on classification of remote sensing data using neural networks. Next "fast" neural network models are addressed in conjunction with classification of multisource remote sensing and geographic data. Finally, methods to implement statistics in neural networks are discussed.

3.1 Neural Network Methods for Pattern Recognition

A neural network is an interconnection of neurons, where a neuron can be described in the following way. A neuron has many (continuous-valued) input signals x_j, $j = 1,2,...,N$, which represent the activity at the input or the momentary frequency of neural impulses delivered by another neuron to this input [48]. In the simplest formal model of a neuron, the output value or the frequency of the neuron, o, is often approximated by a function

$$o = K\,\phi\Big(\sum_{j=1}^{N} w_j x_j - \theta\Big) \tag{3.1}$$

where K is a constant and ϕ is a nonlinear function which takes the value 1 for positive arguments and 0 (or -1) for negative arguments. The w_j are called *synaptic efficacies* [48] or weights, and θ is a threshold.

In the neural network approach to pattern recognition the neural network operates as a black box which receives a set of input vectors **x** (observed signals) and produces responses o_i from its output neurons i (i $=1,...,L$ where L depends on the number of information classes). A general idea followed in neural network theory is that the outputs are either $o_i = 1$, if neuron i is active for the current input vector **x**, or $o_i = 0$ (or -1) if it is inactive. This means the signal values are coded as binary vectors, and for a specific input vector x the outputs give a binary representation of its class number. The process is then to learn the weights through an adaptive (iterative) training procedure in which a set of training samples is presented to the input with some particular representation (see Figure 3.1). The network will give an output response to each sample. The actual output response is compared to the desired response for the input. The error between the desired output and the actual output is used to modify the weights in the neural network. The training procedure is ended when the network has stabilized, i.e., when the weights do not change from one iteration to the next iteration or change less than a threshold amount. Then the data are fed into the network to perform the classification, and the network provides at the output the class representation of a number for each pixel. A schematic diagram of a three-layer neural network classifier is shown in Figure 3.2.

Data representation is very important in application of neural network models. A straightforward coding approach used by most researchers is to

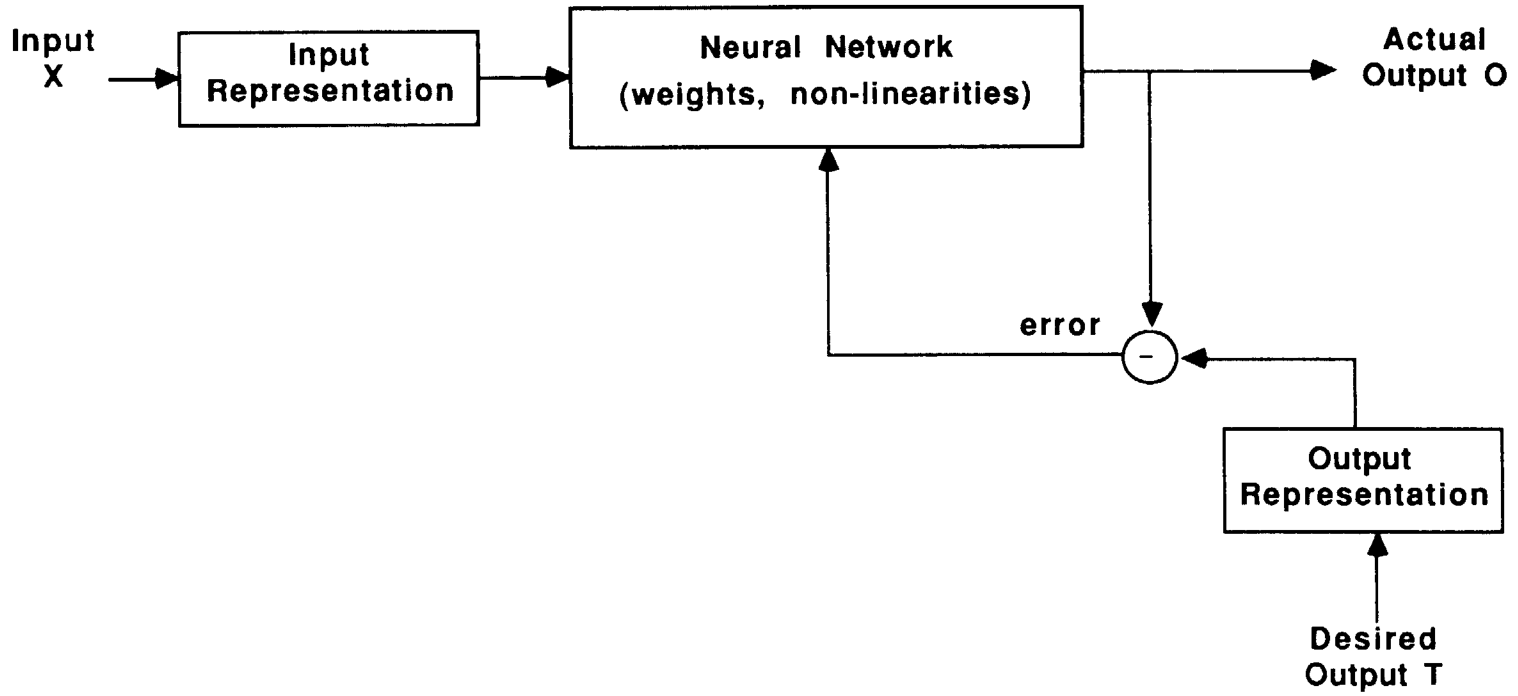


Figure 3.1 Schematic Diagram of Neural Network Training Procedure

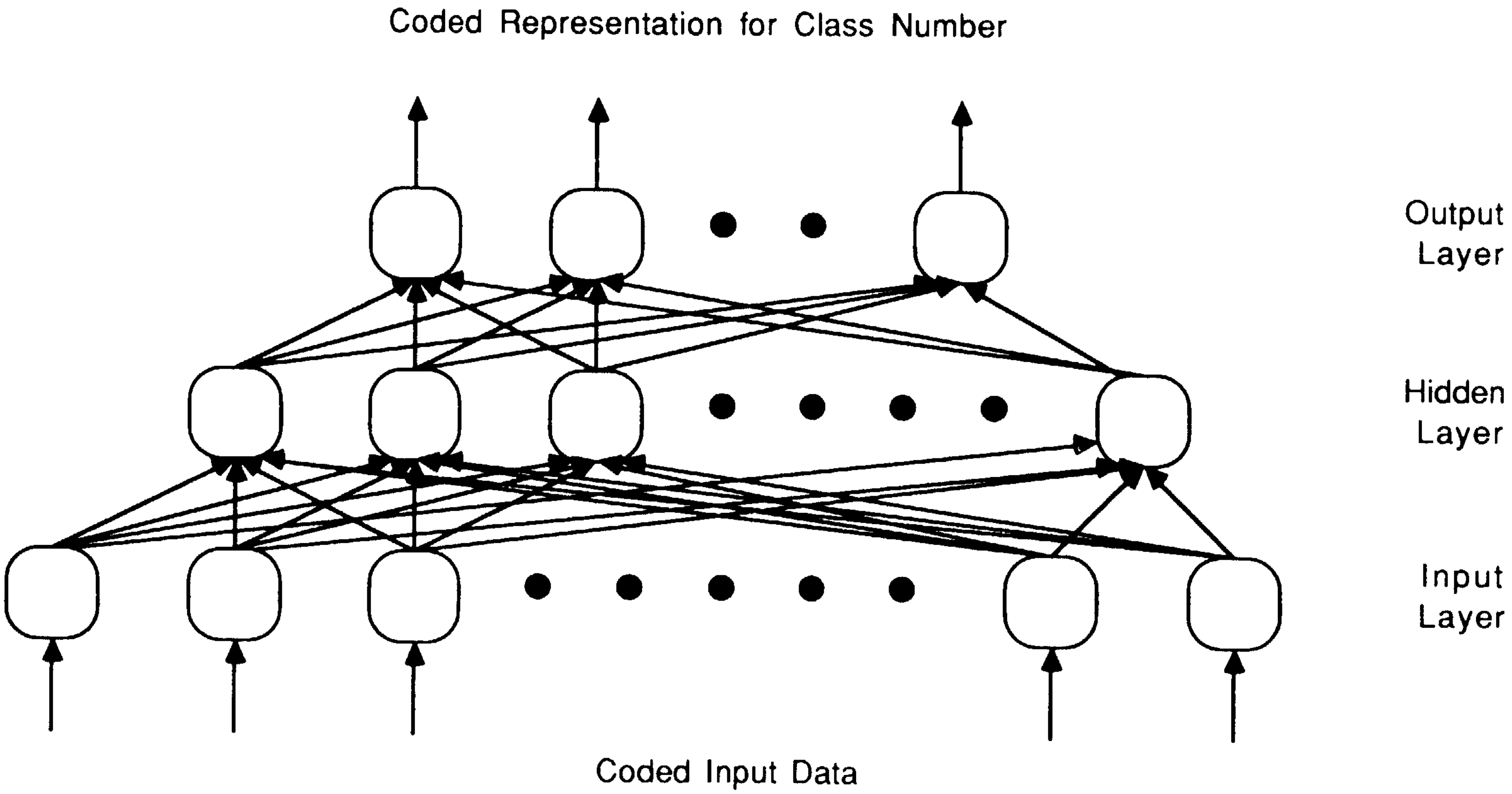


Figure 3.2 Schematic Diagram of Neural Network Used for Classification of Image Data

code the input and output by a binary coding scheme (0 = 00, 1 = 01, 2 = 10, etc.). However, in some respects for our application, it is more appropriate to use the Gray-code representation [49] of the input data. The Gray-code representation can be derived from the binary code representation in the following manner: If $b_1\ b_2\ \ldots\ b_n$ is a code word in an n - digit binary code, the corresponding Gray-code word $g_1\ g_2\ \ldots\ g_n$ is obtained by the rule:

$$g_1 = b_1$$

$$g_k = b_k \oplus b_{k-1} \qquad k \geq 2$$

where $\oplus$ is modulo-two addition [49]. The reason that the Gray-code representation is more appropriate than the binary code in our application is that adjacent integers in the Gray-code differ only by one digit. It can be assumed that adjacent data values in the code space are likely to belong to the same information class. When they belong to the same class, the use of the Gray-code leads to a smaller number of weight changes, since for values from a given class, most of the input digits are identical.

Representation at the output of the neural network is also important. If binary coding is used at the output, the number of output neurons can be reduced to $\lceil \log_2 M \rceil$ where M is the number of information classes. However, it is better to use more output neurons than the minimum $\lceil \log_2 M \rceil$ in order to make the neural network more accurate in classification. Even though adding more output neurons makes the network larger and therefore computationally more complex, it can also lead to fewer learning cycles, since the Hamming distance of the output representations of different classes can be larger. One such coding mechanism is "temperature coding," in which the representation for n has 1 in its first n digits and 0 in the rest (e.g., 4 = 1111000).

However, the most commonly used output representation is the following. The number of output neurons is selected the same as the number of classes and only one output neuron is active (has the value 1) for each class. As an example let us look at a four class problem where this approach is used. Then class #1 would be represented by 1000 and class #3 by 0010. This particular representation has the advantage in classification that only one neuron should be active (1) and all of the others should be inactive (0). Therefore, the "winner take all" principle can be used. In testing the neural network classifier the representation is better for the reason that an input sample can be classified to the class which has the largest output response. If other coding schemes were used for output representation, some samples might need to be rejected in testing since their output would not be close to any of the desired output representations. No such problem is evident with this representation. Therefore, this "winner take all" representation will be used in the experiments in Chapter 4. The Gray-code will used there for input representation.

3.2 Previous Work

Several neural network models have been proposed. Rosenblatt [50] introduced the perceptron in 1952. The perceptron is a two-layer (input and output layers) neural network which has ability to learn and recognize simple patterns. Rosenblatt proved that if the input data were linearly separable, the training procedure of the perceptron would converge and the perceptron could separate the data. However, when distributions overlap and the input data are not separable, the decision boundaries may oscillate continuously when the perceptron algorithm is applied [51]. A modification of the perceptron algorithm is the two-layer delta rule which is discussed in Section 3.2.1. The two-layer neural networks can form decision regions which are convex. The delta rule has been extended to include three or more layers. The extension is called backpropagation. By applying neural networks with three or more layers, arbitrarily shaped decision regions can be formed. Backpropagation is discussed in Section 3.2.2.

The perceptron, the delta rule and the backpropagation are probably the best known neural network models. However, several more are widely used: the Hopfield net [52] introduced by John Hopfield has been used both as an associative memory and to solve optimization problems. The Hopfield network is a relatively simple neural network which can be used as a classifier but is more appropriate for other applications. When it is used as a classifier it has to have exemplar patterns. If an output pattern matches an exemplar pattern then the output is assigned the class of the exemplar pattern. Otherwise a "no match" result occurs.

Grossberg et al. [53,54] have proposed adaptive resonance theory (ART) which includes learned top-down feedback and a matching mechanism. Their network implements a clustering algorithm which is very similar to the leader clustering algorithm [51,55]. This clustering algorithm does not use a fixed number of classes. It selects the first input as the exemplar for the first cluster. The next input is compared to the first cluster exemplar. It "follows the leader" and is clustered with the first if the distance to the first is less than a threshold. Otherwise it is the exemplar for a new cluster. The process is repeated for all the training data. The number of clusters grows with time and depends on the threshold. Since this algorithm, like the Hopfield network, uses exemplars it cannot be very successful in classification of data as complex as remote sensing data.

Kohonen has proposed a neural network called self-organizing feature maps [56] (similar to those that occur in the brain). The self-organizing feature maps is an unsupervised training method which resembles k-means clustering [55] and the algorithm works in the following fashion. After enough input vectors have been presented, weights will specify cluster or vector centers, that sample the input space such that the point density function tends to approximate the probability density function of the input vectors [51,56]. Kohonen has also proposed another neural network, learning vector quantization (LVQ), which is a special case of the self-organizing feature maps. The LVQ network is a a variant of statistical pattern recognition methods but is also in principle related to the perceptron [50,57]. It is different from the self-organizing feature maps in that the LVQ algorithm is supervised and is for that reason more attractive for our application than the self-organizing feature maps. The LVQ uses the nearest neighbor principle and could be successful in classification of complex data sets. Kohonen has recommended the number of training data to be 500 to 5000 times the number of processing elements. Although these numbers are high, the convergence can be achieved in a reasonable time since the LVQ algorithm is computationally extremely simple. It is, though, almost impossible to collect such a large number of training samples in the remote sensing application discussed here. One possibility is to use a smaller training set, recycling through it preferably with a random reordering for each cycle.

Recently, some researchers have applied neural network classifiers to remote sensing data. McClelland et al. [58] used a three-layer backpropagation algorithm to classify Landsat TM (Thematic Mapper) data. Decatur [59,60] used three-layer backpropagation to classify SAR (Synthetic Aperture Radar) data and compared his results to the results of Bayesian classification. Ersoy et al. [61] have developed a hierarchical neural network (HNN) which they have applied to classification of aircraft multispectral scanner data. Heermann et al. [62] used three-layer backpropagation to classify multitemporal data. Maslanik et al. [63] used three-layer neural networks to classify SMMR (Scanning Multichannel Microwave Radiometer) passive microwave data. All these researchers report promising performance by neural networks. However, both the classification problem and motivation are different here. The main reason that neural network methods are applied in this research to the classification of multisource remote sensing data is that these methods are distribution-free. Since multisource data are in general of multiple types, the data in each source can have different statistical distributions. By using neural network approaches we do not have the requirement of explicitly modeling the data in each source. Also, the neural network approaches avoid the problem in statistical multisource analysis of specifying how much influence each data source should have on the classification.

Two neural network approaches on which the results are based are discussed below: the delta rule and the backpropagation algorithm.

3.2.1 The Delta Rule

The delta rule, developed by Widrow and Hoff [64] in the early 1960's, is a supervised training approach where error correction is done with a least mean squares algorithm (LMS) [65]. The delta rule is so named because it changes weights in proportion to the difference between actual and desired output responses. The neural network has two layers: input and output layers. The delta rule for updating weights on the kth presentation (learning cycle = k) of an input pattern can be written as:

$$\mathbf{W}(\mathrm{k}) = \mathbf{W}(\mathrm{k}{-}1) + \eta[\mathbf{t}(\mathrm{k}) - \mathbf{W}(\mathrm{k}{-}1)\mathbf{x}(\mathrm{k})]\mathbf{x}^{\mathrm{T}}(\mathrm{k}) \qquad (3.2)$$

where $\mathbf{x}(\mathrm{k})$ is the input pattern vector, $\mathbf{t}(\mathrm{k})$ is the desired output vector, $\mathbf{W}(\mathrm{k})$ is the state of the weight matrix describing the network after k presentations, and η is a learning rate. Since the magnitudes of the weights change in proportion to η, the optimum learning rate is the one which has the largest value that does not lead to oscillation. A possible choice is $\eta = \mathrm{C/k}$, where C is a constant. That particular choice of η forces the weight matrix $\mathbf{W}(\mathrm{k})$ to stabilize after several iterations. The delta rule, which is identical to the mathematical method of stochastic approximation for regression problems, cannot be used to discriminate data that are not linearly separable and fails, for instance, in the learning of a XOR function.

Since this rule cannot discriminate data that are not linearly separable it is not expected to perform well in very difficult classification problems. However, the delta rule has been generalized to include one or more layers of hidden neurons. The generalization, which is described below, can be used to discriminate data which are not linearly separable.

3.2.2 The Backpropagation Algorithm

The generalized delta rule or the principle of backpropagation of errors was initially proposed by Werbos in 1974 [66] and later independently developed by Parker in 1986 [67], Le Cun in 1986 [68] and Rumelhart, Hinton and Williams in 1986 [69,70]. The application of the backpropagation algorithm involves two phases. During the first phase the input data are presented and propagated forward through the network to compute the output value o_{pj} in presentation of input pattern number p for each neuron j, i.e.,

$$o_{pj} = f_j(net_{pj}) \tag{3.3}$$

where $net_{pj} = \sum_i w_{ji} o_{pi}$, w_{ji} is the weight of the connection from neuron i to neuron j and f_j is the semilinear activation function at neuron j which is differentiable and nondecreasing. A widely used choice for a semilinear activation function is the sigmoid function, which is used in the experiments in Chapter 4:

$$f_j(net_{pj}) = 1/(1 + e^{-(net_{pj} + \theta_j)}) \tag{3.4}$$

where θ_j is the bias of neuron j (similar to a threshold). It is worth noting that the sigmoid function reaches one when net_{pj} goes to infinity and zero when net_{pj} goes to minus infinity. To avoid extremely large values of net_{pj}, the target values of the sigmoid function are usually selected as 0.1 and 0.9 (or -0.9 and 0.9).

The second phase involves a backward pass through the network (analogous to the the initial forward pass) during which the error signal δ_{pj} is passed to each neuron in the network and the appropriate weight changes are made according to:

$$\Delta_p w_{ij} = \eta \delta_{pj} o_{pi} \tag{3.5}$$

This second, backward pass allows the recursive computation of δ_{pj} [69]. The first step is to compute δ_{pj} for each output neuron. This is simply the difference between the actual and desired output values times the derivative of the semilinear activation function, given by

$$\delta_{pj} = (t_{pj} - o_{pj}) f_j'(net_{pj}) \tag{3.6}$$

where t_{pj} is the desired output at output neuron j. Equation (3.6) becomes

$$\delta_{pj} = (t_{pj} - o_{pj}) o_{pj} (1 - o_{pj}) \tag{3.7}$$

if the sigmoid function is used as the semilinear activation function. The weight changes can then be computed according to equation (3.5) for all connections that feed into the final layer. After this is done, the δ_{pj}'s are computed for all neurons in the penultimate layer using [69,70]:

$$\delta_{pj} = f_j'(net_{pj}) \sum_k \delta_{pk} w_{kj} \tag{3.8}$$

which takes the form

$$\delta_{pj} = o_{pj}(1 - o_{pj}) \sum_k \delta_{pk} w_{kj} \tag{3.9}$$

when the sigmoid function is used as f_j (semilinear activation function). This procedure propagates the errors back one layer, and the same process can be repeated for every layer. The backward pass has the same computational complexity as the the forward pass. $\Delta_p w_{ij}$ also gives the negative value of the gradient of the error at the outputs of the neurons multiplied by η. The norm of equation (3.5) is used as the convergence criterion for the training process in Chapter 4. When the norm of this scaled gradient is small there have been little or no weight changes by the neural network and the network has stabilized.

The backpropagation algorithm described above is a gradient descent method for finding weights in any feed-forward network with semilinear neurons. It is interesting that not all weights need be variable. Any number of weights in the network can be fixed. In this case, error is still propagated as before; the fixed weights are simply not modified.

In contrast to the delta rule, the backpropagation algorithm can be used to discriminate data that are not linearly separable. But a problem with the backpropagation is that its training process is computationally very complex. Neural network methods in general need a lot of training samples to be successful in classification. A lot of training samples together with a computationally complex algorithm produce a very long learning time. Also, since the backpropagation is a gradient descent algorithm, it may get stuck in local minima that are not globally optimal. This is mainly due to two reasons: First, gradient descent algorithms use the negative of the gradient vector to reach the minimum of the error surface but the negative gradient vector may not point directly to the minimum of the error surface. Second, the magnitude of a partial derivative of the error with respect to a weight may be such that modifying the weight by a constant proportion to that derivative can yield a minor reduction in the error measure [71].

Rumelhart et al. [69] add a momentum term to equation (3.5) in order to speed up the training. With momentum the weights are updated according to

$$\Delta w_{ij}(k+1) = \eta(\delta_{pj} o_{pi}) + \beta \Delta w_{ij}(k) \tag{3.10}$$

where k indexes the presentation number (iteration), η is the gain factor, and β is a constant which determines the effect of past weight changes on the current direction of movement in weight space. Adding a momentum term has the advantage that it filters out high frequency variations in the weight space. On the other hand momentum has the limitations that there is an upper bound on how large an adjustment it can make to a weight and also that the sign of the momentum term can cause a weight to be adjusted up the slope of the error surface, instead of down the slope as desired. Jacobs [71] introduced his delta-bar-delta learning rule as an attempt to overcome these limitations. The training of the backpropagation method can also be speeded up by using optimization methods other than the gradient descent. Such methods are discussed in the next section.

3.3 "Fast" Neural Networks

Neural network classifiers have been demonstrated to be attractive alternatives to conventional classifiers [72,73]. The two major reasons why these classifiers have not gained wider acceptance are [74]:

1. They have a reputation for being highly wasteful of computational resources during training.

2. Their training has conventionally been associated with the heuristic choice of a number of parameters; if these parameters are chosen incorrectly, poor performance results, yet no theoretical basis exists for choosing them appropriately for a given problem.

Most neural network methods are based on the minimization of a cost function. The most commonly used optimization approach applied in the minimization is the gradient descent method. Both the delta rule and the backpropagation algorithm are commonly used neural network models derived by minimizing the criterion function:

$$\epsilon_p = \frac{1}{2}\sum_{j=1}^{m}(t_{pj} - o_{pj})^2 \tag{3.11}$$

where t_{pj} is the desired output of the jth output neuron, o_{pj} is the actual output of the neuron and m is the number of output neurons. Both the delta rule and the backpropagation algorithm are derived from equation (3.11) using gradient descent. However, both of these models have the two problems listed above. The models can be modified to overcome the problems by using different optimization methods.

Watrous [75] has studied the effectivness of learning in neural networks and has shown that quasi-Newton methods are far superior to the gradient descent approach in training of neural networks. Conjugate gradient optimization [74,76] is another method which is only slightly more complicated than gradient descent but does not need any parameter selections like gradient descent (gain factor). Also, it converges faster. Fast convergence is especially important in classification of very complex data such as multisource data and very-high-dimensional data.

In this report conjugate gradient versions of the delta rule and the backpropagation are applied. The conjugate gradient neural networks are derived from equation (3.11) using conjugate gradient optimization. These methods are called: the conjugate gradient linear classifier (CGLC) (2 layers:

input and output layers) and the conjugate gradient backpropagation (CGBP) (3 layers: input, hidden and output layers) [74].

3.4 Including Statistics in Neural Networks

It is desirable but very difficult to implement first and second order statistics in neural networks by using an adaptive algorithm. White [77] has argued that standard neural network learning procedures (like the delta rule and backpropagation) are inherently statistical techniques. He also showed that certain aspects of the conditional probability law play an important role in what is learned by artificial neural networks using standard techniques. However, White's analysis does not help in including first and second order statistical information in the neural networks. Although he argues that the learning procedures for the neural networks are in essence statistical, it is desirable in many cases to have a mechanism by which first and second order statistics of the data can be explicitly incorporated in the neural network.

Kan and Aleksander [78] have proposed a probabilistic neural network for associative learning. Their network uses a new type of a probabilistic logic neuron (PLN) which has a random access memory (RAM). Training for the PLN network does not involve error propagation but uses instead a faster method of local adjustment based on Hamming distance amplification [78]. The probability portion of the network is not related to the probability distribution of the input data, but instead to the probabilities of "undefined" states in the network. Thus, the PLN network is not the kind of probabilistic neural network of interest here.

Specht has proposed two probabilistic neural network methods which are discussed in the next sections.

3.4.1 The "Probabilistic Neural Network"

The "probabilistic neural network" (PNN) was proposed by Specht [79,80]. The algorithm is as follows: Let us begin with a Parzen density estimate of a density function p_A (X) by using a Gaussian kernel function:

$$p_A(X) = \frac{1}{(2\pi)^{p/2}\sigma^d}\frac{1}{N}\sum_{i=1}^{N}\exp[-\frac{(X-X_{Ai})^T(X-X_{Ai})}{2\sigma^2}] \qquad (3.12)$$

where

i = pattern number

X = input feature vector

X_{Ai} = vector of ith training pattern from category A

σ = smoothing parameter

d = dimensionality of pattern vector

N = number of training vectors from class A

The purpose of the PNN algorithm is to use equation (3.12) to estimate the density of the data. The input layer of the network consists of one neuron for each data channel. The middle layer consists of as many neurons as there are training samples, i.e., there is one neuron for *each* training sample. The weights of the connectors from the input layer to the middle layer are the values of the training samples in each data channel. (For instance if there are five input channels, each neuron in the middle layer will have five input connectors). The activation function at the middle-layer neurons is written:

$$\exp[(XW_i - 1)/\sigma^2] \tag{3.13}$$

where $W_i = X_{Ai}$ (the weight vector). The output layer has one neuron for each information class. The middle layer nodes are connected only to the output node corresponding to the class of the training point represented by a neuron in the middle layer. The output nodes are summation nodes according to equation (3.12) and give the probability of X belonging to class A.

The PNN has several flaws. First of all equation (3.13) is derived from the exponent in equation (3.12). If the exponent in equation (3.12) is rewritten the following result is obtained:

$$\exp[-\frac{X^tX + X_{Ai}{}^tX_{Ai} - 2X^tX_{Ai}}{2\sigma^2}]$$

$$= \exp[-\frac{|X|^2 + |X_{Ai}|^2 - 2X^tX_{Ai}}{2\sigma^2}]$$

$$= \exp[\frac{X^tX_{Ai} - 0.5|X|^2 - 0.5|X_{Ai}|^2}{\sigma^2}] \tag{3.14}$$

In PNN the lengths of both X and X_{Ai} are assumed to be 1 ($|X|^2 == 1$ and $|X_{Ai}|^2 = 1$) which is how equation (3.13) is derived from equation (3.14). Assuming the lengths of the vectors to be 1 is clearly wrong. By normalizing all the data, the length information is lost and feature vectors far from the training patterns in the original data space become much closer in the normalized data space. The effect on equation (3.12) is that the probabilities for all the classes are almost equal at every pixel and the decision from the net will be wrong in most cases. On the other hand, if the data were not normalized, equation (3.13) would not be applicable because the XW_i term is much larger than 1 for most input vectors and the exponent would approach infinity.

Apart from the serious flaw pointed out above it is questionable whether PNN should be called a neural network. It can be considered an attempt to find a parallel implementation of Parzen density estimation. If the approach were correctly derived this method might work well on a parallel computer. However, everything is predetermined by the user rather than by iterative training of the network.

Parzen density estimation has the shortcoming that it requires a large number of training samples for estimating the density when the dimensionality is large. Silverman [45] has investigated Parzen density estimation and reports the results (from [45]) shown in Table 3.1. As seen in Table 3.1 the required sample size grows fast with increasing dimensionality. Clearly this approach is impractical for applications involving very-high-dimensional data.

3.4.2 The Polynomial Adaline

Specht [80] has also proposed the polynomial adaline (Padaline) which is closely related to PNN. The polynomial adaline uses all higher orders and cross products of the input data and has the form:

$$
\begin{aligned}
P(X) = {} & D_{0\ldots0} + D_{10\ldots0}X_1 + D_{010\ldots0}X_2 \\
& + \ldots + D_{0\ldots01}X_p + D_{20\ldots0}X_1{}^2 \\
& + D_{z1z2\ldots zp}X_1{}^{z1}X_2{}^{z2}\ldots X_p{}^{zp} + \ldots
\end{aligned}
\tag{3.15}
$$

Specht derived a relatively simple method to determine the coefficients D for

Table 3.1

Sample Size Required in Parzen Density Estimation when Estimating a Standard Multivariate Normal Density Using a Normal Kernel [45]

Dimensionality	Required sample size
1	4
2	19
3	67
4	223
5	768
6	2790
7	10700
8	43700
9	187000
10	842000

equation (3.15) based on training patterns. These coefficients are updated for each observed training . sample. The algorithm makes it possible to use hundreds or thousands of terms in the polynomial discriminant function without overfitting the data even if the number of training samples is smaller than the number of coefficients (good behavior because of smoothing).

The Padaline classifier is a one-pass network like the PNN and again it is questionable whether the Padaline should be called a neural network. It is necessary for the user to decide the number of terms being used. The major disadvantage of this method is that it is computationally complex especially if many terms are used. However, the computational and storage requirements increase only linearly with the number of terms used.

3.4.3 Higher Order Neural Networks

The most straight-forward way to include statistical information in neural networks is to use higher order correlations. The higher order correlation method is desirable when the input data are of relatively low dimensionality. When d-dimensional data are mapped with a second order mapping, the resulting dimensionality will be $d + d(d+1)/2$. It is clear that the dimensionality of the higher order mapping increases rapidly with d. High dimensionality makes the neural network training procedures slower. Therefore, higher order mapping is not desirable if d is large.

If second order correlations are used, a "two-layer neural network" can be implemented with deterministic weights to compute the likelihood function of a Gaussian maximum likelihood classifier. The reason for the ease of the implementation is that the log of the likelihood function is quadratic and can therefore be written as:

$$X^tAX + X^tB+C \tag{3.16}$$

where A is a matrix, B a vector and C a constant. A, B and C can be estimated from the mean vectors and the covariance matrices of the training data [81].

When a classification problem has M (M > 1) classes, the "neural network" classifier must have 3 layers. The first 2 layers compute the likelihood function, but an additional neural network is concatenated to the outputs to find the class which has the highest likelihood. This additional neural network is MAXNET [51], a neural network which is easily implemented to find the maximum value from a particular set.

A problem with the Gaussian "neural network" is that it is more a parallel implementation of a Gaussian maximum likelihood classifier than an adaptive neural network. Everything is fixed beforehand. An adaptive approach which could use the pre-fixed values as initial values would be of more interest.

3.4.4 Overview of Statistics in Neural Network Models

From the above discussion it can be concluded that implementing statistics in an adaptive neural network is a very difficult problem. Several authors have suggested "neural networks" which are actually parallel realizations of well-known statistical methods. These methods are only attractive alternatives to common statistical methods if they are implemented on parallel machines. However, the PNN has to be considered questionable for almost any problem and the Gaussian network is not practical for very-high-dimensional problems.

Although it would be desirable to include first and second order statistics in the neural networks it will not be done here. One of the advantages of using neural networks for classification of multitype data is that the neural networks model the dependence between all the data whereas most of the statistical methods discussed in Chapter 2 cannot do that when a convenient multivariate statistical model does not exist or is unknown. If the neural networks could be provided with some parametric statistical information, it would have to be on a source-by-source basis, if second order statistics were used. Evidently this statistical implementation problem needs a lot of work.

In the experiments in this report, conjugate gradient versions of the delta rule and the backpropagation algorithm will be the only neural networks applied.

CHAPTER 4
EXPERIMENTAL RESULTS

The methods discussed in Chapters 2 and 3 were applied to classification of multisource and very-high-dimensional data sets. Three data sets were used in experiments. Two of the data sets were multisource remote sensing and geographic data. The third data set consisted of very-high-dimensional simulated High Resolution Imaging Spectrometer (HIRIS) data. The linear opinion pool, statistical multisource classifier, the minimum Euclidean distance algorithm and the maximum likelihood method for Gaussian data were the statistical methods used in classification (when these methods were appropriate). For the multisource remote sensing and geographic data sets, the linear opinion pool and the statistical multisource classifier were used in conjunction with three non-Gaussian modeling methods: the histogram method, the maximum penalized likelihood method and Parzen density estimation. The objective of using all these non-Gaussian methods was to see how well they performed in statistical multisource classification.

The conjugate gradient linear classifier and the conjugate gradient backpropagation were the neural network models used in the experiments. The statistical methods and the neural network models were compared based on classification accuracies for different sample sizes of training data, dimensionalities of input data and on classification time.

4.1 Source-Specific Probabilities

In order to apply the statistical multisource classifier and the linear opinion pool, the source-specific probabilities can be written in the following form:

$$p(\omega_j | x_i) = [p(x_i)]^{-1} \sum_{k=1}^{m_i} p(x_i | d_k, \omega_j) p(d_k, \omega_j) \qquad (4.1)$$

Here m_i is the number of data classes for source i and $p(x_i)$ is computed by:

$$p(x_i) = \sum_{j=1}^{M} \sum_{k=1}^{m_i} p(x_i | d_k, \omega_j) p(d_k, \omega_j) \qquad (4.2)$$

where M is the number of information classes. For each source, the joint probabilities $p(d_k, \omega_j)$ can be tabulated in a joint occurrence matrix by comparing single-source data-class classifications to information classes in a reference map. To reduce considerably the computation and memory requirements, the class-conditional probabilities can be computed independently of information classes, i.e., by setting:

$$p(x_i | d_k, \omega_j) = p(x_i | d_k) \qquad \text{for all } \omega_j \qquad (4.3)$$

This approximation is useful if the distribution of a data class is the same regardless of information class and if the number of data classes is different from the number of information classes. However, if the number of data classes and information classes are the same and the information and data classes have a one-to-one correspondence, the source-specific probabilities can be modeled by:

$$p(\omega_j \mid x_i) = p(d_j \mid x_i) \tag{4.4}$$

In the following experiments, the approximation in equation (4.3) was used when the information classes did not directly correspond to the data classes. As said previously, the approximation is useful if the distribution of a data class is the same regardless of information class. However, the approximation is unlikely to hold exactly in the case of unsupervised classification.

All of the experiments in this chapter were run on a Gould NP1 mini super computer. Although the NP1 machine is fast, the approximation in equation (4.3) was essential to reduce the memory requirements in the classifications of the statistical multisource classifier and the linear opinion pool.

4.2 The Colorado Data Set

The statistical and neural network classification methods were used to classify a data set consisting of the following 4 data sources:

1) Landsat MSS data (4 data channels)

2) Elevation data (in 10 m contour intervals, 1 data channel)

3) Slope data (0-90 degrees in 1 degree increments, 1 data channel)

4) Aspect data (1-180 degrees in 1 degree increments, 1 data channel)

Each channel comprises an image of 135 rows and 131 columns; all channels are co-registered.

The area used for classification is a mountainous area in Colorado. This area is a part of a larger region which has previously been analyzed by Hoffer et al. [7,10]. The area has 10 ground cover classes which are listed in Table 4.1. One class is water; the others are forest type classes. It was very difficult to distinguish between the forest types using the Landsat MSS data alone since the forest classes showed very similar spectral responses. With the help of elevation, slope and aspect data, they could be better distinguished.

Ground reference data were compiled for the area by comparing a cartographic map to a color composite of the Landsat data and also to a line printer output of each Landsat channel. By this method 2019 ground reference points (11.4% of the area) were selected. Ground reference consisted of two or more homogeneous fields in the imagery for each class. In the first experiments on this data set, the largest field for each class was selected as a training field and the other fields were used for testing the classifiers. Overall 1188 pixels were used for training and 831 pixels for testing the classifiers. This was the same data used in [82] and some of the results in Section 4.2.1 were reported there.

4.2.1 Results: Statistical Approaches

Two statistical methods were used in the experiments reported here: 1) minimum Euclidean distance (MD) [30], and 2) statistical multisource classification (SMC) with the modifications discussed in Section 2.3.1. The MD method is a "simple" stacked-vector approach which has been used with some success in classification of remotely sensed data from single-sources. (Other stacked vector approaches like the maximum likelihood method for Gaussian data and the minimum Mahalanobis distance were not applicable, because the

Table 4.1

Training and Test Samples for Information Classes in the First Experiment on the Colorado Data Set

Class #	Information Class	Training Size	Testing Size
1	water	408	195
2	Colorado blue spruce	88	24
3	mountane/subalpine meadow	45	42
4	aspen	75	65
5	Ponderosa pine	105	139
6	Ponderosa pine/Douglas fir	126	188
7	Engelmann spruce	224	70
8	Douglas fir/white fir	32	44
9	Douglas fir/Ponderosa pine/aspen	25	25
10	Douglas fir/white fir/aspen	60	39
Total		1188	831

data were not truly Gaussian and a few of the stacked vector covariance matrices were singular.)

The results of the classification using the MD method are shown in Tables 4.2 (training) and 4.3 (test) where OA represents overall accuracy and AVE means average (over the classes) accuracy. The results in Tables 4.2 and 4.3 are clearly unacceptable. The MD method gave only 43.27% overall accuracy for training data and 22.26% overall accuracy for test data.

We next turn to the classification unsing the SMC method. To satisfy the underlying assumptions of the SMC algorithm and the global membership function in equations (2.31a) and (2.31b), it was necessary to show that the data sources could be treated independently in the classification. This was accomplished by looking at the class-specific correlations between all seven data channels using the reference data. The correlations between the data sources were in most cases low. For a few of the information classes there was no variation in the topographic data sources and consequently the correlation was undefined. Since the correlations between the sources were low in most defined cases, the data sources could be treated as independent and the global membership function in equations (2.31a) and (2.31b) was used as the classifier.

Each source was used independently for training. The data classes in the Landsat MSS source were modeled by the Gaussian distribution, where the means and covariance matrices were estimated from the training fields. The other data sources had non-Gaussian data classes. For these sources the normalized histograms of the training fields were used to estimate the density functions.

Table 4.2

Classification Results for Training Samples when Minimum Euclidean Distance Classifier is Applied.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	47.3	100.0	31.1	28.0	0.0	0.0	67.4	59.4	44.0	28.3	43.27	40.55
# of pixels	408	88	45	75	105	126	224	32	25	60	1188	1188

CPU time for training and classification: 2 sec.

Table 4.3

Classification Results for Test Samples when Minimum Euclidean Distance Classifier is Applied.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	38.9	100.0	0.0	16.9	0.0	6.9	75.7	4.5	4.0	12.8	22.26	25.99
# of pixels	195	24	42	65	139	188	70	44	25	39	831	831

Statistical multisource classification was performed on the data with varying weights (reliability factors) for the data sources. The results of classification for the training fields are shown in Table 4.4 and for the the test fields in Table 4.5. The reliability and uncertainty measures introduced in Section 2.3.2 were used to rank the data sources. These results indicate that the Landsat MSS data was the most reliable source, elevation second, aspect third and the slope source the least reliable. This was the same ranking produced by the equivocation measure as indicated in Table 4.6. (The separability measures using the Gaussian assumption could not be applied here since some of the data classes in the topographic sources were not truly Gaussian and had singular covariance matrices as mentioned above.) In all the experiments the Landsat MSS data were given the largest weight while the weights of the other sources were varied.

The classification of the training samples (Table 4.4) showed that by combining all the sources with equal weights the overall classification accuracy (OA) improved to 74.2%, i.e., by more than 6% compared to the best accuracy in the single-source classification (Landsat MSS: 67.9%). By lowering the weights on the topographic sources, the overall accuracy could be increased to 78.0%. Therefore, by changing the weights of the sources the overall classification accuracy of the training samples improved by 3.8%. This "best" result was achieved when the Landsat source was given full weight and the other sources were given 40% weight. It was also very nearly achieved when the Landsat MSS data had full weight, the elevation source had 50% weight, the aspect source had 40% weight and the slope source had 30% weight (77.9% overall accuracy). That weighting controlled the influence from

Table 4.4

Statistical Multisource Classification of Colorado Data: Training Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	Single Sources											
MSS	99	48	0	80	9	69	92	0	0	0	67.9	39.7
elevation	100	0	0	23	17	13	98	0	16	20	58.4	28.7
slope	100	0	0	0	5	64	0	0	0	0	41.5	16.9
aspect	100	0	0	44	42	15	59	0	0	0	53.6	26.0
m e s a	Multiple Sources											
1. 1. 1. 1.	100	98	0	35	35	80	100	0	0	0	74.2	44.8
1. .5 .5 .5	100	99	0	65	34	76	94	0	0	62	77.6	53.0
1. .4 .4 .4	100	100	11	71	33	73	95	0	0	58	78.0	54.1
1. .3 .3 .3	100	100	11	75	27	71	96	0	0	42	76.9	52.2
1. .2 .2 .2	100	98	11	75	23	71	96	0	0	26	75.5	50.0
1. .1 .1 .1	100	96	18	75	15	66	97	38	0	0	74.2	50.5
1. .8 .4 .6	100	99	0	64	37	79	93	0	0	60	77.8	53.2
1. .8 .1 .2	100	100	11	74	17	76	95	0	0	35	76.0	50.8
1. .6 .4 .5	100	99	4	67	34	76	94	0	0	60	77.8	53.4
1. .5 .3 .4	100	100	11	73	33	75	95	0	0	49	77.9	53.6
1. .4 .2 .3	100	100	11	75	27	73	96	0	4	38	77.0	52.4
1. .3 .1 .2	100	99	11	75	18	74	96	0	4	22	75.4	49.9
# of pixels	408	88	45	75	105	126	224	32	25	60	1188	1188

The columns labeled m e s a indicate the weights applied to the Landsat MSS (m), elevation (e), slope (s) and aspect (a) sources.

CPU time for training and classification: 14 sec.

Table 4.5

Statistical Multisource Classification of
Colorado Data: Test Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
					Single Sources							
MSS	97	0	0	0	25	79	97	0	0	0	53.1	29.8
elevation	100	0	0	20	2	21	100	0	8	21	40.4	27.2
slope	86	0	0	0	0	5	33	0	0	0	24.3	12.4
aspect	95	0	0	15	1	6	19	0	0	0	26.7	13.6
m e s a					Multiple Sources							
1. 1. 1. 1.	86	0	0	25	35	92	86	0	0	0	56.0	32.4
1. .5 .5 .5	86	0	0	48	45	80	97	0	0	0	57.9	35.6
1. .4 .4 .4	86	0	0	52	49	76	97	0	0	0	57.9	36.0
1. .3 .3 .3	86	0	0	54	51	63	97	0	0	44	57.4	39.5
1. .2 .2 .2	97	0	0	0	54	80	97	0	0	31	59.5	35.9
1. .1 .1 .1	93	0	0	0	54	76	97	0	0	26	57.3	34.6
1. .8 .4 .6	100	0	0	51	38	84	97	0	0	0	60.8	37.0
1. .8 .1 .2	91	0	0	60	48	72	97	0	0	0	58.6	36.8
1. .6 .4 .5	86	0	0	51	44	81	97	0	0	0	58.0	35.9
1. .5 .3 .4	86	0	0	54	48	74	97	0	0	0	57.5	35.9
1. .4 .2 .3	97	0	0	57	51	55	97	0	0	41	58.2	39.8
1. .3 .1 .2	95	0	0	0	55	80	97	0	0	33	59.3	36.0
# of pixels	195	24	42	65	139	188	70	44	25	39	831	831

The columns labeled m e s a indicate the weights applied to the Landsat MSS (m), elevation (e), slope (s) and aspect (a) sources.

Table 4.6

Equivocation of the Data Sources

Source	Equivocation	Rank
MSS	0.216955	1
Elevation	0.252676	2
Aspect	0.277244	3
Slope	0.289636	4

the sources according to the ranking of both the reliability measures. Using some other weight combinations that ranked the sources in the same order as the reliability measures also gave very good results. In summary, the results in Table 4.4 show that the overall classification accuracy could be improved by reducing the weights of some of the data sources. In Table 4.4 it is also seen that if the weights of the data sources were decreased too much, the overall classification accuracy went down, as would be expected.

The results in Table 4.5 are very similar to the ones in Table 4.4. Table 4.5 shows the results of the classification of test fields and therefore the classification accuracy is generally lower than in Table 4.4. If the sources all had equal weights, then the overall accuracy was 56.0% which was 2.9% greater than the overall classification accuracy of the best single-source (Landsat MSS: 53.1%). This was not as much increase as in the case of training data. By lowering the weights on the topographic data sources the overall classification accuracy was improved to 60.8%, which was 4.8% more than with the equal weights. This best result was achieved when the Landsat source had full weight, the elevation source 80% weight, the aspect source 60% weight and the slope source 40% weight. This particular weighting ranked the sources in the same order as the reliability measures.

4.2.2 Results: Neural Network Models

The two neural network approaches, the conjugate gradient linear classifier (CGLC) and the conjugate gradient back propagation (CGBP), were implemented in experiments to classify the data. (The neural network programs were written by Etienne Barnard [74].) The neural networks were trained with Gray-coded input vectors rather than binary input vectors, as discussed in Chapter 3. The author has previously shown empirically that the Gray-code gives good results in classification of this data set [82]. Since five of the seven data channels take values in the range from 0 to 255, each data channel was represented by 8 bits and therefore 8 input neurons. The total number of input neurons was 7*8 = 56. Since the number of information classes was 10, the number of output neurons was selected as 10. The training procedures of the neural networks were considered to have converged if the norm of the gradient of the error at the outputs was less than 0.0001.

a) Experiments with the Conjugate Gradient Linear Classifier

The results using the two-layer CGLC are shown in Tables 4.7 (training) and 4.8 (test). The training procedure for this neural network did not converge but was stopped after 319 iterations because the error function could not be decreased after that. The highest overall accuracy (94.87%) and the highest average accuracy (92.49%) for training data were achieved by 200 iterations. These accuracies were much higher than those achieved with the SMC algorithm in Section 4.2.1. However, the best overall accuracy for test data was reached after only 100 iterations (55.11%). This was significantly lower than the highest overall accuracy achieved with the SMC algorithm. But the neural network was better than the SMC in terms of average classification accuracy. This result shows that the CGLC is better than the SMC in capturing class-specific information but the SMC seeks to achieve the minimum probability of error. A major problem with the CGLC and other neural networks is deciding when to stop the training procedure. If a neural

Table 4.7

Conjugate Gradient Linear Classifier Applied to
Colorado Data: Training Samples.

Number of	CPU	Percent Agreement with Reference for Class											
iterations	time	1	2	3	4	5	6	7	8	9	10	OA	AVE
50	100	100.0	97.7	75.6	94.7	68.6	79.4	99.1	81.3	76.0	96.7	92.26	86.91
100	186	100.0	98.9	82.2	98.7	69.5	84.9	99.6	90.6	84.0	98.3	94.11	90.67
150	270	100.0	98.9	84.4	98.7	69.5	85.7	100.0	96.9	84.0	98.3	94.53	91.64
200	348	100.0	98.9	82.2	98.7	71.5	85.7	100.0	96.9	92.0	100.0	94.87	92.59
250	435	100.0	98.9	82.2	98.7	70.5	85.7	100.0	96.9	92.0	100.0	94.78	92.49
300	524	100.0	98.9	82.2	98.7	70.5	85.7	100.0	96.9	92.0	100.0	94.78	92.49
319	557	100.0	98.9	82.2	98.7	70.5	85.7	100.0	96.9	92.0	100.0	94.78	92.49
# of pixels		408	88	45	75	105	126	224	32	25	60	1188	1188

Table 4.8

Conjugate Gradient Linear Classifier Applied to
Colorado Data: Test Samples.

Number of	Percent Agreement with Reference for Class											
iterations	1	2	3	4	5	6	7	8	9	10	OA	AVE
50	95.4	83.3	33.3	41.5	10.8	39.9	100.0	2.3	12.0	87.2	53.55	50.57
100	96.4	83.3	40.5	41.5	11.5	43.6	100.0	2.3	12.0	87.2	55.11	51.83
150	95.9	83.3	38.1	41.5	10.8	41.5	100.0	4.5	12.0	84.6	54.27	51.22
200	94.9	83.3	33.3	35.4	11.5	43.6	100.0	2.3	12.0	79.5	53.55	49.58
250	94.9	83.3	33.3	36.9	11.5	44.7	100.0	2.3	16.0	79.5	54.03	50.24
300	94.9	83.3	33.3	38.5	11.5	44.1	100.0	2.3	16.0	79.5	54.03	50.34
319	94.9	83.3	33.3	38.5	11.5	44.1	100.0	2.3	16.0	79.5	54.03	50.34
# of pixels	195	24	42	65	139	188	70	44	25	39	831	831

network is overtrained it will not give the best accuracies for test data. The reason is that the network gets too specific to the training data and does not generalize as well.

The CGLC took longer to train than the SMC. Three hundred iterations took 524 CPU sec compared to 104 for the statistical method. Also, the *classification* of the data took 10 sec for the CGLC but 7 sec for the SMC.

b) Experiments with Conjugate Gradient Back Propagation

The CGBP was implemented in experiments with three or more layers (input, output and hidden layers). Having more than one hidden layer did not improve the classification performance of this neural network, so only the results with three layers are discussed here. Three-layer networks with 16, 32, 48 and 64 hidden neurons were tried but the performance of the CGBP in terms of classification accuracy was not improved by using more than 32 hidden neurons. Therefore, 32 hidden neurons were used in the experiments reported here.

The CGBP (Tables 4.9 (training) and 4.10 (test)) showed the best performance of all the methods in terms of overall and average classification accuracies of training data. As with the CGLC, the training procedure of the CGBP did not converge. At 676 iterations the error function could not be decreased and the training procedure stopped. At 350 iterations the highest overall accuracy of training data was reached (98.40%) and at 600 iterations the highest average accuracy of training data (98.04%) was observed. These accuracies did not improve with more than 600 iterations. For test data, the CGBP gave very similar accuracies to the CGLC. At 200 iterations the

Table 4.9

Conjugate Gradient Backpropagation Applied to Colorado Data: Training Samples.

Number of	CPU	Percent Agreement with Reference for Class											
iterations	time	1	2	3	4	5	6	7	8	9	10	OA	AVE
50	384	100.0	100.0	55.6	90.7	66.7	77.8	99.6	37.5	40.0	91.7	88.97	75.96
100	737	100.0	100.0	77.8	97.3	75.2	85.7	100.0	81.3	80.0	96.7	94.19	89.40
150	1073	100.0	100.0	86.7	98.7	81.0	89.7	100.0	100.0	96.0	100.0	96.55	95.21
200	1427	100.0	100.0	93.3	100.0	85.7	92.1	100.0	100.0	100.0	100.0	97.64	97.11
250	1788	100.0	100.0	93.3	100.0	86.7	95.2	100.0	100.0	100.0	100.0	98.06	97.58
300	2102	100.0	100.0	95.6	100.0	88.6	95.2	100.0	100.0	100.0	100.0	98.32	97.94
350	2517	100.0	100.0	95.6	100.0	89.5	95.2	100.0	100.0	100.0	100.0	98.40	98.03
400	2820	100.0	100.0	95.6	100.0	89.5	95.2	100.0	100.0	100.0	100.0	98.40	98.03
600	4198	100.0	100.0	95.6	100.0	90.5	94.4	100.0	100.0	100.0	100.0	98.40	98.04
676	4730	100.0	100.0	95.6	100.0	88.6	96.0	100.0	100.0	100.0	100.0	98.40	98.02
# of pixels		408	88	45	75	105	126	224	32	25	60	1188	1188

Table 4.10

Conjugate Gradient Backpropagation Applied to Colorado Data: Test Samples.

Number of	Percent Agreement with Reference for Class											
iterations	1	2	3	4	5	6	7	8	9	10	OA	AVE
50	97.9	83.3	19.0	47.7	10.8	46.8	100.0	6.8	0.0	94.9	55.72	50.72
100	98.5	83.3	40.5	33.8	10.1	43.1	100.0	2.3	0.0	94.9	54.63	50.65
150	100.0	83.3	42.9	38.5	12.2	44.7	100.0	0.0	12.0	92.3	56.32	52.59
200	99.0	83.3	45.2	38.5	17.3	40.4	100.0	2.3	12.0	94.9	56.32	53.29
250	97.9	83.3	47.6	36.9	13.7	40.4	100.0	2.3	12.0	84.6	54.99	51.87
300	97.9	83.3	42.9	36.9	11.5	42.6	100.0	2.3	16.0	84.6	54.99	51.80
350	97.9	79.2	42.9	36.9	12.2	41.5	100.0	2.3	12.0	79.5	54.39	50.44
400	97.9	79.2	40.5	36.9	12.2	41.0	100.0	2.3	12.0	84.6	54.39	50.66
600	97.9	79.2	38.1	41.5	15.8	42.0	100.0	2.3	12.0	82.1	55.35	51.09
676	97.9	79.2	38.1	41.5	15.1	42.0	100.0	2.3	16.0	82.1	55.35	51.42
# of pixels	195	24	42	65	139	188	70	44	25	39	831	831

highest overall and average accuracies of test data were reached, 56.32% and 52.59% respectively. Therefore, the CGBP did not do as well as the SMC in terms of overall classification accuracy of test data but it did better in terms of average accuracy. In these experiments the CGBP had an overtraining problem similar to the CGLC; it gave somewhat less than optimal results for test data classified by the network giving the most accurate results for training data.

The CGBP was much slower in training than the CGLC because of the 32 hidden neurons. Training the CGBP for 400 iterations took 2663 sec. However, the classification of the data took only 21 sec which is about twice the time consumed by the CGLC and three times the classification time of the SMC (7 sec).

The best results of the first experiment on Colorado data are shown in Figure 4.1. As seen in the figure, the SMC method outperformed the neural networks in classification of test data although the neural networks performed much better in classification of training data. The results in this experiment illustrate how important it is to select representative training samples when training a neural network. The CGBP network gave more than 90% overall accuracy of training data but only just more than 50% for test data. The training data used here might not be representative since only one training field was selected for each information class. This limited each information class to a single subclass. The classification results for the training fields indicate that if representative training samples are available, the neural networks can do well in classification of multisource data. Significantly, arriving at a truly representative set of training samples can be very difficult

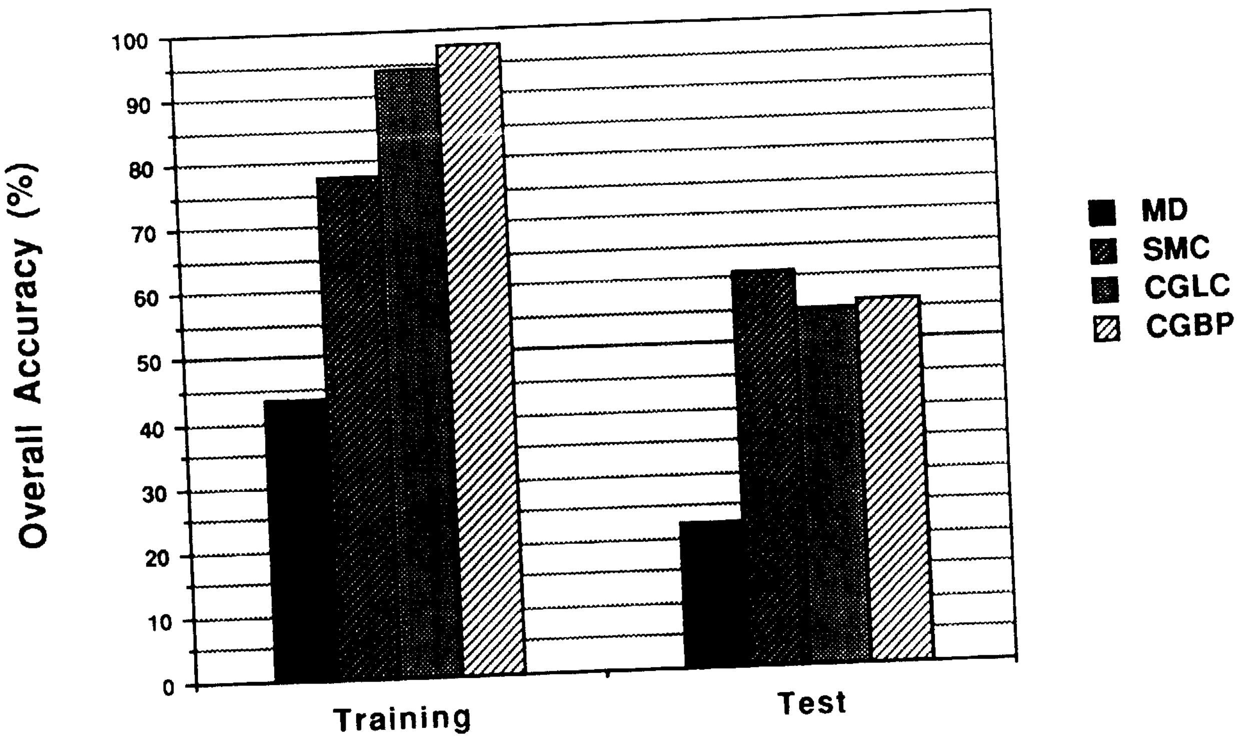


Figure 4.1 Summary of Best Classification Results for First Experiment on Colorado Data.

in practical remote sensing applications. But to demonstrate how well the classification methods would do with a more representative sample, a second experiment on the Colorado data was conducted.

4.2.3 Second Experiment on Colorado Data

To achieve a more representative training sample, uniformly spaced samples were selected from all fields available for each class. By this approach, 1008 samples were obtained for training and 1011 samples for testing (Table 4.11). By considering the JM distances between the different training fields in the MSS data, it was determined that the Landsat MSS source should be trained on 13 data classes. The selection of the data classes was done in the following way. If a field from a specific class was more distant than 1.2 in the sense of JM distance from a field within the same class, the fields were considered to be from two different data classes (JM distance has a maximum of 1.41421). Using this criterion, class 3 (mountane/subalpine meadow) was split into two data classes, and class 7 (Engelmann spruce) was divided into 3 data classes. All the other information classes had only one data class. In the methods applied below, the classifiers were trained on the 13 data classes.

4.2.4 Results of Second Experiment: Statistical Methods

In these experiments three statistical approaches were used: 1) The MD approach, 2) the SMC algorithm and 3) the linear opinion pool (LOP). The results using the MD algorithm are shown in Tables 4.12 (training) and 4.13 (test). Since the training data are more representative than in Section 4.2.1,

Table 4.11

Training and Test Samples for Information Classes in the Second Experiment on the Colorado Data Set

Class #	Information Class	Training Size	Testing Size
1	water	301	302
2	Colorado blue spruce	56	56
3	mountane/subalpine meadow	43	44
4	aspen	70	70
5	Ponderosa pine	157	157
6	Ponderosa pine/Douglas fir	122	122
7	Engelmann spruce	147	147
8	Douglas fir/white fir	38	38
9	Douglas fir/Ponderosa pine/aspen	25	25
10	Douglas fir/white fir/aspen	49	50
Total		1008	1011

Table 4.12

Classification Results for Training Samples when Minimum Euclidean Distance Classifier is Applied.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	41.5	98.2	25.6	37.1	37.6	0.0	73.5	0.0	40.0	24.5	40.28	37.80
# of pixels	301	56	43	70	157	122	147	38	25	49	1008	1008

CPU time for training and classification: 2 sec.

Table 4.13

Classification Results for Test Samples when Minimum Euclidean Distance Classifier is Applied.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	40.1	100.0	34.1	30.0	32.5	0.8	69.4	0.0	28.0	20.0	37.98	35.49
# of pixels	302	56	44	70	157	122	147	38	25	50	1011	1011

the test results are significantly better (Table 4.3). However, the results in Tables 4.2, 4.3 and 4.12 and 4.13 show that the MD is not an acceptable choice for classification of this data set.

By looking more closely at the four data sources it is easy to see why the data were difficult to classify. In Table 4.14 the JM distances between the 10 information classes of the Landsat MSS data are shown. Although the average separability of the MSS data (1.308) was relatively high, it is seen from Table 4.14 that only classes 1 (water) and 7 (Engelmann spruce) were very separable from the other 8 classes. Also, water and Engelmann spruce were the largest classes and therefore had the biggest impact on the average separability. With the exception of Engelmann spruce, other forest classes (classes 2 to 6 and 8 to 10) were not very separable from each other. Using the topographic information would be expected to help distinguish the forest classes. Figures 4.2, 4.3 and 4.4 show the class-specific histograms (information classes) of the topographic training data. The magnitude of class 1 is actually 301 in each figure. It was reduced in the figures to make the magnitudes of the other classes more visible.

Looking at Figure 4.2 (elevation histograms), it is seen that class 1 dominates in the lower elevations, but several other classes, especially class 7, can be distinguished from it for the higher elevations. In Figure 4.3 (slope histograms), the data are not as distinguishable as in Figure 4.2. Class 1 dominates the zero slope, but class 7 has several peaks with higher slope values. Classes 4 and 6 are also separable from the other classes but the slope source is clearly not as informative overall as the elevation data. In Figure 4.4 the class-specific histograms of the aspect data are shown. The aspect data

Table 4.14

Pairwise JM Distances Between the 10 Information Classes in the Landsat MSS Data Source (Maximum Separability is 1.41421)

Class #	2	3	4	5	6	7	8	9	10
1	1.41274	1.37295	1.40880	1.40250	1.41421	1.41336	1.41331	1.40238	1.41419
2	-	1.16528	1.05169	0.99912	1.36284	1.40287	1.24416	1.07844	1.08332
3	-	-	1.29855	1.28122	1.38693	1.38369	1.36175	1.30351	1.33886
4	-	-	-	0.95808	1.27051	1.40729	1.15989	0.49988	1.00649
5	-	-	-	-	1.02387	1.39967	0.73897	1.02265	0.94368
6	-	-	-	-	-	1.40999	0.73667	1.26707	1.15118
7	-	-	-	-	-	-	1.40714	1.40779	1.40772
8	-	-	-	-	-	-	-	1.16382	0.92488
9	-	-	-	-	-	-	-	-	1.04157
Average:	1.30809								

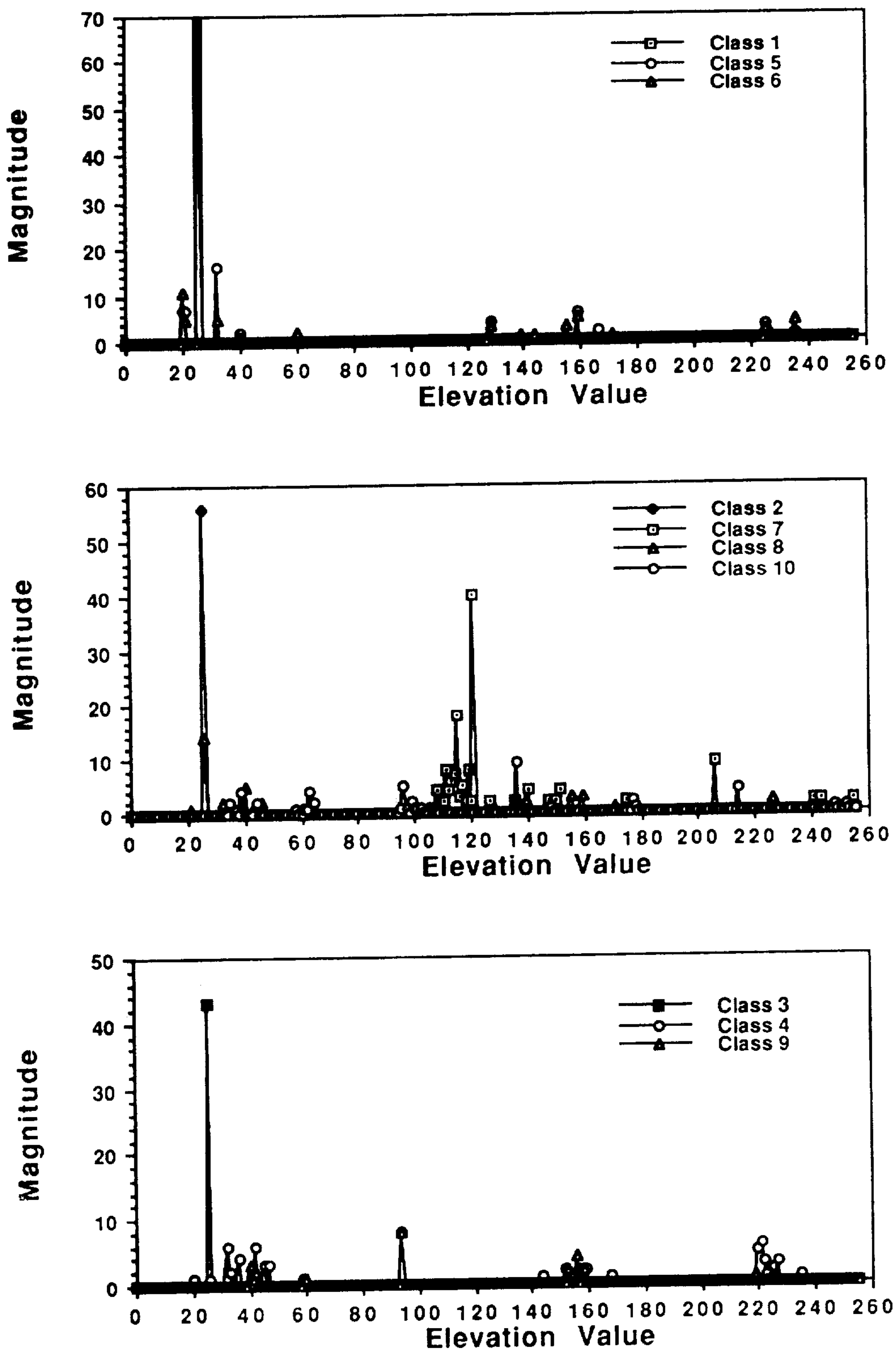


Figure 4.2 Class Histograms of Elevation Data in the Colorado Data Set

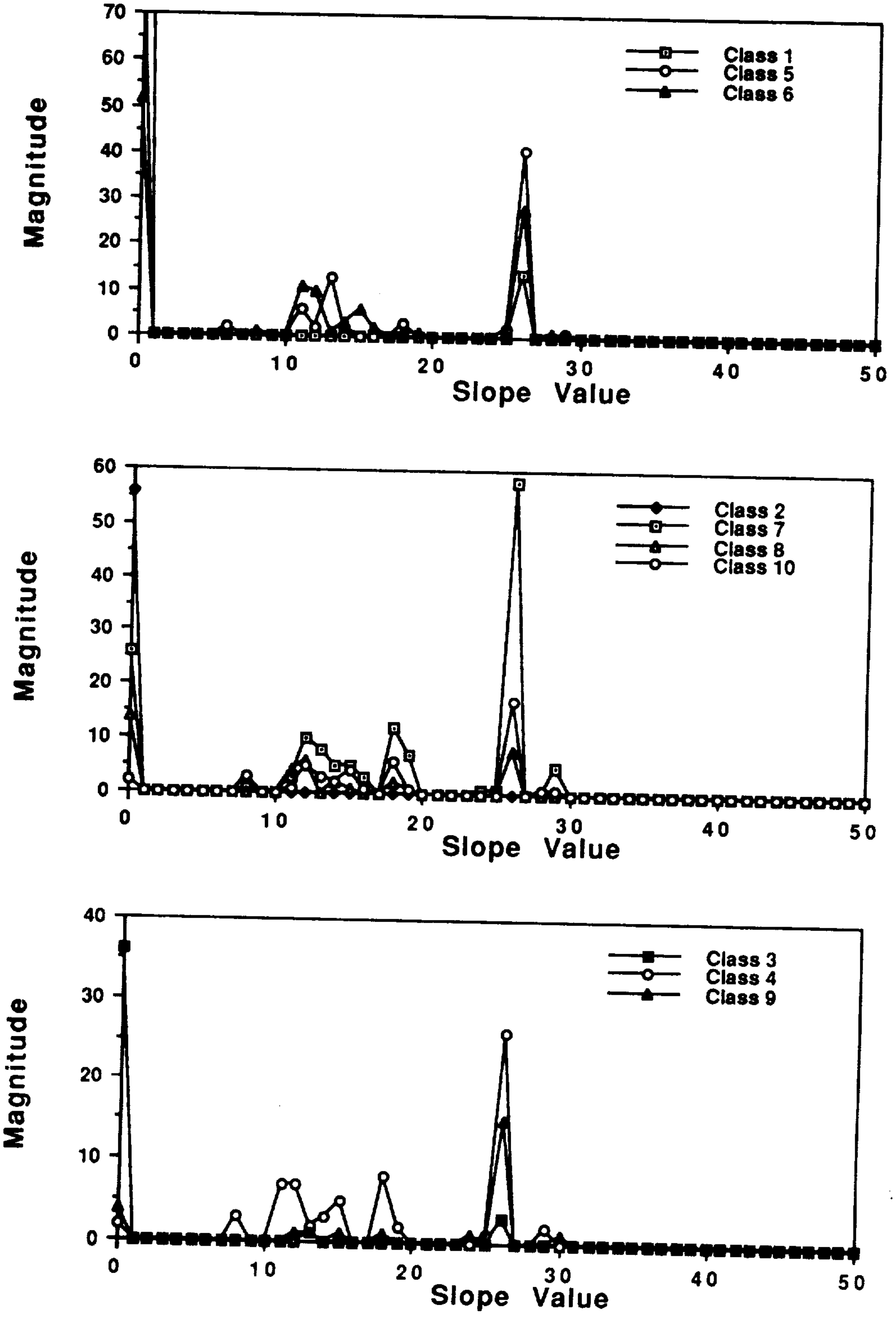


Figure 4.3 Class Histograms of Slope Data in the Colorado Data Set

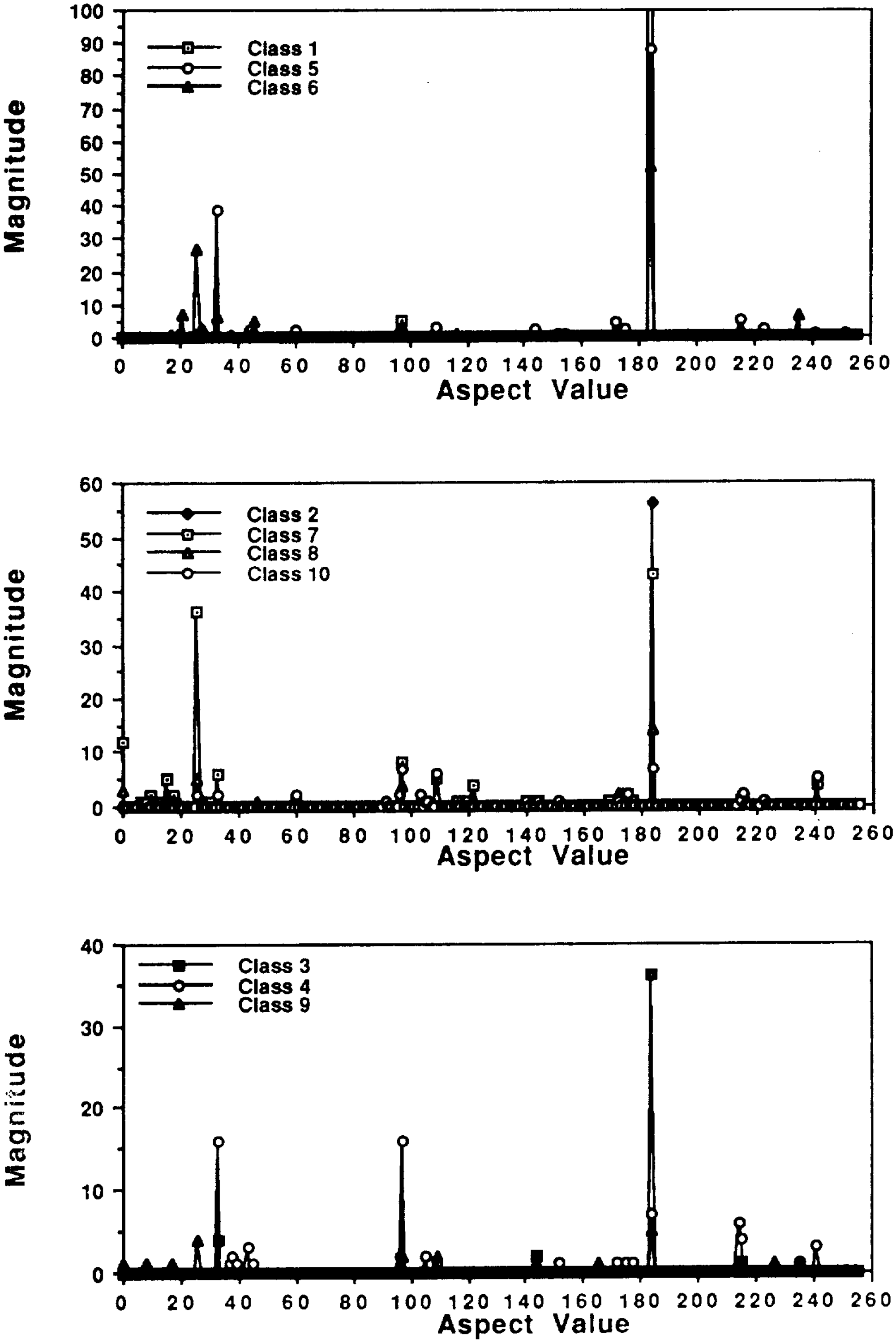


Figure 4.4 Class Histograms of Aspect Data in the Colorado Data Set

are evidently more informative than the slope data. Several of the classes have small peaks and class 1 has the biggest peak around 180 degrees. Since it did not help in terms of source-specific overall accuracy to use the 13 data classes for the topographic data, the topographic sources were trained only on the 10 information classes when used in conjunction with the SMC and LOP classifiers.

The experiments with the SMC and LOP methods were done using three different density estimation methods for the topographic data sources in order to see how well different methods modeled the data. The density estimation methods were discussed in Chapter 2: 1) the histogram approach, 2) the maximum penalized likelihood method and 3) Parzen density estimation. Experiments with each modeling method are treated separately below. As mentioned in Section 4.2.1, the data sources can be treated independently and thus the SMC method can be applied in classification of this data set.

a) Topographic Data Modeled by Histograms

The results of the SMC classifications are shown in Tables 4.15 (training) and 4.16 (test). The multisource classifications are shown with several values of weights (reliability factors) in each table. The tables are organized as follows: In the top portion of the tables the single-source classifications are shown. In the boxes below, the multisource classifications are shown with different values of weights. The first box with the multisource classifications shows the result with equal weights and then the results with a uniform but equal decrease in the weights of the topographic sources. The second box shows the results when all the sources except the slope source have equal and

Table 4.15

Statistical Multisource Classification of Colorado Data when Topographic Sources were Modeled by Histogram Approach: Training Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	Single Sources											
MSS	99.3	64.3	20.9	68.6	16.6	85.2	89.8	5.3	28.0	67.3	69.05	54.33
Elevation	98.7	0.0	0.0	80.0	24.2	17.2	99.3	31.6	28.0	98.0	62.15	47.70
Slope	95.0	0.0	0.0	4.3	10.8	27.0	61.9	0.0	4.0	0.0	42.81	20.31
Aspect	96.1	0.0	4.7	51.4	36.9	22.1	48.3	2.6	8.0	34.7	50.15	30.49
	Multiple Sources											
m e s a												
1. 1. 1. 1.	99.7	96.4	20.9	98.6	59.9	48.4	100.0	34.2	60.0	100.0	80.26	71.80
1. .9 .9 .9	99.7	94.6	20.9	98.6	56.1	60.7	100.0	23.7	60.0	100.0	80.65	71.42
1. .8 .8 .8	99.7	91.1	23.3	98.6	50.3	73.4	100.0	23.7	60.0	100.0	81.25	72.03
1. .7 .7 .7	99.7	91.1	23.3	98.6	51.0	77.9	100.0	15.8	56.0	100.0	81.45	71.32
1. .6 .6 .6	99.7	91.1	23.2	97.1	48.4	82.8	100.0	5.3	52.0	100.0	81.05	69.96
1. .5 .5 .5	99.7	87.5	23.3	95.7	45.9	85.2	100.0	0.0	36.0	100.0	80.06	67.32
1. .4 .4 .4	100.0	83.9	23.3	94.3	43.9	86.9	100.0	0.0	32.0	100.0	79.66	66.43
1. .3 .3 .3	100.0	73.2	25.6	92.9	40.8	91.0	100.0	0.0	28.0	100.0	78.97	65.14
1. .2 .2 .2	100.0	73.2	25.6	88.6	38.9	91.0	100.0	0.0	8.0	100.0	77.88	62.52
1. .1 .1 .1	100.0	66.1	25.6	82.9	35.0	91.8	100.0	0.0	8.0	100.0	76.59	60.93
1. .0 .0 .0	100.0	57.1	16.3	65.7	19.8	90.2	89.8	0.0	0.0	67.3	68.65	50.62
1. 1. .9 1.	99.7	96.4	20.9	98.6	58.6	52.5	100.0	26.3	60.0	100.0	80.26	71.30
1. 1. .8 1.	99.7	94.6	20.9	98.6	57.3	54.1	100.0	28.9	60.0	100.0	80.26	71.42
1. 1. .7 1.	99.7	92.9	20.9	98.6	57.3	57.4	100.0	28.9	60.0	100.0	80.56	71.57
1. 1. .6 1.	99.7	91.1	20.9	98.6	57.3	58.2	100.0	31.6	60.0	100.0	80.65	71.73
1. 1. .5 1.	99.7	91.1	20.9	98.6	55.4	64.8	100.0	31.6	60.0	100.0	81.15	72.20
1. 1. .4 1.	99.7	91.1	20.9	98.6	56.1	68.9	100.0	31.6	60.0	100.0	81.75	72.67
1. 1. .3 1.	99.7	91.1	20.9	98.6	52.9	72.1	100.0	31.6	60.0	100.0	81.65	72.68
1. 1. .2 1.	99.7	91.1	20.9	98.6	54.1	73.0	100.0	31.6	60.0	100.0	81.94	72.89
1. 1. .1 1.	99.7	89.2	20.9	97.1	54.1	74.6	100.0	31.6	60.0	100.0	81.94	72.73
1. 1. .0 1.	99.7	87.5	20.9	97.1	53.5	75.4	100.0	31.6	60.0	100.0	81.85	72.57
1. 1. .8 .9	99.7	94.6	23.3	98.6	57.3	60.7	100.0	28.9	60.0	100.0	81.15	72.31
1. .9 .8 .9	99.7	92.9	20.9	98.6	56.7	60.7	100.0	23.7	60.0	100.0	80.65	71.31
1. .9 .7 .8	99.7	91.1	23.3	98.6	51.6	74.6	100.0	26.3	60.0	100.0	81.65	72.51
1. .9 .6 .8	99.7	91.1	23.3	98.6	51.6	75.4	100.0	26.3	60.0	100.0	81.75	72.59
1. .9 .6 .7	99.7	91.1	23.3	97.1	50.3	77.9	100.0	28.9	60.0	100.0	81.85	72.83
1. .9 .5 .7	99.7	91.1	23.3	97.1	49.7	78.7	100.0	28.9	60.0	100.0	81.85	72.85
1. .9 .5 .6	99.7	91.1	23.3	97.1	49.0	81.1	100.0	18.4	56.0	100.0	81.55	71.58
1. .8 .5 .6	99.7	91.1	23.3	97.1	48.4	82.0	100.0	13.2	52.0	100.0	81.25	70.67
1. .8 .4 .6	99.7	89.3	23.3	97.1	47.1	82.0	100.0	7.9	52.0	100.0	80.75	69.83
1. .8 .4 .5	99.7	87.5	23.3	95.7	47.1	84.4	100.0	5.3	52.0	100.0	80.75	69.50
1. .7 .4 .5	99.7	87.5	23.3	95.7	46.5	85.2	100.0	2.6	52.0	100.0	80.65	69.25
# of pixels	301	56	43	70	157	122	147	38	25	49	1008	1008

The columns labeled m e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

CPU time for training and classification: 13 sec.

Table 4.16

Statistical Multisource Classification of Colorado Data when Topographic Sources were Modeled by Histogram Approach: Test Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
						Single Sources						
MSS	100.0	53.6	20.5	54.3	13.4	79.5	89.1	5.3	4.0	54.0	65.08	47.36
Elevation	100.0	0.0	0.0	77.1	22.9	14.8	98.0	26.3	24.0	90.0	60.83	45.31
Slope	95.4	0.0	0.0	5.7	7.6	24.6	55.8	0.0	4.0	0.0	41.25	19.31
Aspect	98.0	0.0	2.3	35.7	34.4	15.6	45.6	0.0	0.0	18.0	46.59	24.95
m e s a						Multiple Sources						
1. 1. 1. 1.	99.3	100.0	18.2	85.7	52.9	49.2	99.3	26.3	44.0	94.0	77.25	66.89
1. .9 .9 .9	99.3	98.2	18.2	85.7	48.4	64.8	99.3	10.5	40.0	94.0	77.65	65.84
1. .8 .8 .8	99.3	98.2	18.2	90.0	42.0	72.1	99.3	10.5	36.0	94.0	77.74	65.97
1. .7 .7 .7	100.0	98.2	18.2	90.0	42.0	77.9	99.3	10.5	32.0	94.0	78.54	66.21
1. .6 .6 .6	100.0	94.6	18.2	90.0	42.7	83.6	99.3	10.5	28.0	96.0	79.13	66.30
1. .5 .5 .5	100.0	94.6	18.2	90.0	40.1	87.7	99.3	5.3	24.0	96.0	78.93	65.52
1. .4 .4 .4	100.0	89.3	20.5	87.1	38.9	87.7	99.3	2.6	12.0	94.0	77.84	63.14
1. .3 .3 .3	100.0	80.4	22.7	85.7	35.7	90.2	99.3	0.0	4.0	94.0	76.85	61.20
1. .2 .2 .2	100.0	75.0	29.5	80.0	35.0	91.8	99.3	0.0	0.0	92.0	76.36	60.27
1. .1 .1 .1	100.0	69.6	27.3	72.9	34.4	89.3	99.3	0.0	0.0	90.0	74.98	58.28
1. .0 .0 .0	100.0	62.5	18.2	55.7	26.1	85.2	91.2	0.0	0.0	52.0	68.15	49.09
1. 1. .9 1.	99.3	98.2	18.2	85.7	50.3	54.1	99.3	23.7	44.0	94.0	77.25	66.69
1. 1. .8 1.	99.3	98.2	18.2	84.3	49.7	59.0	99.3	23.7	44.0	94.0	77.65	66.97
1. 1. .7 1.	99.3	98.2	18.2	85.7	49.7	63.1	99.3	21.1	44.0	94.0	78.14	67.26
1. 1. .6 1.	99.7	98.2	18.2	85.7	48.4	63.9	99.3	18.4	44.0	94.0	78.04	66.99
1. 1. .5 1.	99.7	98.2	18.2	85.7	47.8	65.6	99.3	15.8	44.0	94.0	78.04	66.82
1. 1. .4 1.	100.0	98.2	18.2	85.7	45.2	68.9	99.3	13.2	44.0	96.0	78.14	66.82
1. 1. .3 1.	100.0	96.4	18.2	85.7	46.5	68.9	99.3	13.2	44.0	96.0	78.24	66.82
1. 1. .2 1.	100.0	94.6	18.2	85.7	44.6	73.0	99.3	15.8	44.0	96.0	78.44	67.12
1. 1. .1 1.	100.0	94.6	20.5	87.1	45.2	74.6	99.3	15.8	40.0	96.0	78.83	67.32
1. 1. .0 1.	100.0	94.6	20.5	85.7	45.9	74.6	99.3	15.8	40.0	96.0	78.83	67.24
1. 1. .8 .9	99.7	98.2	18.2	85.7	47.8	64.8	99.3	21.1	40.0	94.0	78.04	66.87
1. .9 .8 .9	99.3	98.2	18.2	87.1	47.1	64.8	99.3	13.2	40.0	94.0	77.65	66.12
1. .9 .7 .8	100.0	98.2	18.2	88.6	43.9	72.1	99.3	10.5	40.0	94.0	78.24	66.49
1. .9 .6 .8	100.0	98.2	18.2	88.6	43.3	74.6	99.3	10.5	36.0	96.0	78.44	66.47
1. .9 .6 .7	100.0	96.4	18.2	87.1	42.7	76.2	99.3	10.5	36.0	96.0	78.34	66.25
1. .9 .5 .7	100.0	96.4	18.2	87.1	42.7	77.0	99.3	10.5	36.0	96.0	78.44	66.33
1. .9 .5 .6	100.0	94.6	18.2	88.6	41.4	80.3	99.3	13.2	36.0	96.0	78.73	66.76
1. .8 .5 .6	100.0	94.6	18.2	90.0	41.4	80.3	99.3	10.5	32.0	96.0	78.64	66.24
1. .8 .4 .6	100.0	94.6	20.5	90.0	42.0	82.0	99.3	10.5	32.0	96.0	79.03	66.69
1. .8 .4 .5	100.0	94.6	20.5	90.0	41.4	84.4	99.3	10.5	32.0	96.0	79.23	66.88
1. .7 .4 .5	100.0	92.9	20.5	90.0	40.8	85.2	99.3	10.5	32.0	96.0	79.13	66.72
# of pixels	302	56	44	70	157	122	147	38	25	50	1011	1011

The columns labeled m e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

full weights. The weight of the slope source is varied from 1 to 0 to see the effect of including this data source in the classification. The last box in the tables shows the classification accuracies when reliability measures are used to select the weights.

Looking at the single-source classifications in Table 4.15 and using the overall classification accuracy as the reliability measure, it is seen that the MSS source is the most reliable source, elevation ranks second, aspect third and slope fourth. That is the same ranking given by the equivocation measure shown in Tables 4.17 (Landsat MSS data) and 4.18 (topographic data).

When all the data sources were classified with equal weights the overall accuracy for the training data improved to 80.26% which was over 11% better than the best single-source classification (Landsat MSS: 69.05%). The average classification accuracy also improved greatly (71.80%), more than 17% better than the best average single-source classification (Landsat MSS: 54.33%). Reducing the weights of the less reliable sources improved the classification accuracy as long as the selected weights were not too low. The "best" overall and average accuracies were achieved when the MSS, elevation and aspect were given full weights (1.) and the slope weight was reduced to 0.2. The overall accuracy with these weights was 81.94% which is 1.65% higher than the overall accuracy when all the sources had equal weights. These weights gave average accuracy of 72.89% which was an improvement of just over 1% compared to the classification with equal weights. Several other weights gave good results as shown in Table 4.15. For the most part the results show that when a source with a low class-specific accuracy is decreased in weight the classification accuracy of the class goes up.

Table 4.17

Equivocation of MSS Data Source.

Data Source	Equivocation
Landsat MSS	0.800

Table 4.18

Equivocation of Topographic Data Sources with Respect to Different Modeling Methods.

Data Source	Histogram Estimation	Maximum Penalized Likelihood Method	Parzen Estimation
Elevation	1.058	1.054	1.056
Slope	1.687	1.687	1.687
Aspect	1.550	1.560	1.556

As shown in Table 4.16 the classification accuracy of the test data was improved from the single-source classification when the data were combined. As for the training data, the Landsat MSS source had the highest overall and average classification accuracy (65.08% and 47.36%, respectively). When all the data sources were classified with equal weights, these accuracies increased to 77.25% and 66.89% or by more than 12% for the overall accuracy and nearly 20% for the average accuracy (as compared to the Landsat MSS classifications). By changing the weights, both the overall and average accuracies were improved. The highest overall accuracy for test data was reached when the MSS source had full weight, the elevation source had a weight of 0.8, slope the weight 0.4 and aspect the weight 0.5. This weighting was suggested by the reliability measures and gave overall accuracy of 79.23% and average accuracy of 66.88% With these weights the overall accuracy increased by nearly 2% compared to the result with equal weights, but the average accuracy stayed almost the same. The highest average accuracy for the test data was achieved when the slope was given a weight of 0.1 and all the other sources were given full weights. The average accuracy achieved by this weighting was 67.32%, which is an increase of 0.43% from the equal weights result.

The results using the LOP are shown in Tables 4.19 and 4.20. These results are clearly inferior to those obtained for the SMC. The LOP is especially poor in accurate classification of classes with low prior probabilities. It is also seen that equal weights are questionable for this classification method. When the training data were combined with equal weights (Table 4.19), the results were an overall accuracy of 68.15% and an average accuracy

Table 4.19

Linear Opinion Pool Applied to Colorado Data Set. Topographic Sources were Modeled by Histogram Approach: Training Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
						Single Sources						
MSS	99.3	64.3	20.9	68.6	16.6	85.2	89.8	5.3	28.0	67.3	69.05	54.33
Elevation	98.7	0.0	0.0	80.0	24.2	17.2	99.3	31.6	28.0	98.0	62.15	47.70
Slope	95.0	0.0	0.0	4.3	10.8	27.0	61.9	0.0	4.0	0.0	42.81	20.31
Aspect	96.1	0.0	4.7	51.4	36.9	22.1	48.3	2.6	8.0	34.7	50.15	30.49
m e s a						Multiple Sources						
1. 1. 1. 1.	100.0	0.0	0.0	91.4	39.5	50.0	100.0	0.0	12.0	100.0	68.15	49.29
1. .9 .9 .9	100.0	0.0	0.0	91.4	39.5	50.0	100.0	0.0	12.0	100.0	68.15	49.29
1. .8 .8 .8	100.0	0.0	0.0	92.9	39.5	50.0	100.0	0.0	12.0	100.0	68.25	49.43
1. .7 .7 .7	100.0	0.0	0.0	92.9	39.5	51.6	100.0	0.0	12.0	100.0	68.45	49.60
1. .6 .6 .6	100.0	0.0	0.0	94.3	38.9	53.3	100.0	0.0	8.0	100.0	68.55	49.44
1. .5 .5 .5	100.0	0.0	16.3	94.3	36.3	53.3	100.0	0.0	0.0	100.0	68.65	50.01
1. .4 .4 .4	100.0	0.0	16.3	91.4	37.6	85.2	100.0	0.0	0.0	100.0	72.52	53.05
1. .3 .3 .3	100.0	26.8	16.3	91.4	40.8	90.2	99.3	0.0	0.0	100.0	75.00	56.47
1. .2 .2 .2	100.0	51.8	16.3	84.3	43.9	91.0	99.3	0.0	0.0	91.8	76.09	57.84
1. .1 .1 .1	100.0	55.4	16.3	82.9	33.1	90.2	93.2	0.0	0.0	73.5	72.62	54.44
1. .0 .0 .0	100.0	57.1	16.3	65.7	19.8	90.2	89.8	0.0	0.0	67.3	68.65	50.62
1. 1. .9 1.	100.0	0.0	0.0	92.9	39.5	50.0	100.0	0.0	12.0	100.0	68.25	49.43
1. 1. .8 1.	100.0	0.0	0.0	94.3	38.9	50.8	100.0	0.0	16.0	100.0	68.45	50.00
1. 1. .7 1.	100.0	0.0	0.0	94.3	38.2	50.8	100.0	0.0	16.0	100.0	68.35	49.93
1. 1. .6 1.	100.0	0.0	0.0	94.3	38.2	50.8	100.0	0.0	16.0	100.0	68.35	49.93
1. 1. .5 1.	100.0	0.0	0.0	94.3	37.6	50.8	100.0	0.0	16.0	100.0	68.25	49.87
1. 1. .4 1.	100.0	0.0	0.0	94.3	37.6	50.8	100.0	0.0	16.0	100.0	68.25	49.87
1. 1. .3 1.	100.0	0.0	0.0	94.3	37.6	50.8	100.0	0.0	20.0	100.0	68.35	50.27
1. 1. .2 1.	100.0	0.0	0.0	94.3	37.6	50.8	100.0	0.0	20.0	100.0	68.35	50.27
1. 1. .1 1.	100.0	0.0	0.0	94.3	37.6	50.8	100.0	0.0	20.0	100.0	68.35	50.27
1. 1. .0 1.	100.0	0.0	0.0	94.3	38.2	50.8	100.0	0.0	24.0	100.0	68.55	50.73
1. 1. .8 .9	100.0	0.0	0.0	94.3	38.2	50.0	100.0	0.0	16.0	100.0	68.25	49.85
1. .9 .8 .9	100.0	0.0	0.0	92.9	39.5	50.0	100.0	0.0	12.0	100.0	68.25	49.43
1. .9 .7 .8	100.0	0.0	0.0	94.3	38.2	50.0	100.0	0.0	12.0	100.0	68.15	49.45
1. .9 .6 .8	100.0	0.0	0.0	94.3	38.2	50.8	100.0	0.0	16.0	100.0	68.35	49.93
1. .9 .6 .7	100.0	0.0	0.0	94.3	38.2	51.6	100.0	0.0	16.0	100.0	68.45	50.01
1. .9 .5 .7	100.0	0.0	0.0	94.3	37.6	51.6	100.0	0.0	20.0	100.0	68.45	50.35
1. .9 .5 .6	100.0	0.0	0.0	94.3	37.6	51.6	100.0	0.0	20.0	100.0	68.45	50.35
1. .8 .5 .6	100.0	0.0	0.0	94.3	38.2	51.7	100.0	0.0	16.0	100.0	68.45	50.01
1. .8 .4 .6	100.0	0.0	0.0	94.3	37.6	51.6	100.0	0.0	16.0	100.0	68.35	49.95
1. .8 .4 .5	100.0	0.0	0.0	92.9	36.3	52.5	100.0	0.0	8.0	100.0	67.96	48.96
1. .7 .4 .5	100.0	0.0	9.3	92.9	37.6	52.5	100.0	0.0	12.0	100.0	68.65	50.42
# of pixels	301	56	43	70	157	122	147	38	25	49	1008	1008

The columns labeled m e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

CPU time for training and classification: 11 sec.

Table 4.20

Linear Opinion Applied to Colorado Data Set. Topographic Sources were Modeled by Histogram Approach: Test Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	Single Sources											
MSS	100.0	53.6	20.5	54.3	13.4	79.5	89.1	5.3	4.0	54.0	65.08	47.36
Elevation	100.0	0.0	0.0	77.1	22.9	14.8	98.0	26.3	24.0	90.0	60.83	45.31
Slope	95.4	0.0	0.0	5.7	7.6	24.6	55.8	0.0	4.0	0.0	41.25	19.31
Aspect	98.0	0.0	2.3	35.7	34.4	15.6	45.6	0.0	0.0	18.0	46.59	24.95
m e s a	Multiple Sources											
1. 1. 1. 1.	100.0	0.0	0.0	88.6	36.9	51.6	100.0	0.0	0.0	88.0	66.86	46.52
1. .9 .9 .9	100.0	0.0	0.0	85.6	36.3	54.1	100.0	0.0	0.0	90.0	67.16	46.90
1. .8 .8 .8	100.0	0.0	0.0	88.6	34.4	54.9	100.0	0.0	0.0	90.0	66.96	46.79
1. .7 .7 .7	100.0	0.0	0.0	87.2	33.8	55.7	100.0	0.0	0.0	90.0	66.86	46.66
1. .6 .6 .6	100.0	0.0	0.0	87.1	34.4	56.6	100.0	0.0	0.0	92.0	67.16	47.01
1. .5 .5 .5	100.0	0.0	13.6	85.7	32.5	57.4	100.0	0.0	0.0	92.0	67.46	48.12
1. .4 .4 .4	100.0	0.0	15.9	85.7	33.1	84.4	100.0	0.0	0.0	92.0	70.92	51.11
1. .3 .3 .3	100.0	30.4	18.2	82.9	37.6	86.9	100.0	0.0	0.0	90.0	73.39	54.59
1. .2 .2 .2	100.0	50.0	18.2	74.3	43.9	86.9	98.0	0.0	0.0	74.0	73.79	54.53
1. .1 .1 .1	100.0	58.9	18.2	67.1	33.1	86.1	92.5	0.0	0.0	68.0	70.92	52.40
1. .0 .0 .0	100.0	62.5	18.2	55.7	26.1	85.2	91.2	0.0	0.0	52.0	68.15	49.09
1. 1. .9 1.	100.0	0.0	0.0	88.6	37.6	52.5	100.0	0.0	0.0	90.0	67.16	46.86
1. 1. .8 1.	100.0	0.0	0.0	88.6	37.6	52.5	100.0	0.0	0.0	90.0	67.16	46.86
1. 1. .7 1.	100.0	0.0	0.0	88.6	36.9	52.5	100.0	0.0	0.0	90.0	67.06	46.78
1. 1. .6 1.	100.0	0.0	0.0	90.0	36.3	52.5	100.0	0.0	0.0	90.0	67.06	46.88
1. 1. .5 1.	100.0	0.0	0.0	90.0	36.3	52.5	100.0	0.0	0.0	90.0	67.06	46.88
1. 1. .4 1.	100.0	0.0	0.0	90.0	36.3	51.6	100.0	0.0	0.0	90.0	66.96	46.79
1. 1. .3 1.	100.0	0.0	0.0	88.6	35.7	51.6	100.0	0.0	4.0	92.0	66.96	47.19
1. 1. .2 1.	100.0	0.0	0.0	88.6	35.0	51.6	100.0	0.0	8.0	92.0	66.96	47.52
1. 1. .1 1.	100.0	0.0	0.0	88.6	35.7	51.6	100.0	0.0	8.0	92.0	67.06	47.59
1. 1. .0 1.	100.0	0.0	0.0	88.6	33.8	51.6	100.0	0.0	12.0	92.0	66.86	47.80
1. 1. .8 .9	100.0	0.0	0.0	88.6	35.7	53.3	100.0	0.0	0.0	90.0	66.96	46.75
1. .9 .8 .9	100.0	0.0	0.0	88.6	36.3	53.3	100.0	0.0	0.0	90.0	67.06	46.82
1. .9 .7 .8	100.0	0.0	0.0	90.0	35.7	54.1	100.0	0.0	0.0	90.0	67.16	46.98
1. .9 .6 .8	100.0	0.0	0.0	87.1	35.0	54.9	100.0	0.0	0.0	92.0	67.06	46.91
1. .9 .6 .7	100.0	0.0	0.0	88.6	35.0	55.7	100.0	0.0	4.0	92.0	67.36	47.53
1. .9 .5 .7	100.0	0.0	0.0	90.0	34.4	54.9	100.0	0.0	4.0	92.0	67.26	47.53
1. .9 .5 .6	100.0	0.0	0.0	88.6	33.1	54.9	100.0	0.0	8.0	92.0	67.06	47.66
1. .8 .5 .6	100.0	0.0	0.0	90.0	33.1	54.1	100.0	0.0	4.0	92.0	66.96	47.32
1. .8 .4 .6	100.0	0.0	0.0	90.0	34.4	54.9	100.0	0.0	4.0	92.0	67.26	47.53
1. .8 .4 .5	100.0	0.0	0.0	87.1	31.8	56.6	100.0	0.0	4.0	92.0	66.87	47.15
1. .7 .4 .5	100.0	0.0	11.4	88.6	31.2	55.7	100.0	0.0	0.0	92.0	67.16	47.89
# of pixels	302	56	44	70	157	122	147	38	25	50	1011	1011

The columns labeled m e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

of 49.29%, both lower than the results achieved in the single-source classification of the Landsat MSS data. However, the accuracies could be improved by lowering the weights on the less reliable sources. The highest overall and average accuracies in Table 4.19 were achieved when the Landsat MSS source had full weight and all the other sources were given the weight 0.2. The overall accuracy with these weights was 76.09%, about 7% better than the best single-source classification. The average accuracy was 57.84%, about 3.5% better than the one for Landsat MSS. As noted above, these results were worse than the ones achieved with the SMC. It is also more difficult to see any similar behavior for the LOP as compared to the SMC when a source was given a lower weight and had a low classification accuracy. In contrast to the SMC, that type of weight selection did not mean that the accuracy for multiple sources would improve.

The test results using the LOP (Table 4.20) were similar to the training results in most cases, although the overall accuracy when equal weights were used was better than the best single-source classification. The overall accuracy improved by 1.78% but the average accuracy decreased by 0.84% as compared to the Landsat MSS result. The highest overall accuracy was achieved when the Landsat source had full weight and all the topographic sources were given the weight 0.3. This highest overall accuracy was 73.79%, an improvement of 6.93% as compared to the combination result with equal weights. This particular weighting gave an average accuracy of 54.53% which was close to the highest average accuracy in Table 4.20 (54.59%). The average accuracy could thus be improved by over 8.0% as compared to the equal weights case. As noted earlier, the results using the LOP were clearly worse than the ones using the SMC. However, it is also evident from the results in Tables 4.19 and 4.20 that the weighting of the sources is more important in the LOP than in the SMC.

b) Topographic Data Modeled by the Maximum Penalized Likelihood Method

The topographic data were modeled by the maximum penalized likelihood method, with all the topographic sources given a smoothing parameter (γ) of 10. That value of γ gave the best classification results. The maximum penalized likelihood estimation was done using the IMSL subroutine D3SPL. This subroutine uses $\int(f''(t))^2dt$ as its roughness term R(f). The results of SMC classifications are shown in Tables 4.21 and 4.22. The results were similar to the histogram modeling for source specific classifications in Tables 4.15 and 4.16. However, as seen in the tables the maximum penalized likelihood method did a better job of modeling the aspect data than the histogram approach. The rankings of the sources were the same as with the histogram method: 1. Landsat MSS, 2. elevation, 3. aspect and 4. slope. This was indicated both by the source-specific classifications in Table 4.21 and the equivocations in Tables 4.17 and 4.18.

When the sources were combined with equal weights the result (Table 4.21) was the same as with the histogram approach in terms of overall accuracy of training data (80.26%). The average accuracy was 71.54%, which was slightly below the average accuracy of the histogram approach (71.80%). Somewhat surprisingly the highest overall accuracy of training data was

Table 4.21

Statistical Multisource Classification of Colorado Data when Topographic Sources were Modeled by Maximum Penalized Likelihood Method: Training Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	Single Sources											
MSS	99.3	64.3	20.9	68.6	16.6	85.2	89.8	5.3	28.0	67.3	69.05	54.33
Elevation	100.0	0.0	0.0	74.3	24.2	17.2	98.6	34.2	36.0	100.0	62.30	48.46
Slope	95.3	0.0	0.0	4.3	9.6	27.0	61.2	0.0	8.0	0.0	42.66	20.55
Aspect	98.3	0.0	4.7	50.0	37.6	20.5	51.0	10.5	8.0	22.4	50.50	30.31
m e s a	Multiple Sources											
1. 1. 1. 1.	100.0	96.4	23.3	97.1	61.1	48.4	100.0	21.1	68.0	100.0	80.26	71.54
1. .9 .9 .9	100.0	96.4	23.5	97.1	61.8	53.3	100.0	21.1	64.0	100.0	80.85	71.69
1. .8 .8 .8	100.0	94.6	23.3	97.1	59.9	62.3	100.0	18.4	60.0	100.0	81.35	71.56
1. .7 .7 .7	100.0	91.1	23.3	97.2	54.1	74.6	100.0	10.5	52.0	100.0	81.25	70.27
1. .6 .6 .6	100.0	91.1	23.3	95.7	52.2	81.1	100.0	2.6	52.0	100.0	81.35	69.80
1. .5 .5 .5	100.0	89.3	25.6	95.7	49.0	86.1	100.0	0.0	32.0	100.0	80.85	67.77
1. .4 .4 .4	100.0	83.9	25.6	92.9	44.6	86.9	100.0	0.0	32.0	100.0	79.76	66.58
1. .3 .3 .3	100.0	76.8	25.6	92.9	43.3	90.2	100.0	0.0	16.0	100.0	79.17	64.47
1. .2 .2 .2	100.0	73.2	25.6	88.6	41.4	91.0	100.0	0.0	8.0	100.0	78.27	62.78
1. .1 .1 .1	100.0	67.9	25.6	82.9	35.0	91.8	97.3	0.0	8.0	100.0	76.29	60.84
1. .0 .0 .0	100.0	57.1	16.3	65.7	19.8	90.2	89.8	0.0	0.0	67.3	68.65	50.62
1. 1. .9 1.	100.0	96.4	23.3	97.1	61.1	48.4	100.0	21.1	68.0	100.0	80.26	71.54
1. 1. .8 1.	100.0	96.4	23.3	97.1	61.8	49.2	100.0	21.1	68.0	100.0	80.46	71.68
1. 1. .7 1.	100.0	96.4	23.3	97.1	63.1	51.6	100.0	21.1	64.0	100.0	80.85	71.66
1. 1. .6 1.	100.0	94.6	23.3	97.1	61.1	54.1	100.0	23.7	64.0	100.0	80.85	71.80
1. 1. .5 1.	100.0	92.9	23.3	97.1	60.5	56.6	100.0	23.7	64.0	100.0	80.95	71.80
1. 1. .4 1.	100.0	91.1	23.3	97.1	59.2	59.8	100.0	23.7	64.0	100.0	81.05	71.82
1. 1. .3 1.	100.0	91.1	23.3	97.1	56.7	64.8	100.0	26.3	64.0	100.0	81.35	72.32
1. 1. .2 1.	100.0	91.1	23.3	95.7	55.4	70.5	100.0	26.3	64.0	100.0	81.75	72.63
1. 1. .1 1.	100.0	91.1	23.3	94.3	53.5	73.0	100.0	26.3	64.0	100.0	81.65	72.54
1. 1. .0 1.	100.0	91.1	25.6	94.3	53.5	75.4	100.0	23.7	60.0	100.0	81.85	72.35
1. 1. .8 .9	100.0	96.4	23.3	97.1	60.5	54.1	100.0	23.7	64.0	100.0	80.85	71.91
1. .9 .8 .9	100.0	96.4	23.3	97.1	60.5	56.6	100.0	21.1	64.0	100.0	81.05	71.89
1. .9 .7 .8	100.0	92.9	23.3	97.1	59.2	62.3	100.0	21.1	64.0	100.0	81.35	71.98
1. .9 .6 .8	100.0	91.1	23.3	97.1	56.7	70.5	100.0	18.4	64.0	100.0	81.75	72.11
1. .9 .6 .7	100.0	91.1	23.3	97.1	52.9	74.6	100.0	18.4	60.0	100.0	81.55	71.73
1. .9 .5 .7	100.0	91.1	23.3	97.1	52.2	76.2	100.0	15.8	56.0	100.0	81.45	71.17
1. .9 .5 .6	100.0	91.1	25.6	97.1	50.3	80.3	100.0	7.9	56.0	100.0	81.45	70.83
1. .8 .5 .6	100.0	91.1	25.6	95.7	49.7	80.3	100.0	5.3	52.0	100.0	81.05	69.96
1. .8 .4 .6	100.0	91.1	25.6	95.7	51.0	80.3	100.0	2.6	52.0	100.0	81.15	69.83
1. .8 .4 .5	100.0	91.1	25.6	95.7	49.0	83.6	100.0	2.6	52.0	100.0	81.25	69.97
1. .7 .4 .5	100.0	89.3	25.6	95.7	48.4	84.4	100.0	0.0	44.0	100.0	80.85	68.74
# of pixels	301	56	43	70	157	122	147	38	25	49	1008	1008

The columns labeled m e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

CPU time for training and classification: 102 sec.

Table 4.22

Statistical Multisource Classification of Colorado Data when Topographic Sources were Modeled by Maximum Penalized Likelihood Method: Test Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	Single Sources											
MSS	100.0	53.6	20.5	54.3	13.4	79.5	89.1	5.3	4.0	54.0	65.08	47.36
Elevation	100.0	0.0	0.0	68.6	22.9	14.8	98.0	31.6	32.0	92.0	60.73	45.98
Slope	95.4	0.0	0.0	5.7	7.6	24.6	54.4	0.0	8.0	0.0	41.15	19.57
Aspect	98.0	0.0	2.3	35.7	34.4	14.8	49.0	7.9	0.0	16.0	47.18	25.80
m e s a	Multiple Sources											
1. 1. 1. 1.	100.0	100.0	18.2	88.6	58.0	48.4	99.3	10.5	44.0	94.0	77.74	66.09
1. .9 .9 .9	100.0	100.0	18.2	90.0	55.4	56.6	99.3	10.5	40.0	96.0	78.44	66.60
1. .8 .8 .8	100.0	100.0	18.2	88.6	51.0	70.5	99.3	10.5	28.0	96.0	79.03	66.20
1. .7 .7 .7	100.0	98.2	20.5	90.0	45.2	76.2	99.3	10.5	28.0	96.0	78.93	66.40
1. .6 .6 .6	100.0	96.4	20.5	90.0	45.6	83.6	99.3	5.3	24.0	96.0	79.53	66.09
1. .5 .5 .5	100.0	94.6	22.7	90.0	41.4	88.5	99.3	5.3	16.0	96.0	79.23	65.39
1. .4 .4 .4	100.0	91.1	22.7	85.7	40.1	88.5	99.3	0.0	4.0	96.0	78.04	62.75
1. .3 .3 .3	100.0	82.1	29.5	84.3	40.1	91.0	99.3	0.0	0.0	96.0	77.94	62.24
1. .2 .2 .2	100.0	75.0	29.5	80.0	36.9	91.0	99.3	0.0	0.0	92.0	76.56	60.38
1. .1 .1 .1	100.0	69.6	27.3	72.9	36.3	89.3	97.3	0.0	0.0	92.0	75.07	58.47
1. .0 .0 .0	100.0	62.5	18.2	55.7	26.1	85.2	91.2	0.0	0.0	52.0	68.15	49.09
1. 1. .9 1.	100.0	100.0	18.2	88.6	58.6	49.2	99.3	10.5	40.0	96.0	77.94	66.04
1. 1. .8 1.	100.0	100.0	18.2	88.6	59.2	51.6	99.3	10.5	40.0	96.0	78.34	66.35
1. 1. .7 1.	100.0	100.0	18.2	88.6	55.4	55.7	99.3	10.5	40.0	96.0	77.24	66.38
1. 1. .6 1.	100.0	100.0	18.2	88.6	52.2	62.3	99.3	10.5	44.0	96.0	78.64	67.11
1. 1. .5 1.	100.0	98.2	18.2	87.1	51.6	68.0	99.3	10.5	44.0	96.0	79.03	67.30
1. 1. .4 1.	100.0	98.2	20.5	87.1	50.3	68.9	99.3	15.8	44.0	96.0	79.23	68.01
1. 1. .3 1.	100.0	98.2	20.5	87.1	49.7	69.7	99.3	15.8	40.0	96.0	79.13	67.63
1. 1. .2 1.	100.0	98.2	20.5	87.1	47.8	72.1	99.3	15.8	40.0	96.0	79.13	67.68
1. 1. .1 1.	100.0	96.4	20.5	87.1	46.5	74.6	99.3	15.8	40.0	96.0	79.13	67.62
1. 1. .0 1.	100.0	94.6	22.7	87.1	47.1	73.8	99.3	15.8	40.0	96.0	79.13	67.65
1. 1. .8 .9	100.0	100.0	18.2	88.6	54.8	56.6	99.3	10.5	40.0	96.0	78.24	66.39
1. .9 .8 .9	100.0	100.0	18.2	88.6	52.2	61.5	99.3	10.5	40.0	96.0	78.44	66.63
1. .9 .7 .8	100.0	98.2	20.5	88.6	48.4	69.7	99.3	10.5	36.0	96.0	78.73	66.71
1. .9 .6 .8	100.0	98.2	20.5	88.6	44.6	73.8	99.3	10.5	32.0	96.0	78.54	66.34
1. .9 .6 .7	100.0	98.2	20.5	90.0	45.9	75.4	99.3	10.5	32.0	96.0	79.03	66.78
1. .9 .5 .7	100.0	98.2	20.5	90.0	45.2	78.7	99.3	10.5	32.0	96.0	79.33	67.04
1. .9 .5 .6	100.0	96.4	22.7	88.6	45.2	82.0	99.3	10.5	32.0	96.0	79.62	67.28
1. .8 .5 .6	100.0	96.4	22.7	88.6	45.2	82.8	99.3	10.5	28.0	96.0	77.62	66.96
1. .8 .4 .6	100.0	96.4	22.7	88.6	45.9	84.4	99.3	10.5	28.0	96.0	79.92	67.19
1. .8 .4 .5	100.0	94.6	22.7	90.0	43.9	87.7	99.3	10.5	28.0	96.0	80.02	67.29
1. .7 .4 .5	100.0	94.6	22.7	90.0	42.7	87.7	99.3	5.3	28.0	96.0	79.62	66.63
# of pixels	302	56	44	70	157	122	147	38	25	50	1011	1011

The columns labeled m e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

reached when all the sources except the slope source were given full weights and the slope source was given zero weight. This highest overall accuracy was 81.85%, slightly below the highest overall accuracy of training data reached by the histogram approach (81.94%). The histogram approach also gave a better result in terms of average accuracy.

The test results using SMC are shown in Table 4.22. Looking at the combination result, it is clear that the SMC with the maximum penalized likelihood method outperformed the SMC with the histogram approach in terms of overall classification accuracy of test data. When the sources were combined with equal weights, the overall classification accuracy in Table 4.22 was 77.74% an increase of 12.66% as compared to the best single-source classification. It was also 0.49% higher than the comparable SMC with histogram result. However, the histogram approach (Table 4.16) gave a 0.80% better result in terms of average accuracy. When the weights were varied, the maximum penalized likelihood method gave a better result as compared to the histogram combination both for overall accuracy and average accuracy. The best overall accuracy result in Table 4.22 was reached with the same "best" weights as in Table 4.16. Those weights were indicated by the reliability measures (MSS:1.0, elevation:0.8, slope:0.4, aspect:0.5) and gave overall accuracy of 80.02% and average accuracy of 67.29%. The overall accuracy was increased by 2.28% and the average accuracy by 1.2% as compared to the equal weights classification. Both these results were better than the ones achieved with the histogram combination. The best average accuracy achieved in Table 4.22 was 68.01% when all the sources except the slope had full weights, and the slope was given the weight 0.4.

The results for the LOP with the maximum penalized likelihood method are shown in Tables 4.23 and 4.24. The training result (Table 4.23) was very similar to the result with the histograms (Table 4.19). However, the LOP with the maximum penalized likelihood method reached a higher overall accuracy than its counterpart with the histogram method. When the Landsat MSS source was given a full weight and all the other sources were given the weight 0.2, the overall accuracy reached 76.19% which was 0.10% over the "best" result (same weights) with the histogram approach. For most of the weights the histogram combination did better in terms of higher average accuracy of training data as compared to the maximum penalized likelihood method.

Looking at the LOP test results in Table 4.24, it is seen that the LOP with the maximum penalized likelihood approach did a little better in terms of overall accuracy as compared to the LOP with the histogram approach in Table 4.20. When equal weights were used, the overall accuracy with the maximum penalized likelihood method was 67.06% as compared to 66.86% with the histogram approach. The average accuracy was the same (46.52%). When the weights were changed, the overall accuracy improved to 73.79%, the same result achieved with the same weights for the histogram method. The average accuracy was almost the same, although a little higher in the histogram result (0.06% difference). For the most part the results in Tables 4.24 and 4.20 were very similar. The maximum penalized likelihood modeling could not improve the classification accuracy of test data as much as it did with the SMC.

Table 4.23

Linear Opinion Pool Applied to Colorado Data Set when Topographic Sources were Modeled by Maximum Penalized Likelihood Method: Training Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
						Single Sources						
MSS	99.3	64.3	20.9	68.6	16.6	85.2	89.8	5.3	28.0	67.3	69.05	54.33
Elevation	100.0	0.0	0.0	74.3	24.2	17.2	98.6	34.2	36.0	100.0	62.30	48.46
Slope	95.3	0.0	0.0	4.3	9.6	27.0	61.2	0.0	8.0	0.0	42.66	20.55
Aspect	98.3	0.0	4.7	50.0	37.6	20.5	51.0	10.5	8.0	22.4	50.50	30.31
m e s a						Multiple Sources						
1. 1. 1. 1.	100.0	0.0	0.0	88.6	41.4	50.8	100.0	0.0	0.0	100.0	68.06	48.10
1. .9 .9 .9	100.0	0.0	0.0	90.0	41.4	51.6	100.0	0.0	0.0	100.0	68.25	48.30
1. .8 .8 .8	100.0	0.0	0.0	90.0	42.0	50.8	100.0	0.0	0.0	100.0	68.25	48.29
1. .7 .7 .7	100.0	0.0	0.0	90.0	41.4	52.5	100.0	0.0	0.0	100.0	68.35	48.39
1. .6 .6 .6	100.0	0.0	0.0	90.0	41.4	52.5	100.0	0.0	0.0	100.0	68.35	48.39
1. .5 .5 .5	100.0	0.0	14.0	91.4	40.1	53.3	100.0	0.0	0.0	100.0	68.95	49.88
1. .4 .4 .4	100.0	0.0	16.3	91.4	37.6	81.1	100.0	0.0	0.0	100.0	72.02	52.64
1. .3 .3 .3	100.0	19.6	16.3	90.0	40.1	90.2	99.3	0.0	0.0	100.0	74.31	55.35
1. .2 .2 .2	100.0	51.8	16.3	84.3	46.5	90.2	99.3	0.0	0.0	98.0	76.19	57.61
1. .1 .1 .1	100.0	55.4	16.3	82.9	33.1	90.2	92.5	0.0	0.0	87.8	72.52	54.38
1. .0 .0 .0	100.0	57.1	16.2	65.7	19.2	90.2	89.8	0.0	0.0	67.3	68.65	50.62
1. 1. .9 1.	100.0	0.0	0.0	90.0	41.4	50.8	100.0	0.0	0.0	100.0	68.15	48.22
1. 1. .8 1.	100.0	0.0	0.0	90.0	41.4	50.8	100.0	0.0	4.0	100.0	68.25	48.62
1. 1. .7 1.	100.0	0.0	0.0	91.4	41.4	50.8	100.0	0.0	4.0	100.0	68.35	48.76
1. 1. .6 1.	100.0	0.0	0.0	94.3	42.0	50.8	100.0	0.0	8.0	100.0	68.75	49.51
1. 1. .5 1.	100.0	0.0	0.0	94.3	42.0	50.8	100.0	0.0	8.0	100.0	68.75	49.51
1. 1. .4 1.	100.0	0.0	0.0	94.3	42.0	50.8	100.0	0.0	8.0	100.0	68.75	49.51
1. 1. .3 1.	100.0	0.0	0.0	94.3	42.0	50.8	100.0	0.0	8.0	100.0	68.75	49.51
1. 1. .2 1.	100.0	0.0	0.0	94.3	42.0	50.8	100.0	0.0	8.0	100.0	68.75	49.51
1. 1. .1 1.	100.0	0.0	0.0	94.3	42.0	50.8	100.0	0.0	8.0	100.0	68.75	49.51
1. 1. .0 1.	100.0	0.0	0.0	94.3	42.0	50.8	100.0	0.0	8.0	100.0	68.75	49.51
1. 1. .8 .9	100.0	0.0	0.0	91.4	41.4	50.8	100.0	0.0	0.0	100.0	68.25	48.36
1. .9 .8 .9	100.0	0.0	0.0	90.0	41.4	51.6	100.0	0.0	0.0	100.0	68.25	48.30
1. .9 .7 .8	100.0	0.0	0.0	92.9	41.4	50.8	100.0	0.0	0.0	100.0	68.35	48.51
1. .9 .6 .8	100.0	0.0	0.0	92.9	40.8	50.8	100.0	0.0	0.0	100.0	68.25	48.44
1. .9 .6 .7	100.0	0.0	0.0	91.4	40.8	52.5	100.0	0.0	0.0	100.0	68.35	48.47
1. .9 .5 .7	100.0	0.0	0.0	91.4	40.8	52.5	100.0	0.0	4.0	100.0	68.45	48.87
1. .9 .5 .6	100.0	0.0	0.0	91.4	40.8	52.5	100.0	0.0	0.0	100.0	68.35	48.47
1. .8 .5 .6	100.0	0.0	0.0	91.4	40.8	52.5	100.0	0.0	0.0	100.0	68.35	48.47
1. .8 .4 .6	100.0	0.0	0.0	92.9	40.8	52.5	100.0	0.0	0.0	100.0	68.45	48.61
1. .8 .4 .5	100.0	0.0	0.0	91.4	40.8	53.3	100.0	0.0	0.0	100.0	68.45	48.55
1. .7 .4 .5	100.0	0.0	7.0	91.4	40.1	53.3	100.0	0.0	0.0	100.0	68.65	49.18
# of pixels	301	56	43	70	157	122	147	38	25	49	1008	1008

The columns labeled m e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

CPU time for training and classification: 100 sec.

Table 4.24

Linear Opinion Pool Applied to Colorado Data Set
when Topographic Sources were Modeled by Maximum Penalized
Likelihood Method: Test Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	Single Sources											
MSS	100.0	53.6	20.5	54.3	13.4	79.5	89.1	5.3	4.0	54.0	65.08	47.36
Elevation	100.0	0.0	0.0	68.6	22.9	14.8	98.0	31.6	32.0	92.0	60.73	45.98
Slope	95.4	0.0	0.0	5.7	7.6	24.6	54.4	0.0	8.0	0.0	41.15	19.57
Aspect	98.0	0.0	2.3	35.7	34.4	14.8	49.0	7.9	0.0	16.0	47.18	25.80
	Multiple Sources											
m e s a												
1. 1. 1. 1.	100.0	0.0	0.0	82.9	38.2	54.1	100.0	0.0	0.0	90.0	67.06	46.52
1. .9 .9 .9	100.0	0.0	0.0	82.9	38.2	54.1	100.0	0.0	0.0	90.0	67.06	46.52
1. .8 .8 .8	100.0	0.0	0.0	85.7	38.2	54.9	100.0	0.0	0.0	90.0	67.36	46.88
1. .7 .7 .7	100.0	0.0	0.0	84.3	37.6	55.7	100.0	0.0	0.0	90.0	67.26	46.76
1. .6 .6 .6	100.0	0.0	0.0	84.3	36.3	56.6	100.0	0.0	0.0	92.0	67.26	46.91
1. .5 .5 .5	100.0	0.0	13.6	85.7	35.0	57.4	100.0	0.0	0.0	92.0	67.85	48.38
1. .4 .4 .4	100.0	0.0	15.9	85.7	34.4	83.6	100.0	0.0	0.0	92.0	71.02	51.16
1. .3 .3 .3	100.0	28.6	18.2	81.4	40.1	86.9	100.0	0.0	0.0	90.0	73.59	54.52
1. .2 .2 .2	100.0	50.0	18.2	72.9	43.9	87.7	98.0	0.0	0.0	74.0	73.79	54.47
1. .1 .1 .1	100.0	58.9	18.2	67.1	35.0	86.1	92.5	0.0	0.0	66.0	71.12	52.39
1. .0 .0 .0	100.0	62.5	18.2	55.7	26.1	85.2	91.2	0.0	0.0	52.0	68.15	49.09
1. 1. .9 1.	100.0	0.0	0.0	82.9	38.2	54.1	100.0	0.0	0.0	90.0	67.06	46.52
1. 1. .8 1.	100.0	0.0	0.0	84.3	38.2	54.1	100.0	0.0	0.0	90.0	67.16	46.66
1. 1. .7 1.	100.0	0.0	0.0	84.3	38.2	54.1	100.0	0.0	0.0	90.0	67.16	46.66
1. 1. .6 1.	100.0	0.0	0.0	84.3	38.2	53.3	100.0	0.0	0.0	90.0	67.06	46.58
1. 1. .5 1.	100.0	0.0	0.0	85.7	38.2	53.3	100.0	0.0	0.0	90.0	67.16	46.72
1. 1. .4 1.	100.0	0.0	0.0	85.7	38.9	53.3	100.0	0.0	0.0	90.0	67.26	46.78
1. 1. .3 1.	100.0	0.0	0.0	85.7	37.6	51.6	100.0	0.0	0.0	92.0	66.96	46.69
1. 1. .2 1.	100.0	0.0	0.0	85.7	36.9	51.6	100.0	0.0	0.0	92.0	66.86	46.63
1. 1. .1 1.	100.0	0.0	0.0	85.7	36.3	51.6	100.0	0.0	0.0	92.0	66.77	46.57
1. 1. .0 1.	100.0	0.0	0.0	85.7	36.9	52.5	100.0	0.0	4.0	92.0	67.06	47.11
1. 1. .8 .9	100.0	0.0	0.0	85.7	38.9	54.1	100.0	0.0	0.0	90.0	67.36	46.87
1. .9 .8 .9	100.0	0.0	0.0	84.3	38.9	54.1	100.0	0.0	0.0	90.0	67.26	46.72
1. .9 .7 .8	100.0	0.0	0.0	85.7	38.9	54.9	100.0	0.0	0.0	90.0	67.46	46.95
1. .9 .6 .8	100.0	0.0	0.0	85.7	38.9	54.9	100.0	0.0	0.0	92.0	67.56	47.15
1. .9 .6 .7	100.0	0.0	0.0	85.7	38.2	55.7	100.0	0.0	0.0	92.0	67.56	47.17
1. .9 .5 .7	100.0	0.0	0.0	85.7	36.9	55.7	100.0	0.0	0.0	92.0	67.36	47.04
1. .9 .5 .6	100.0	0.0	0.0	82.9	34.4	55.7	100.0	0.0	0.0	92.0	66.77	46.50
1. .8 .5 .6	100.0	0.0	0.0	85.7	35.7	56.6	100.0	0.0	0.0	92.0	67.26	46.99
1. .8 .4 .6	100.0	0.0	0.0	85.7	35.0	55.7	100.0	0.0	0.0	92.0	67.06	46.85
1. .8 .4 .5	100.0	0.0	0.0	81.4	31.8	55.7	100.0	0.0	0.0	92.0	66.27	46.10
1. .7 .4 .5	100.0	0.0	9.1	82.9	33.1	55.7	100.0	0.0	0.0	92.0	66.96	47.28
# of pixels	302	56	44	70	157	122	147	38	25	50	1011	1011

The columns labeled m e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

c) Topographic Data Modeled by Parzen Density Estimation

The topographic data sources were then modeled by Parzen density estimation using a Gaussian kernel function. Smoothing parameters (σ) were selected to give the highest source-specific overall accuracies. The smoothing parameters chosen were: elevation data (0.25), slope data (0.50) and aspect data (0.75). The results using the SMC are shown in Tables 4.25 and 4.26. Compared to the source-specific histogram classifications (Tables 4.15 and 4.16), the Parzen density estimation did better in modeling the elevation data both for classification accuracy of training and test data. In fact it also gave higher classification accuracies for test data for all the topographic data channels when compared to the histogram approach. Parzen density estimation also gave higher accuracies for the elevation data when compared to maximum penalized likelihood approach (Tables 4.21 and 4.22). The Parzen density estimation and the maximum penalized likelihood method were similar for the slope data in terms of training but the Parzen density estimation gave higher accuracies for testing. The maximum penalized likelihood approach showed better performance in modeling the elevation data.

Again the rank of the sources was not changed by using different modeling methods. For the Parzen density estimation and the source-specific classification accuracies of training data, the sources were ranked as follows: 1. MSS, 2. elevation, 3. aspect and 4. slope. This was the same ranking produced by the equivocation measures in Table 4.12. Looking at the training results using the SMC in Table 4.25, it is seen that the overall accuracy increased to 79.76% for the combination. However, this result was lower than both the

Table 4.25

Statistical Multisource Classification of Colorado Data when Topographic Sources were Modeled by Parzen Density Estimation: Training Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	Single Sources											
MSS	99.3	64.3	20.9	68.6	16.6	85.2	89.8	5.3	28.0	67.3	69.05	54.33
Elevation	100.0	0.0	0.0	78.6	24.2	17.2	98.6	31.6	32.0	100.0	62.40	48.22
Slope	95.3	0.0	0.0	4.3	9.6	13.1	72.1	0.0	8.0	2.0	42.66	20.45
Aspect	98.3	0.0	4.7	52.9	33.1	16.4	50.0	7.9	8.0	40.8	50.00	31.11
m e s a	Multiple Sources											
1. 1. 1. 1.	99.7	96.4	20.9	98.6	57.3	47.5	100.0	31.6	64.0	100.0	79.76	71.60
1. .9 .9 .9	99.7	94.6	20.9	98.6	54.1	57.4	100.0	23.7	60.0	100.0	79.96	70.90
1. .8 .8 .8	99.7	91.1	23.3	98.6	49.0	72.1	100.0	21.1	60.0	100.0	80.75	71.48
1. .7 .7 .7	99.7	91.1	23.3	98.6	47.8	77.9	100.0	15.8	52.0	100.0	80.85	70.60
1. .6 .6 .6	99.7	91.1	23.3	95.7	45.2	83.6	100.0	5.3	48.0	100.0	80.46	69.18
1. .5 .5 .5	99.7	87.5	23.3	95.7	40.1	86.1	100.0	0.0	36.0	100.0	79.27	66.83
1. .4 .4 .4	100.0	83.9	23.3	94.3	40.8	86.9	100.0	0.0	32.0	100.0	79.17	66.11
1. .3 .3 .3	100.0	73.2	25.6	91.4	38.2	91.0	100.0	0.0	28.0	100.0	78.47	64.74
1. .2 .2 .2	100.0	73.2	25.6	88.6	36.9	91.0	100.0	0.0	8.0	100.0	77.58	62.33
1. .1 .1 .1	100.0	66.1	25.6	82.9	34.4	91.8	100.0	0.0	8.0	100.0	76.49	60.87
1. .0 .0 .0	100.0	57.1	16.3	65.7	19.8	90.2	89.8	0.0	0.0	67.3	68.65	50.62
1. 1. .9 1.	99.7	96.4	20.9	98.6	56.1	50.8	100.0	28.9	64.0	100.0	79.86	71.54
1. 1. .8 1.	99.7	94.6	20.9	98.6	56.1	53.3	100.0	28.9	64.0	100.0	80.06	71.61
1. 1. .7 1.	99.7	92.9	20.9	98.6	56.1	56.6	100.0	31.6	60.0	100.0	80.36	71.62
1. 1. .6 1.	99.7	91.1	20.9	98.6	54.8	58.2	100.0	31.6	60.0	100.0	80.26	71.48
1. 1. .5 1.	99.7	91.1	20.9	98.6	52.9	66.4	100.0	31.6	60.0	100.0	80.95	72.11
1. 1. .4 1.	99.7	91.1	20.9	98.6	51.0	70.5	100.0	31.6	60.0	100.0	81.15	72.33
1. 1. .3 1.	99.7	91.1	20.9	97.1	49.7	72.1	100.0	31.6	60.0	100.0	81.05	72 22
1. 1. .2 1.	99.7	91.1	20.9	97.1	51.0	74.6	100.0	31.6	60.0	100.0	81.55	72.59
1. 1. .1 1.	99.7	89.3	20.9	97.1	52.9	74.6	100.0	28.9	60.0	100.0	81.65	72.34
1. 1. .0 1.	99.7	87.5	20.9	97.1	49.7	76.2	100.0	28.9	60.0	100.0	81.25	72.01
1. 1. .8 .9	99.7	94.6	32.3	98.6	54.8	54.4	100.0	31.6	64.0	100.0	80.56	72.39
1. .9 .8 .9	99.7	92.9	20.9	98.6	53.5	58.2	100.0	23.7	60.0	100.0	79.86	70.74
1. .9 .7 .8	99.7	91.1	23.3	98.6	49.0	72.1	100.0	23.7	60.0	100.0	80.85	71.74
1. .9 .6 .8	99.7	91.1	23.3	98.6	48.4	73.0	100.0	23.7	60.0	100.0	80.85	71.76
1. .9 .6 .7	99.7	91.1	23.3	97.1	48.4	77.0	100.0	23.7	60.0	100.0	81.25	72.03
1. .9 .5 .7	99.7	91.1	23.3	97.1	47.1	77.9	100.0	23.7	60.0	100.0	81.15	71.98
1. .9 .5 .6	99.7	91.1	23.3	97.1	45.9	82.0	100.0	15.8	64.0	100.0	81.25	71.88
1. .8 .5 .6	99.7	91.1	23.3	97.1	45.2	82.8	100.0	13.2	52.0	100.0	80.85	70.43
1 .8 .4 .6	99.7	89.3	23.3	95.7	45.2	83.6	100.0	7.9	52.0	100.0	80.56	69 66
1. .8 .4 .5	99.7	87.5	23.3	95.7	40.8	85.2	100.0	7.9	52.0	100.0	79.96	69 20
1. .7 .4 .5	99.7	87.5	23.3	95.7	41.4	86.1	100.0	2.6	48.0	100.0	79.86	68.42
# of pixels	301	56	43	70	157	122	147	38	25	49	1008	1008

The columns labeled m e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

CPU time for training and classification: 101 sec.

Table 4.26

Statistical Multisource Classification of Colorado Data when Topographic Sources were Modeled by Parzen Density Estimation: Test Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	Single Sources											
MSS	100.0	53.6	20.5	54.3	13.4	79.5	89.1	5.3	4.0	54.0	65.08	47.36
Elevation	100.0	0.0	0.0	75.7	22.9	14.8	98.0	26.3	28.0	100.0	61.33	46.57
Slope	95.4	0.0	0.0	5.7	7.6	13.9	68.0	0.0	8.0	0.0	41.84	19.87
Aspect	98.0	0.0	2.3	41.4	31.8	13.9	49.0	2.6	0.0	34.0	47.77	27.31
m e s a	Multiple Sources											
1. 1. 1. 1.	99.3	100.0	18.2	90.0	51.6	53.3	99.3	28.9	52.0	100.0	78.44	69.27
1. .9 .9 .9	99.3	98.2	18.2	92.9	47.8	65.6	99.3	15.8	40.0	100.0	78.64	67.70
1. .8 .8 .8	99.3	98.2	18.2	92.9	40.8	73.0	99.3	10.5	36.0	100.0	78.14	66.82
1. .7 .7 .7	100.0	98.2	18.2	91.4	41.4	80.3	99.3	10.5	24.0	100.0	78.93	66.34
1. .6 .6 .6	100.0	96.4	18.2	91.4	40.8	87.7	99.3	10.5	24.0	100.0	79.62	66.84
1. .5 .5 .5	100.0	94.6	18.2	91.4	38.2	90.2	99.3	5.3	16.0	100.0	79.03	65.32
1. .4 .4 .4	100.0	89.3	20.5	88.6	36.3	90.2	99.3	2.6	8.0	100.0	78.04	63.47
1. .3 .3 .3	100.0	80.4	22.7	84.3	34.4	92.6	99.3	0.0	0.0	100.0	77.05	61.37
1. .2 .2 .2	100.0	75.0	29.5	80.0	35.0	92.6	99.3	0.0	0.0	98.0	76.76	60.95
1. .1 .1 .1	100.0	69.6	27.3	72.9	33.1	91.0	99.3	0.0	0.0	98.0	75.37	59.12
1. .0 .0 .0	100.0	62.5	18.2	55.7	26.1	85.2	91.2	0.0	0.0	52.0	68.15	49.09
1. 1. .9 1.	99.3	98.2	18.2	88.6	49.7	55.7	99.3	23.7	44.0	100.0	77.84	67.67
1. 1. .8 1.	99.3	98.2	18.2	88.6	48.4	61.5	99.3	23.7	44.0	100.0	78.34	68.12
1. 1. .7 1.	99.3	98.2	18.2	88.6	47.1	66.4	99.3	23.7	44.0	100.0	78.73	68.48
1. 1. .6 1.	99.7	98.2	18.2	90.0	47.1	66.4	99.3	23.7	44.0	100.0	78.93	68.66
1. 1. .5 1.	99.7	98.2	18.2	90.0	44.6	70.5	99.3	23.7	44.0	100.0	79.03	68.81
1. 1. .4 1.	100.0	98.2	18.2	90.0	44.6	69.7	99.3	23.7	40.0	100.0	78.93	68.37
1. 1. .3 1.	100.0	96.4	18.2	90.0	43.3	71.3	99.3	21.1	40.0	100.0	78.73	67.96
1. 1. .2 1.	100.0	96.4	18.2	90.0	44.6	71.3	99.3	18.4	40.0	100.0	78.83	67.82
1. 1. .1 1.	100.0	94.6	20.5	90.0	45.2	73.0	99.3	18.4	36.0	100.0	79.03	67.70
1. 1. .0 1.	100.0	94.6	20.5	87.1	45.9	74.6	99.3	15.8	32.0	100.0	78.93	66.98
1. 1. .8 .9	99.3	98.2	18.2	91.4	46.5	66.4	99.3	21.1	44.0	100.0	78.73	68.44
1. .9 .8 .9	99.3	98.2	18.2	92.9	46.5	66.4	99.3	13.2	36.0	100.0	78.34	67.00
1. .9 .7 .8	100.0	98.2	18.2	92.9	42.0	71.3	99.3	10.5	40.0	100.0	78.44	67.24
1. .9 .6 .8	100.0	98.2	18.2	91.4	41.4	74.6	99.3	13.2	40.0	100.0	78.73	67.63
1. .9 .6 .7	100.0	98.2	18.2	91.4	42.0	77.0	99.3	10.5	36.0	100.0	78.93	67.28
1. .9 .5 .7	100.0	96.4	18.2	91.4	42.7	77.9	99.3	10.5	36.0	100.0	79.03	67.24
1. .9 .5 .6	100.0	96.4	18.2	90.0	40.0	82.0	99.3	13.2	40.0	100.0	79.23	67.92
1. .8 .5 .6	100.0	94.6	18.2	91.4	40.1	83.6	99.3	10.5	28.0	100.0	79.03	66.58
1. .8 .4 .6	100.0	94.6	20.5	90.0	38.9	83.6	99.3	10.5	24.0	100.0	78.73	66.14
1. .8 .4 .5	100.0	94.6	20.5	90.0	39.5	86.9	99.3	10.5	28.0	100.0	79.33	66.93
1. .7 .4 .5	100.0	92.9	20.5	90.0	39.5	88.5	99.3	10.5	24.0	100.0	79.33	66.52
# of pixels	302	56	44	70	157	122	147	38	25	50	1011	1011

The columns labeled m e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

histogram combination and the maximum penalized likelihood combination. By weighting the sources differently, the overall accuracy increased to 81.65% and the average accuracy became 72.34% (weighting was MSS 1.0, elevation 1.0, slope 0.1 and aspect 1.0). These results were again lower than achieved with the histogram method and the maximum penalized likelihood method.

Looking at the SMC testing result with the Parzen density estimation (Table 4.26), it is seen that the best combination result was achieved with full weights. The Parzen density estimation combination gave an overall accuracy of 78.44% and an average accuracy of 69.27%, an increase in overall accuracy of 1.19% compared to the histogram counterpart and 0.7% over the maximum penalized likelihood combination with full weights. The increase in average accuracy was more dramatic: 2.38% above the histogram combination with equal weights and 3.18% above the maximum penalized likelihood counterpart. When the weights were changed to (1.0,0.8,0.4,0.5) the overall accuracy increased to 79.33%, only 0.89% higher than the overall accuracy achieved with equal weights. The average accuracy also decreased to 66.93%, or 2.34% lower than the average accuracy with equal weights. The maximum penalized likelihood method with the weights (1.0,0.8,0.4,0.5) gave 80.02% overall accuracy and a 67.29% average. So the maximum penalized likelihood estimate combination could be improved more in terms of overall accuracy in the experiments although the Parzen density estimation combination gave higher accuracy with equal weights. Apart from this the results using these two density estimation methods for test data were similar and better in terms of accuracies of test data than the histogram results.

The results using the LOP with the Parzen density estimation are shown in Tables 4.27 and 4.28. The training results in Table 4.27 were very similar to the results with the maximum penalized likelihood method in Table 4.23. The highest overall accuracy was 75.89% (with the weights 1.0,0.2,0.2,0.2) which was 0.27% lower than the result with the maximum penalized likelihood approach and the same weights. However, the average classification results were slightly higher in Table 4.27 than in Table 4.23. Looking at the test result with the Parzen density estimate in Table 4.28, it is seen that the overall test accuracy with equal weights was 66.67%, which was 0.39% lower than the counterpart with the maximum penalized likelihood method in Table 4.24. The average accuracy of 46.58% was slightly higher than the one in Table 4.24 (46.52%). With the weighting (1.0,0.2,0.2,0.2) the overall accuracy for the test data with the LOP and Parzen density estimation increased to 74.09%, higher than the one achieved by the maximum penalized likelihood method with the same weights (73.79%) and also better than the histogram counterpart (73.79%). The average accuracy with the Parzen density estimation (54.93%) was also slightly higher than with the other density estimation methods (Tables 4.20 and 4.24).

d) General Comments on the Statistical Methods

Looking at the results for this second experiment using statistical methods, it is evident that the SMC did a much better job in terms of overall and average accuracy than the linear opinion pool. The linear opinion pool had the weakness that it was very poor in classifying the classes with the lowest prior probabilities. The SMC performed much better. However, the

Table 4.27

Linear Opinion Pool Applied to Colorado Data Set when Topographic Sources were Modeled by Parzen Density Estimation: Training Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	Single Sources											
MSS	99.3	64.3	20.9	68.6	16.6	85.2	89.8	5.3	28.0	67.3	69.05	54.33
Elevation	100.0	0.0	0.0	78.6	24.2	17.2	98.6	31.6	32.0	100.0	62.40	48.22
Slope	95.3	0.0	0.0	4.3	9.6	13.1	72.1	0.0	8.0	2.0	42.66	20.45
Aspect	98.3	0.0	4.7	52.9	33.1	16.4	50.0	7.9	8.0	40.8	50.00	31.11
	Multiple Sources											
m e s a												
1. 1. 1. 1.	100.0	0.0	0.0	91.4	38.9	50.0	100.0	0.0	12.0	100.0	68.06	49.23
1. .9 .9 .9	100.0	0.0	0.0	91.4	38.9	50.0	100.0	0.0	12.0	100.0	68.06	49.23
1. .8 .8 .8	100.0	0.0	0.0	91.4	38.9	50.0	100.0	0.0	12.0	100.0	68.06	49.23
1. .7 .7 .7	100.0	0.0	0.0	91.4	39.5	50.8	100.0	0.0	12.0	100.0	68.25	49.37
1. .6 .6 .6	100.0	0.0	0.0	92.9	38.9	54.1	100.0	0.0	4.0	100.0	68.45	48.98
1. .5 .5 .5	100.0	0.0	16.3	92.9	34.4	54.1	100.0	0.0	0.0	100.0	68.35	49.76
1. .4 .4 .4	100.0	0.0	16.3	91.4	35.7	86.1	100.0	0.0	0.0	100.0	72.32	52.94
1. .3 .3 .3	100.0	25.0	16.3	91.4	36.3	90.2	99.3	0.0	0.0	100.0	74.21	55.84
1. .2 .2 .2	100.0	51.8	16.3	84.3	42.7	91.0	99.3	0.0	0.0	91.8	75.89	57.72
1. .1 .1 .1	100.0	55.4	16.3	82.9	32.5	89.3	93.2	0.0	0.0	73.5	72.42	54.30
1. .0 .0 .0	100.0	57.1	16.3	65.7	19.7	90.2	89.8	0.0	0.0	67.3	68.65	50.62
1. 1. .9 1.	100.0	0.0	0.0	91.4	38.9	50.0	100.0	0.0	12.0	100.0	68.06	49.23
1. 1. .8 1.	100.0	0.0	0.0	92.9	38.9	50.0	100.0	0.0	12.0	100.0	68.15	49.37
1. 1. .7 1.	100.0	0.0	0.0	92.9	38.2	50.0	100.0	0.0	16.0	100.0	68.15	49.71
1. 1. .6 1.	100.0	0.0	0.0	92.9	37.6	50.0	100.0	0.0	16.0	100.0	68.06	49.64
1. 1. .5 1.	100.0	0.0	0.0	92.9	37.6	50.0	100.0	0.0	16.0	100.0	68.06	49.64
1. 1. .4 1.	100.0	0.0	0.0	92.9	37.6	50.0	100.0	0.0	16.0	100.0	68.06	49.64
1. 1. .3 1.	100.0	0.0	0.0	92.9	37.6	50.0	100.0	0.0	16.0	100.0	68.06	49.64
1. 1. .2 1.	100.0	0.0	0.0	92.9	37.6	50.8	100.0	0.0	20.0	100.0	68.25	50.13
1. 1. .1 1.	100.0	0.0	0.0	92.9	37.6	50.8	100.0	0.0	20.0	100.0	68.25	50.13
1. 1. .0 1.	100.0	0.0	0.0	92.9	36.9	50.8	100.0	0.0	20.0	100.0	68.15	50.06
1. 1. .8 .9	100.0	0.0	0.0	92.9	38.2	50.0	100.0	0.0	12.0	100.0	68.06	49.31
1. .9 .8 .9	100.0	0.0	0.0	91.4	38.9	50.0	100.0	0.0	12.0	100.0	68.06	49.23
1. .9 .7 .8	100.0	0.0	0.0	92.9	38.9	50.8	100.0	0.0	12.0	100.0	68.25	49.45
1. .9 .6 .8	100.0	0.0	0.0	92.9	37.6	50.8	100.0	0.0	12.0	100.0	68.06	49.33
1. .9 .6 .7	100.0	0.0	0.0	92.9	37.6	50.8	100.0	0.0	16.0	100.0	68.15	49.73
1. .9 .5 .7	100.0	0.0	0.0	92.9	37.6	50.8	100.0	0.0	16.0	100.0	68.15	49.73
1. .9 .5 .6	100.0	0.0	0.0	92.9	36.9	51.6	100.0	0.0	12.0	100.0	68.06	49.34
1. .8 .5 .6	100.0	0.0	0.0	92.9	36.9	52.5	100.0	0.0	12.0	100.0	68.15	49.43
1. .8 .4 .6	100.0	0.0	0.0	92.9	36.9	52.5	100.0	0.0	16.0	100.0	68.25	49.83
1. .8 .4 .5	100.0	0.0	0.0	92.9	34.4	52.5	100.0	0.0	8.0	100.0	67.66	48.77
1. .7 .4 .5	100.0	0.0	9.3	92.9	33.8	53.3	100.0	0.0	8.0	100.0	68.06	49.72
# of pixels	301	56	43	70	157	122	147	38	25	49	1008	1008

The columns labeled m e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

CPU time for training and classification: 99 sec.

Table 4.28

Linear Opinion Pool Applied to Colorado Data Set when Topographic Sources were Modeled by Parzen Density Estimation: Test Samples.

	Percent Agreement with Reference for Class											
	1	2	3	4	5	6	7	8	9	10	OA	AVE
	Single Sources											
MSS	100.0	53.6	20.5	54.3	13.4	79.5	89.1	5.3	4.0	54.0	65.08	47.36
Elevation	100.0	0.0	0.0	75.7	22.9	14.8	98.0	26.3	28.0	100.0	61.33	46.57
Slope	95.4	0.0	0.0	5.7	7.6	13.9	68.0	0.0	8.0	0.0	41.84	19.87
Aspect	98.0	0.0	2.3	41.4	31.8	13.9	49.0	2.6	0.0	34.0	47.77	27.31
m e s a	Multiple Sources											
1. 1. 1. 1.	100.0	0.0	0.0	87.1	35.0	51.6	100.0	0.0	0.0	92.0	66.67	46.58
1. .9 .9 .9	100.0	0.0	0.0	85.7	33.8	52.5	100.0	0.0	0.0	94.0	66.57	46.59
1. .8 .8 .8	100.0	0.0	0.0	85.7	33.8	55.7	100.0	0.0	0.0	94.0	66.96	46.92
1. .7 .7 .7	100.0	0.0	0.0	85.7	33.8	56.6	100.0	0.0	0.0	96.0	67.16	47.20
1. .6 .6 .6	100.0	0.0	0.0	85.7	33.1	57.4	100.0	0.0	0.0	98.0	67.26	47.42
1. .5 .5 .5	100.0	0.0	13.6	84.3	32.5	59.0	100.0	0.0	0.0	100.0	67.95	48.94
1. .4 .4 .4	100.0	0.0	15.9	84.3	31.9	86.1	100.0	0.0	0.0	100.0	71.22	51.81
1. .3 .3 .3	100.0	30.4	18.2	80.0	35.7	88.5	100.0	0.0	0.0	96.0	73.39	54.87
1. .2 .2 .2	100.0	50.0	18.2	72.9	43.9	87.7	98.6	0.0	0.0	78.0	74.09	54.93
1. .1 .1 .1	100.0	58.9	18.2	65.7	32.5	86.1	93.2	0.0	0.0	70.0	70.92	52.46
1. .0 .0 .0	100.0	62.5	18.2	55.7	26.1	85.2	91.2	0.0	0.0	52.0	68.15	49.09
1. 1. .9 1.	100.0	0.0	0.0	85.7	35.7	51.6	100.0	0.0	0.0	94.0	66.77	46.70
1. 1. .8 1.	100.0	0.0	0.0	85.7	35.7	52.5	100.0	0.0	0.0	94.0	66.86	46.78
1. 1. .7 1.	100.0	0.0	0.0	85.7	35.7	53.3	100.0	0.0	0.0	94.0	66.96	46.87
1. 1. .6 1.	100.0	0.0	0.0	85.7	35.7	53.3	100.0	0.0	0.0	94.0	66.96	46.87
1. 1. .5 1.	100.0	0.0	0.0	87.1	35.7	51.6	100.0	0.0	0.0	96.0	66.96	47.04
1. 1. .4 1.	100.0	0.0	0.0	87.1	35.7	50.8	100.0	0.0	0.0	98.0	66.96	47.16
1. 1. .3 1.	100.0	0.0	0.0	87.1	36.3	50.0	100.0	0.0	4.0	100.0	67.16	47.74
1. 1. .2 1.	100.0	0.0	0.0	87.1	35.0	49.2	100.0	0.0	8.0	100.0	66.96	47.94
1. 1. .1 1.	100.0	0.0	0.0	87.1	34.4	49.2	100.0	0.0	8.0	100.0	66.86	47.87
1. 1. .0 1.	100.0	0.0	0.0	87.1	33.8	50.0	100.0	0.0	12.0	100.0	66.96	48.29
1. 1. .8 .9	100.0	0.0	0.0	87.1	35.0	53.3	100.0	0.0	0.0	94.0	66.96	46.95
1. .9 .8 .9	100.0	0.0	0.0	85.7	34.4	53.3	100.0	0.0	0.0	94.0	66.77	46.74
1. .9 .7 .8	100.0	0.0	0.0	85.7	34.4	55.7	100.0	0.0	0.0	96.0	67.16	47.18
1. .9 .6 .8	100.0	0.0	0.0	85.7	34.4	55.7	100.0	0.0	0.0	96.0	67.16	47.18
1. .9 .6 .7	100.0	0.0	0.0	84.3	33.8	55.7	100.0	0.0	4.0	96.0	67.06	47.38
1. .9 .5 .7	100.0	0.0	0.0	84.3	33.1	55.7	100.0	0.0	4.0	98.0	67.06	47.51
1. .9 .5 .6	100.0	0.0	0.0	84.3	32.5	55.7	100.0	0.0	8.0	98.0	67.06	47.85
1. .8 .5 .6	100.0	0.0	0.0	82.9	32.5	56.6	100.0	0.0	4.0	98.0	66.96	47.39
1. .8 .4 .6	100.0	0.0	0.0	84.3	33.1	57.4	100.0	0.0	4.0	100.0	67.36	47.88
1. .8 .4 .5	100.0	0.0	0.0	84.3	31.2	58.2	100.0	0.0	4.0	100.0	67.16	47.77
1. .7 .4 .5	100.0	0.0	11.4	84.3	31.8	58.2	100.0	0.0	0.0	100.0	67.66	48.60
# of pixels	302	56	44	70	157	122	147	38	25	50	1011	1011

The columns labeled m e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

LOP was a little faster than the SMC. The maximum penalized likelihood method gave the highest overall accuracy of test data, but that method and the Parzen density estimation showed a very similar performance in terms of accuracy of test data. The histogram approach was best for training data and it is clear that it is very hard to improve on it there.

The CPU times for the different methods are shown in Table 4.30. The histogram estimation is clearly the fastest (1 sec); the Parzen density estimation (30 sec) and the maximum penalized likelihood method (31 sec) were very close in speed in this experiment. The training and test samples were very small in this experiment. In Section 4.3 it will be seen how well these methods perform in terms of speed with larger sample sizes.

4.2.5 Results of the Second Experiment: Neural Network Methods

The neural network methods were trained as in Section 4.2.2. There were 56 input neurons and 13 output neurons to account for the 13 data classes. The input data was Gray-coded and the convergence criterion for the training procedures was the same as in Section 4.2.2.

a) Experiments with the Conjugate Gradient Linear Classifier

The classification results for the CGLC network are shown in Tables 4.31 (training) and 4.32 (test). The training procedure did not converge but stopped after 344 iteration when the error function did not decrease further. The highest overall accuracy of training data was reached after 344 iterations (82.24%). However, the highest average accuracy of training data was achieved after 250 iterations (73.44%). The highest overall accuracy of test

Table 4.29

Source-Specific CPU Time (Training Plus Classification): Landsat MSS Data Source.

Sensor	MSS
# of channels	4
CPU time	4

Table 4.30

Source-Specific CPU Times (Training Plus Classification) for Topographic Data Sources with Respect to Different Modeling Methods.

Method	Histogram Estimation	Maximum Penalized Likelihood Method	Parzen Estimation
CPU time	1	31	30

Table 4.31

Conjugate Gradient Linear Classifier Applied to Colorado Data: Training Samples.

Number of	CPU	Percent Agreement with Reference for Class											
iterations	time	1	2	3	4	5	6	7	8	9	10	OA	AVE
50	110	100.0	92.9	37.2	87.1	52.2	73.8	98.6	21.1	20.0	87.8	79.66	66.07
100	209	100.0	94.6	39.5	85.7	56.7	70.5	100.0	28.9	52.0	89.8	81.45	71.77
150	295	100.0	85.7	58.1	85.7	59.2	72.1	100.0	28.9	56.0	87.8	82.34	73.35
200	375	100.0	85.7	53.5	84.3	58.6	74.6	100.0	23.7	56.0	91.8	82.24	72.82
250	483	100.0	85.7	55.8	82.9	59.2	74.6	100.0	26.3	56.0	93.9	82.54	73.44
300	569	100.0	85.7	55.8	82.9	61.1	69.7	100.0	26.3	56.0	91.8	82.14	72.93
343	644	100.0	85.7	55.8	82.9	61.1	69.6	100.0	26.3	56.0	93.9	82.24	73.13
# of pixels		301	56	43	70	157	122	147	38	25	49	1008	1008

Table 4.32

Conjugate Gradient Linear Classifier Applied to Colorado Data: Test Samples.

Number of	Percent Agreement with Reference for Class											
iterations	1	2	3	4	5	6	7	8	9	10	OA	AVE
50	100.0	87.5	31.8	80.0	49.7	67.2	97.3	18.4	20.0	80.0	76.76	63.19
100	100.0	96.4	38.6	71.4	54.8	74.6	98.6	18.4	48.0	80.0	79.53	68.08
150	100.0	85.7	50.0	74.3	55.4	73.8	98.0	21.1	56.0	76.0	79.62	69.03
200	100.0	85.7	45.5	75.7	54.8	74.6	98.0	18.4	60.0	78.0	79.62	69.07
250	100.0	85.7	47.7	72.9	54.8	74.6	98.0	18.4	60.0	78.0	79.53	69.01
300	100.0	85.7	47.7	71.4	55.4	73.0	98.0	18.4	60.0	78.0	79.33	68.76
343	100.0	85.7	47.7	72.9	55.4	73.0	98.0	18.4	60.0	78.0	79.43	68.91
# of pixels	302	56	44	70	157	122	147	38	25	50	1011	1011

data was reached after both 150 and 200 iterations (79.62%). The highest average accuracy of test data was observed after 200 iterations. After 343 iterations the overall accuracy of test data was 79.43% and the average accuracy was 68.91%.

b) Experiments with the Conjugate Gradient Backpropagation

The three layer CGBP was trained with 8, 16 and 32 hidden neurons. Using more than three layers did not improve the accuracy of the network. The classification results with 8 hidden neurons are shown in Tables 4.33 (training) and 4.34 (test). The training procedure stopped after 933 iterations and the highest overall accuracy was reached after 900 iterations (87.80%) together with the highest average accuracy (79.62%). Using the 8 hidden neurons improved the overall accuracy of training data by over 5% and the average accuracy by over 6% as compared to the CGLC. However, the CGBP training procedure was more time consuming than the CGLC as seen in Tables 4.31 and 4.33. Although the training results were better for the CGBP with 8 hidden neurons as compared to the CGLC, the test results were worse, both in terms of overall accuracies and average accuracies. The best accuracy for test results in Table 4.34 were achieved after 150 iterations (overall: 79.23%, average: 65.62%). The results after 933 iterations were lower (overall: 77.65%, average: 65.05%).

The CGBP results with 16 hidden neurons are shown in Tables 4.35 (training) and 4.36 (test). After 979 iterations the error function did not decrease and the highest values of overall accuracy (92.46%) and average accuracy (90.03%) were reached. Although these accuracies were significantly

Table 4.33

Conjugate Gradient Backpropagation with 8 Hidden Neurons Applied to Colorado Data: Training Samples.

Number of	CPU	Percent Agreement with Reference for Class											
iterations	time	1	2	3	4	5	6	7	8	9	10	OA	AVE
50	112	100.0	96.4	4.7	85.7	39.5	67.2	99.3	5.3	0.0	87.8	74.60	58.59
100	202	100.0	89.3	41.9	88.6	58.6	75.4	100.0	18.4	8.0	91.8	80.95	67.20
150	292	100.0	91.1	46.5	87.1	66.9	77.0	100.0	34.2	8.0	91.8	83.23	70.26
200	378	100.0	82.1	62.8	91.4	64.3	85.2	100.0	42.1	24.0	95.9	85.22	74.78
250	473	100.0	85.7	55.8	92.9	66.2	82.8	100.0	47.4	28.0	95.9	85.52	75.47
300	558	100.0	87.5	55.8	94.3	65.6	86.9	100.0	50.0	28.0	95.9	86.21	76.40
350	641	100.0	87.6	58.1	94.3	62.4	88.5	100.0	47.4	32.0	98.0	86.11	76.82
400	873	100.0	87.5	60.5	92.9	65.0	86.9	100.0	50.0	40.0	98.0	86.61	78.08
600	1102	100.0	96.4	46.5	94.3	68.8	90.2	100.0	52.6	40.0	98.0	87.70	78.68
900	1644	100.0	96.4	48.8	95.7	65.0	91.8	100.0	60.5	40.0	98.0	87.80	79.62
933	1719	100.0	96.4	48.8	95.7	64.3	91.8	100.0	60.5	40.0	98.0	87.70	79.55
# of pixels		301	56	43	70	157	122	147	38	25	49	1008	1008

Table 4.34

Conjugate Gradient Backpropagation with 8 Hidden Neurons Applied to Colorado Data: Test Samples.

Number of	Percent Agreement with Reference for Class											
iterations	1	2	3	4	5	6	7	8	9	10	OA	AVE
50	100.0	75.0	15.9	75.7	28.0	83.6	98.0	2.6	0.0	80.0	72.70	55.88
100	100.0	100.0	18.2	78.6	57.3	73.0	98.6	18.4	8.0	84.0	78.73	63.61
150	100.0	96.4	27.3	82.9	61.1	65.6	98.6	26.3	20.0	78.0	79.23	65.62
200	100.0	75.0	47.7	77.1	57.3	65.6	97.3	21.1	4.0	80.0	77.25	62.51
250	100.0	76.8	45.5	75.7	56.7	64.8	98.0	21.1	4.0	74.0	76.76	61.66
300	100.0	83.9	47.7	75.7	54.1	66.4	98.0	23.7	4.0	76.0	77.25	62.95
350	100.0	82.5	43.2	74.3	53.5	65.6	97.3	31.6	12.0	76.0	77.15	63.60
400	100.0	83.9	43.2	77.1	54.8	66.4	97.3	31.6	20.0	71.4	77.62	64.57
600	100.0	87.5	36.4	77.1	54.8	63.1	97.3	34.2	13.2	70.0	77.15	63.36
900	100.0	87.5	38.6	77.1	56.1	63.1	96.6	36.8	20.0	72.0	77.55	64.78
933	100.0	87.5	38.6	77.1	56.1	63.1	96.6	39.5	20.0	72.0	77.65	65.05
# of pixels	302	56	44	70	157	122	147	38	25	50	1011	1011

Table 4.35

Conjugate Gradient Backpropagation with 16 Hidden Neurons Applied to Colorado Data: Training Samples.

Number of	CPU	Percent Agreement with Reference for Class											
iterations	time	1	2	3	4	5	6	7	8	9	10	OA	AVE
50	180	100.0	69.6	18.6	84.3	28.7	82.8	98.0	5.3	4.0	85.7	73.61	57.70
100	353	100.0	94.6	18.6	87.1	60.5	71.3	100.0	15.8	24.0	95.9	80.46	66.78
150	524	100.0	92.9	32.6	92.9	65.6	68.9	100.0	34.2	52.0	93.9	83.13	73.30
200	685	100.0	94.6	34.9	92.9	65.6	80.3	100.0	52.6	68.0	95.9	85.91	78.48
250	847	100.0	92.9	39.5	98.6	67.5	82.0	100.0	68.4	76.0	98.0	87.80	82.29
300	1015	100.0	91.6	48.8	98.6	71.3	85.2	100.0	76.3	84.0	98.0	89.58	85.38
350	1161	100.0	92.9	46.5	100.0	75.2	82.8	100.0	76.3	92.0	98.0	90.18	86.37
400	1341	100.0	91.1	46.5	100.0	76.4	86.1	100.0	78.9	100.0	100.0	91.07	87.90
600	2006	100.0	92.9	51.2	100.0	77.1	86.1	100.0	84.2	100.0	100.0	91.67	89.15
900	3005	100.0	91.1	53.4	100.0	77.7	87.7	100.0	86.8	100.0	100.0	92.06	89.67
979	3257	100.0	91.1	55.8	100.0	82.2	84.4	100.0	86.8	100.0	100.0	92.46	90.03
# of pixels		301	56	43	70	157	122	147	38	25	49	1008	1008

Table 4.36

Conjugate Gradient Backpropagation with 16 Hidden Neurons Applied to Colorado Data: Test Samples.

Number of	Percent Agreement with Reference for Class											
iterations	1	2	3	4	5	6	7	8	9	10	OA	AVE
50	100.0	75.0	15.9	75.7	28.0	83.6	98.0	2.6	0.0	80.0	72.70	55.88
100	100.0	100.0	18.2	78.6	57.3	73.0	98.6	18.4	8.0	84.0	78.73	63.61
150	100.0	96.4	27.3	82.9	61.1	65.6	98.6	26.3	20.0	78.0	79.23	65.62
200	100.0	92.9	31.8	77.1	53.5	70.5	98.6	31.6	28.0	74.0	78.44	65.80
250	100.0	83.9	29.5	68.6	53.5	68.0	98.6	36.8	32.0	74.0	77.25	64.49
300	100.0	83.9	45.5	64.3	56.1	63.1	98.0	31.6	36.0	72.0	77.15	65.05
350	100.0	83.9	40.9	61.4	63.1	60.7	98.0	28.9	36.0	68.0	77.25	64.09
400	100.0	82.1	47.7	65.7	59.2	61.5	96.6	26.3	36.0	74.0	77.35	64.91
600	100.0	80.4	47.7	60.0	56.1	63.1	96.6	28.9	36.0	70.0	76.36	63.88
900	100.0	80.4	47.7	58.6	55.4	62.3	97.3	28.9	36.0	58.0	75.77	62.46
979	100.0	78.6	47.7	58.6	59.2	58.2	97.3	28.9	36.0	58.0	75.57	62.25
# of pixels	302	56	44	70	157	122	147	38	25	50	1011	1011

improved from the results with 8 hidden neurons, the test results (Table 4.36) were no better than the ones with 8 hidden neurons. Also, after 350 iterations the test results with 16 hidden neurons were worse than those with 8 hidden neurons. Similar results were observed with the CGBP when 32 hidden neurons were used (Tables 4.37 (training) and 4.38 (test)). The highest overall (93.45%) and average (91.74%) accuracies were reached after 807 iterations with 32 hidden neurons. The overall and average accuracies of test data (Table 4.38) were still lower when 16, 8 or no hidden neurons (CGLC) were used. As pointed out above, using hidden neurons makes the training procedure more time consuming (see Tables 4.31 (no hidden neurons), 4.33, 4.35, 4.37). The classification time for training and test data was also longer as seen below:

1) No hidden neurons: 11 sec.
2) 8 hidden neurons: 17 sec.
3) 16 hidden neurons: 20 sec.
4) 32 hidden neurons: 26 sec.

4.2.6 Summary

The best results from the second experiment on the Colorado data are shown in Figure 4.5. The results of this experiment showed that the neural network methods can do as well as the statistical methods when representative training samples are used. The neural network methods always outperformed the statistical methods in terms of classification of training data, but in terms of overall classification accuracy of test data, the SMC method was slightly better than the neural networks. This was in contrast to the results achieved

Table 4.37

Conjugate Gradient Backpropagation with 32 Hidden Neurons Applied to Colorado Data: Training Samples.

Number of iterations	CPU time	Percent Agreement with Reference for Class 1	2	3	4	5	6	7	8	9	10	OA	AVE
50	349	100.0	96.4	32.6	82.9	51.6	73.8	100.0	10.5	20.0	87.8	79.07	65.56
100	666	100.0	83.9	44.2	90.0	65.0	72.1	100.0	34.2	44.0	93.9	83.04	72.73
150	967	100.0	94.6	46.5	97.1	65.6	84.4	100.0	47.4	68.0	100.0	87.20	80.36
200	1287	100.0	92.9	62.8	100.0	73.2	85.2	100.0	73.7	100.0	100.0	91.07	88.78
250	1609	100.0	91.1	53.5	100.0	77.1	91.0	100.0	81.6	100.0	100.0	92.16	89.43
300	1967	100.0	94.6	53.5	100.0	79.0	88.5	100.0	84.2	100.0	100.0	92.46	89.98
350	2260	100.0	87.5	62.8	100.0	84.1	86.1	100.0	89.5	100.0	100.0	93.15	91.00
400	2558	100.0	87.5	62.8	100.0	80.9	90.2	100.0	89.5	100.0	100.0	93.15	91.09
600	3812	100.0	89.3	62.8	100.0	83.4	87.7	100.0	89.5	100.0	100.0	93.35	91.27
807	5045	100.0	92.9	65.1	100.0	82.2	87.7	100.0	89.5	100.0	100.0	93.45	91.74
# of pixels		301	56	43	70	157	122	147	38	25	49	1008	1008

Table 4.38

Conjugate Gradient Backpropagation with 32 Hidden Neurons Applied to Colorado Data: Test Samples.

Number of iterations	Percent Agreement with Reference for Class 1	2	3	4	5	6	7	8	9	10	OA	AVE
50	100.0	94.6	31.8	80.0	51.0	74.6	98.6	13.2	8.0	82.0	78.04	63.38
100	100.0	82.1	45.5	75.7	55.4	65.6	98.6	21.1	28.0	76.0	77.74	64.80
150	100.0	92.9	36.4	67.1	57.3	70.5	98.6	28.9	36.0	80.0	78.93	66.77
200	100.0	82.1	52.3	60.0	85.4	65.6	98.6	31.6	48.0	74.0	77.74	69.76
250	100.0	82.1	43.2	60.0	56.7	62.3	98.6	31.6	56.0	78.0	77.55	66.85
300	100.0	83.9	45.5	61.4	61.1	62.3	98.6	31.6	48.0	80.0	78.44	67.24
350	99.7	76.8	40.9	57.1	57.3	58.2	98.6	34.2	52.0	78.0	76.46	65.28
400	99.7	76.8	40.9	58.6	57.3	60.7	98.6	34.2	48.0	74.0	76.56	64.88
600	99.7	75.0	40.9	54.3	55.4	59.0	98.0	28.9	56.0	70.0	75.37	63.72
807	99.7	80.4	40.9	55.7	54.8	57.4	98.0	26.3	56.0	68.0	75.27	63.72
# of pixels	302	56	44	70	157	122	147	38	25	50	1011	1011

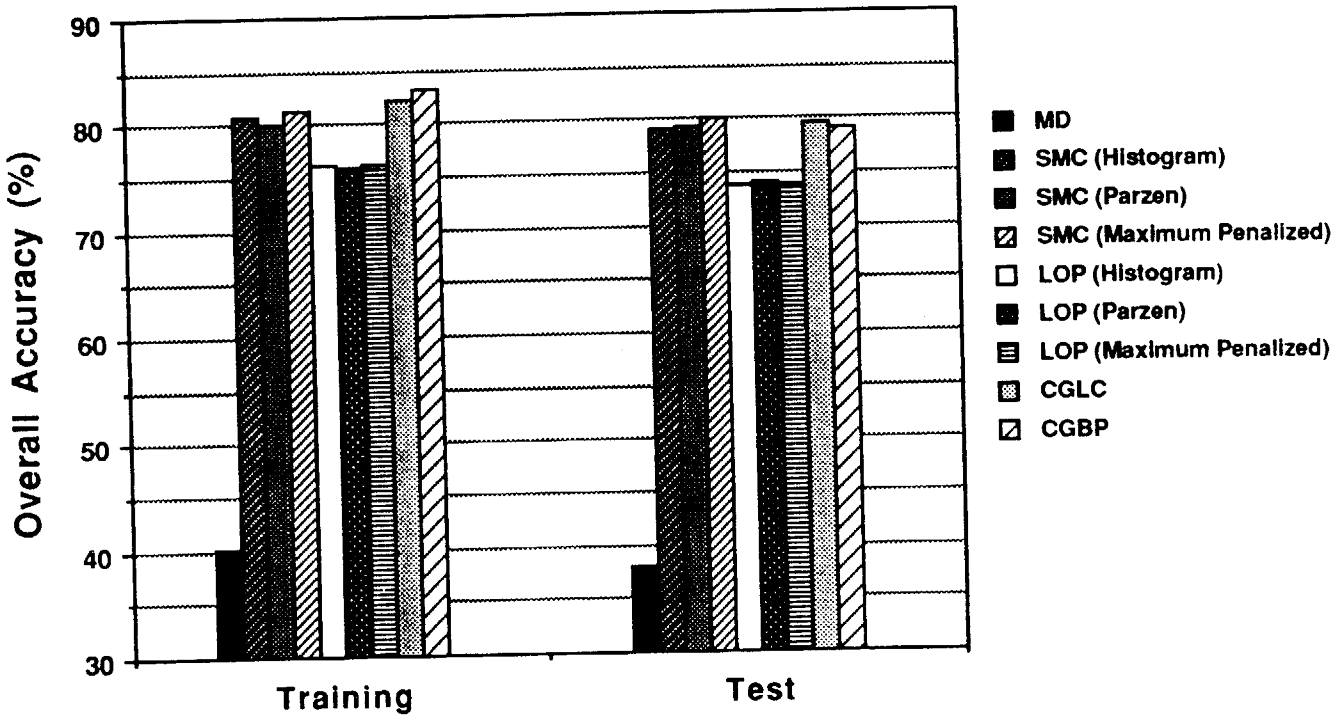


Figure 4.5 Summary of Best Classification Results for Second Experiment on Colorado Data.

in the first experiment on the Colorado data where the training data were not as representative. In the first experiment the SMC method outperformed the neural networks by more than 4% in overall test accuracy.

In the second experiment the SMC showed very good performance with equal weights but could be improved by more than 2% with different weight selections. The SMC outperformed the LOP by much in classification of these data. The highest overall classification accuracy for test data was reached by the SMC (80.02%) when the topographic data sources were modeled by the maximum penalized likelihood method. The highest overall accuracy for test data with the neural network methods was reached with the CGLC (79.62%). Adding hidden neurons did not improve the performance of the neural networks in terms of classification accuracy for test data, although it did improve the accuracy for training data. Using hidden neurons also slowed the training procedure. In general the neural networks took longer to train than the statistical methods. They were also more time consuming in classification of training and test data. The SMC and LOP needed only 7 and 5 sec of CPU time respectively.

In both experiments on the Colorado data the neural network methods were better in terms of accuracy than the statistical methods in classification of training data. The class prior probabilities in the statistical methods have an overwhelming effect on those methods which favors certain classes. Although a number of training samples for a class provides the neural network with "prior" information, the effect is different than multiplying the priors as in the statistical case. One of the major problems with the neural network methods is determining how to prevent them from "overtraining." In order to achieve the highest accuracy for test data, the networks often need fewer iterations than the training procedures go through.

4.3 Experiments with Anderson River Data

The Anderson River data set is a multisource data set made available by the Canada Centre for Remote Sensing (CCRS) [83]. The imagery involves a 2.8 km by 2.8 km forestry site in the Anderson River area of British Columbia, Canada, characterized by rugged topography, with terrain elevations ranging from 330 to 1100 m above sea level. The forest cover is primarily coniferous, with Douglas fir predominating up to approximately 1050 m elevation, and cedar, hemlock and spruce types predominating at higher elevations. The Anderson River data set consists of six data sources:

1) Airborne Multispectral Scanner (ABMSS) with 11 data channels (10 channels from 380 to 1100 nm and 1 channel from 8 to 14 μm).

2) Steep Mode Synthetic Aperture Radar (SAR) with 4 data channels (X–HH,X–HV,L–HH,L–HV)[3].

3) Shallow Mode SAR with 4 data channels (X-HH, X-HV, L-HH, L-HV).

4) Elevation data, 1 data channel, with elevation in meters = 61.996 + 7.2266 * pixel value.

5) Slope data, 1 data channel, with slope in degrees = pixel value.

6) Aspect data, 1 data channel, where aspect in degrees = 2 * pixel value.

[3] X- and L-band synthetic aperature radar imagery (horizontal polarization transmit (HH) and horizontal/vertical polarization receive (HV)).

The ABMSS and SAR data were detected during the week of July 25 to 31, 1978. Each channel comprises an image of 256 lines and 256 columns. All of the images are co-registered with pixel resolution of 12.5m.

There are 19 information classes in the ground reference map provided by CCRS. In the experiments reported here only the 6 predominant classes were used, as listed in Table 4.39. Three of these classes, Douglas fir (21-30m), Douglas fir + lodgepole pine, and forest clearings (classes 2,4 and 6), each covered two spatially distinct fields. Therefore, these classes were trained as two different data classes, and the total number of data classes in the experiments became 9. Training samples were selected on a uniform grid as 10% of the total the sample size of a class.

The separability of the information classes for each of the data sources was examined. The ABMSS and SAR data sources were modeled as Gaussian and their separability was estimated by computing the JM distances between the information classes. On the other hand, the topographic data sources were non-Gaussian with one feature each. A convenient way of examining the discriminability of the classes in the topographic sources is to look at the class histograms for the information classes.

In Tables 4.40 to 4.42 the JM distances (maximum of 1.41421) between the information classes are shown for the ABMSS (Table 4.40) and SAR (Tables 4.41 and 4.42) data sources. The ABMSS source had an average separability of 1.199, the SAR sh (Shallow) source an average of 0.4631 and the SAR st (Steep) source an average of 0.4311. The information classes in the SAR sources are apparently hard to discriminate.

Table 4.39

Information Classes, Training and Test Samples
Selected from the Anderson River Data Set.

Class #	Size	Information Class	Training	Testing
1	9715	Douglas Fir (31-40m)	971	8744
2	5511	Douglas Fir (21-30m)	551	4960
3	5480	Douglas Fir + Other Species (31-40m)	548	4932
4	5423	Douglas Fir + Lodgepole Pine (21-30m)	542	4881
5	3173	Hemlock + Cedar (31-40m)	317	2856
6	12600	Forest Clearings	1260	11340
Total	41902		4189	37713

Training samples are 10% of total. The training samples were selected UNIFORMLY over the image.

Data Sources:

s1 - ABMSS	(11 spectral data channels)
s2 - SAR sh	(4 radar data channels)
s3 - SAR st	(4 radar data channels)
s4 - Elevation	(1 elevation data channel)
s5 - Slope	(1 slope data channel)
s6 - Aspect	(1 aspect data channel)

Table 4.40

Pairwise JM Distances: ABMSS Data

Class #	2	3	4	5	6
1	0.73312	1.18274	1.31614	1.34177	1.01742
2	-	1.06912	1.33300	1.39373	1.21309
3	-	-	1.12051	1.36116	1.35036
4	-	-	-	1.24573	1.39253
5	-	-	-	-	1.39599
Average:	1.19877				

Table 4.41

Pairwise JM Distances: SAR Shallow Data.

Class #	2	3	4	5	6
1	0.57811	0.73556	0.63660	0.77470	0.54628
2	-	0.46706	0.40635	0.35228	0.20056
3	-	-	0.32671	0.37080	0.35582
4	-	-	-	0.38421	0.33648
5	-	-	-	-	0.34333
Average:	0.46305				

Table 4.42

Pairwise JM Distances: SAR Steep Data.

Class #	2	3	4	5	6
1	0.27652	0.41365	0.33141	0.51332	0.46221
2	-	0.39351	0.33445	0.39685	0.38786
3	-	-	0.45034	0.44442	0.40551
4	-	-	-	0.33897	0.61177
5	-	-	-	-	0.57957
Average:	0.43109				

The class-specific histograms of the topographic data sources are shown in Figures 4.6, 4.7 and 4.8. Looking at these figures it is seen that the class-specific histograms for all three data sources are highly overlapping. The elevation data (Figure 4.6) has the most distinct peaks for specific classes, the aspect data (Figure 4.8) has a few, but the slope data (Figure 4.7) can mostly only distinguish Douglas fir (31-40 m) from forest clearings. It is seen from the figures and Tables 4.40, 4.41 and 4.42 that the information classes in the Anderson River data set are very difficult to discriminate.

4.3.1 Results: Statistical Methods.

Four statistical classification methods were applied in the experiments performed here: 1) The minimum Euclidean distance (MD), 2) the maximum likelihood method for Gaussian data (ML), 3) the statistical multisource classifier (SMC) and 4) the linear opinion pool (LOP). The first two methods are "stacked vector" approaches but the other two are pooling methods which treat the data sources independently as previously discussed.

The results using the two stacked vector approaches are shown in Tables 4.43 (training) and 4.44 (test). Although the MD method did much better in classification of training and test data than for the Colorado data, it did significantly worse than the multivariate Gaussian ML method. It is questionable whether it is appropriate, from a theoretical standpoint, to use multivariate Gaussianity between all the sources for two reasons: first, because the topographic sources were not Gaussian; and second, because no information was available for modeling the dependencies between all the data sources. In view of this the ML method showed surprisingly good performance

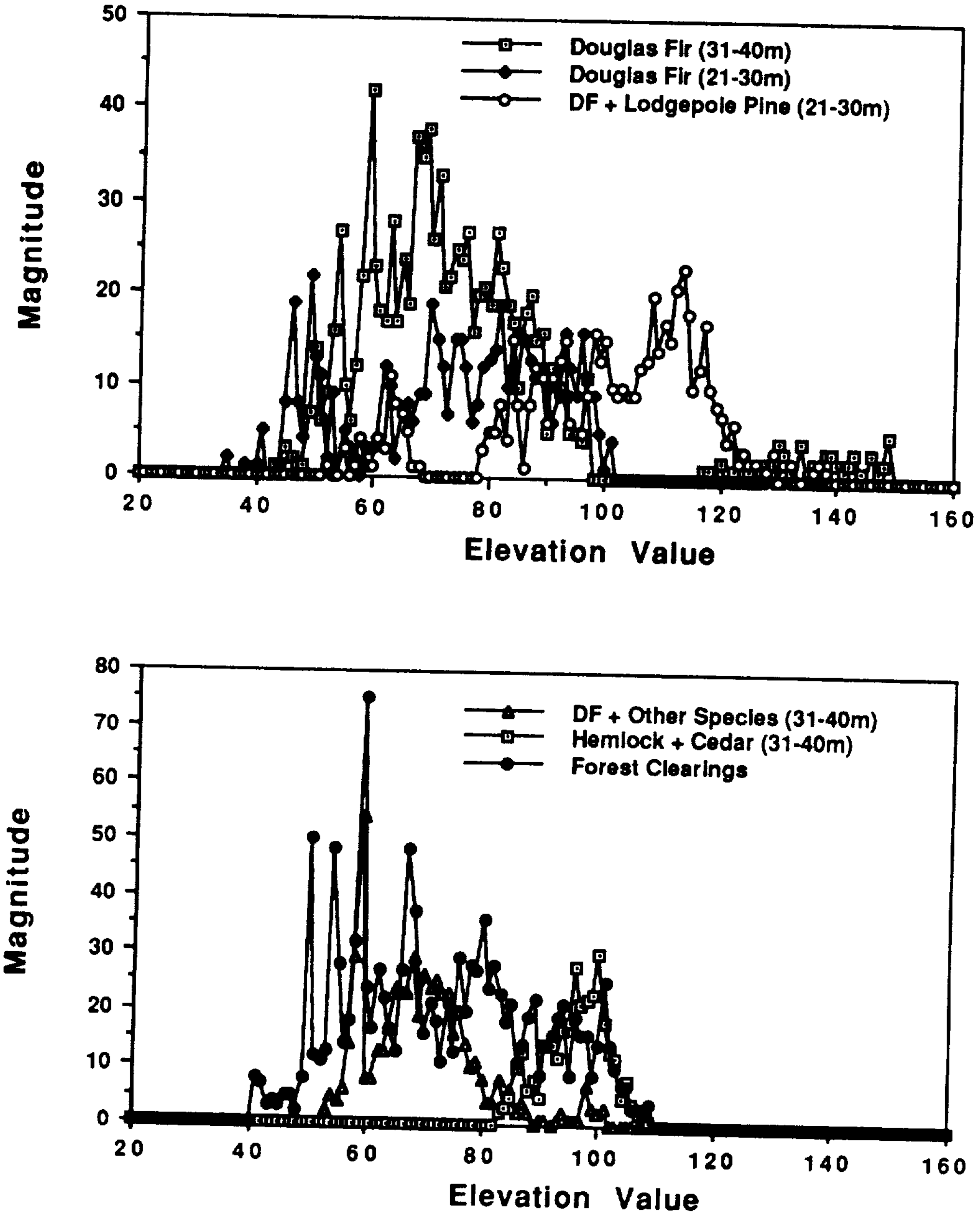
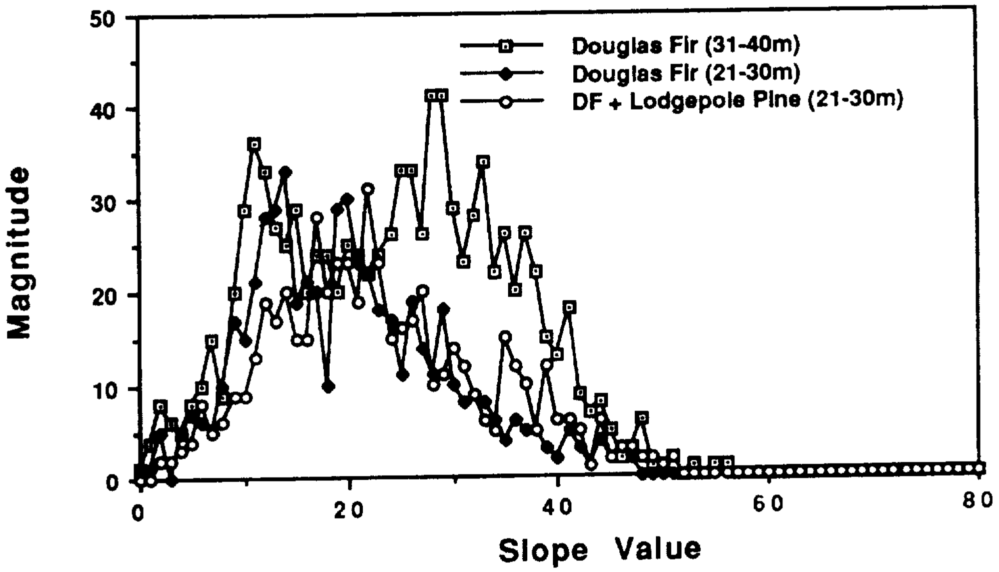


Figure 4.6 Class Histograms of Elevation Data in the Anderson River Data Set



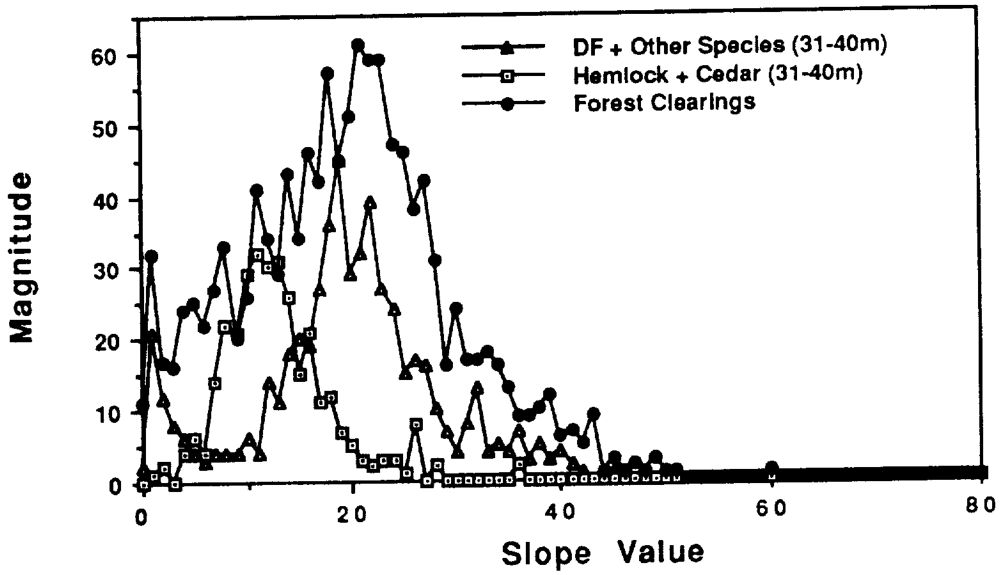


Figure 4.7 Class Histograms of Slope Data in the Anderson River Data Set

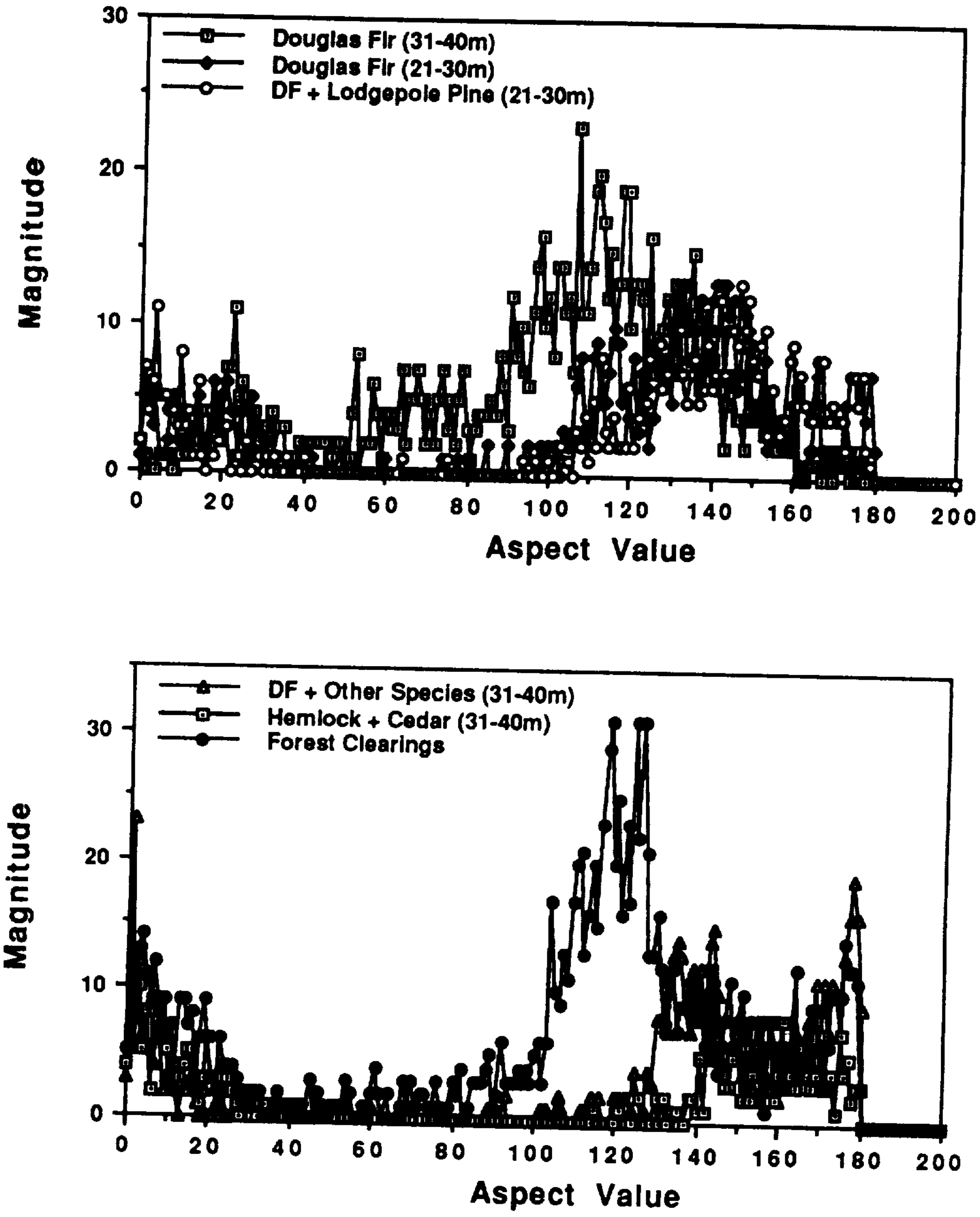


Figure 4.8 Class Histograms of Aspect Data in the Anderson River Data Set

Table 4.43

Classification Results of Training Samples for the Anderson River Data Set when the Minimum Euclidean Distance Method and the Maximum Likelihood Method for Gaussian Data are Applied.

Method	CPU	Percent Agreement with Reference for Class							
								OA	AVE
	time	1	2	3	4	5	6		
MD	68	40.4	8.9	47.6	67.7	42.3	72.4	50.51	46.55
ML	1095	54.6	31.6	87.8	90.9	81.4	73.3	68.23	69.92
# of pixels		971	551	548	542	317	1260	4189	4189

Table 4.44

Classification Results of Test Samples for the Anderson River Data Set when the Minimum Euclidean Distance Method and the Maximum Likelihood Method for Gaussian Data are Applied.

Method	Percent Agreement with Reference for Class							
							OA	AVE
	1	2	3	4	5	6		
MD	39.7	8.9	48.4	70.2	46.0	71.7	50.83	47.48
ML	50.8	27.7	84.5	81.9	73.8	72.0	64.30	65.12
# of pixels	8744	4960	4932	4881	2856	11340	37713	37713

in terms of training and test accuracy. Looking at Figures 4.6, 4.7 and 4.8 it is doubtful that the topographic sources should be modeled as Gaussian. However, the other three data sources (ABMSS, SAR sh, SAR st) can be modeled as Gaussian. Those three sources consist of 19 of the 22 data channels used in the classification. The number of the Gaussian channels is one of the reasons for the relatively good performance of the ML method.

Next the statistical pooling methods were applied. The class-specific correlation matrices were examined to make sure that the underlying independence assumptions of the SMC were not violated. In fact for one information class (Douglas fir + lodgepole pine (21-30m)), the elevation source was relatively highly correlated to the ABMSS data (the magnitudes of some correlations were as high as 0.71). Although this correlation was observed for one information class, the elevation data were used in the SMC classifications. However, the effect of removing the elevation data from the data set was investigated in the experiments. All other data sources were virtually uncorrelated.

All the data sources were trained on 9 data classes except the SAR data sources which showed better performance with only 6 data classes. As in Section 4.2.4 three density estimation methods (histogram, maximum penalized likelihood estimation and Parzen density estimation) were applied to model the (non-Gaussian) topographic data sources. The results for the different methods are discussed below.

Looking at the single source classifications in Table 4.45 (classifications of training samples), the ABMSS source was the best source in classification of training samples, both in terms of overall (49.84%) and average (50.53%) accuracies. The elevation data was second with overall accuracy of 40.75% and average accuracy of 40.47%. The aspect data came third (overall accuracy: 38.94%, average accuracy: 27.37%). The SAR data showed very poor performance (as seen in Tables 4.41 and 4.42, they were not separable). The SAR sh source was a little better (36.81% overall accuracy and 24.19%) than the SAR st source (overall accuracy: 36.57%, average accuracy: 23.50%). The slope source was the worst source with overall accuracy of 33.44% and average accuracy of 27.37%. The source-specific accuracy showed how difficult the data set is in classification. Using these classification accuracies as a reliability measure, the sources were ranked as: 1) ABMSS, 2) elevation, 3) aspect, 4) SAR sh, 5) SAR st and 6) slope. The equivocation measure (shown in Tables 4.47 and 4.48) ranked the sources somewhat differently. The equivocation ranking was: 1) ABMSS, 2) elevation, 3) aspect, 4) slope, 5) SAR sh, 6) SAR st. In the experiments weights were selected according to these different rankings.

Classifying all the data sources in Table 4.45 (training) with equal weights gave a significant improvement in both overall and average accuracies as compared to best single source classification (ABMSS). The overall accuracy was increased to 70.26% (or by 20.42%) and the average accuracy was increased to 69.89% (or by 19.36%). By changing the weights, the overall accuracy could only be improved to 70.40% and the average accuracy was increased to 69.95%. This was achieved by a weighting suggested by the

Table 4.45

Statistical Multisource Classification of Anderson River Data: Training Samples. Topographic Sources were Modeled by Histogram Approach.

	Percent Agreement with Reference for Class							
	1	2	3	4	5	6	OA	AVE
	Single Sources							
ABMSS	13.3	4.5	83.6	84.7	48.6	68.5	49.84	50.53
SAR sh	45.0	2.4	12.0	7.6	0.0	78.2	36.81	24.19
SAR st	35.5	1.3	4.0	12.9	1.3	86.0	36.57	23.50
Elevation	22.0	18.3	44.3	48.0	53.0	57.2	40.75	40.47
Slope	33.8	0.5	8.2	2.6	51.9	67.2	33.44	27.37
Aspect	42.6	25.0	52.2	17.7	17.4	51.0	38.94	34.32
m h t e s a	Multiple Sources							
1. 1. 1. 1. 1. 1.	70.0	35.2	79.0	78.2	81.0	75.8	70.26	69.89
1. .9 .9 1. 1. 1.	70.3	35.2	79.4	78.0	80.8	75.7	70.30	69.01
1. .8 .8 1. 1. 1.	70.6	35.4	79.2	78.2	80.4	75.8	70.40	69.95
1. .7 .7 1. 1. 1.	70.4	33.9	79.6	78.2	80.1	76.0	70.23	69.71
1. .6 .6 1. 1. 1.	70.8	33.4	79.2	78.2	80.8	76.1	70.28	69.74
1. .5 .5 1. 1. 1.	70.3	32.7	80.3	78.0	81.1	76.0	70.18	69.73
1. .4 .4 1. 1. 1.	69.7	31.9	81.2	78.0	80.1	75.7	69.92	69.46
1. .3 .3 1. 1. 1.	69.2	32.1	81.8	78.0	79.8	75.6	69.83	69.42
1. .2 .2 1. 1. 1.	69.8	32.7	82.3	77.9	80.1	75.5	70.09	67.71
1. .1 .1 1. 1. 1.	69.3	31.8	82.1	78.0	80.1	75.4	69.83	69.46
1. .0 .0 1. 1. 1.	68.8	31.8	81.9	78.0	80.1	74.9	69.54	69.26
1. 1. 1. .9 .9 .9	70.3	33.6	77.9	79.3	77.9	76.2	69.99	69.21
1. 1. 1. .8 .8 .8	71.1	31.8	77.7	79.2	75.4	76.3	69.71	68.56
1. 1. 1. .7 .7 .7	71.5	30.3	76.5	79.3	70.7	76.3	69.11	67.42
1. 1. 1. .6 .6 .6	71.6	28.1	75.9	78.8	64.0	76.3	68.20	65.78
1. 1. 1. .5 .5 .5	72.3	24.3	74.6	78.6	60.6	76.3	67.44	64.46
1. 1. 1. .4 .4 .4	72.4	20.7	73.0	79.0	57.7	76.4	66.63	63.20
1. 1. 1. .3 .3 .3	73.4	16.5	70.4	78.8	52.7	76.6	65.62	61.40
1. 1. 1. .2 .2 .2	73.4	10.0	66.4	78.8	48.9	76.6	63.95	59.02
1. 1. 1. .1 .1 .1	71.9	5.1	61.9	78.2	45.1	76.4	61.95	56.43
1. 1. 1. .0 .0 .0	70.2	3.6	55.5	77.7	42.0	76.5	60.25	54.25
1. 1. 1. .0 1. 1.	71.6	16.9	76.8	78.0	63.4	75.8	66.56	63.75
1. .9 .9 .9 .9 .9	70.5	33.2	78.6	79.3	77.9	75.9	69.99	69.26
1. .8 .8 .9 .9 .9	70.6	33.0	78.8	79.3	77.9	76.1	70.09	69.31
1. .8 .8 1. .9 .9	70.2	35.2	79.0	79.1	79.5	76.0	70.37	69.86
1. .8 .8 1. .9 1.	70.2	34.5	78.8	79.0	79.2	76.0	70.18	69.61
1. .8 .8 1. .8 1.	69.6	34.7	78.7	79.4	78.2	76.0	70.04	69.42
1. .8 .8 .8 .8 .8	71.3	30.7	78.1	79.2	75.1	76.3	69.66	68.44
# of pixels	971	551	548	542	317	1260	4189	4189

The columns labeled m h t e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

CPU time for training and classification: 402 sec.

Table 4.46

Statistical Multisource Classification of Anderson River Data: Test Samples. Topographic Sources were Modeled by Histogram Approach.

	Percent Agreement with Reference for Class							
	1	2	3	4	5	6	OA	AVE
				Single Sources				
ABMSS	12.4	4.1	81.0	80.5	49.5	67.1	48.34	49.10
SAR sh	44.4	1.9	13.1	7.9	0.0	77.6	36.62	24.15
SAR st	34.2	0.9	3.8	13.4	0.7	85.5	36.02	23.07
Elevation	18.3	15.9	43.1	47.3	50.9	55.3	38.56	38.47
Slope	32.1	0.4	6.9	1.8	50.9	64.9	32.03	26.17
Aspect	39.0	22.7	43.9	13.5	11.3	46.5	34.36	29.48
m h t e s a				Multiple Sources				
1. 1. 1. 1. 1. 1.	68.2	31.6	75.0	77.4	78.7	74.7	68.19	67.59
1. .9 .9 1. 1. 1.	68.1	31.2	75.0	77.5	78.6	74.8	68.19	67.53
1. .8 .8 1. 1. 1.	67.7	31.1	74.9	77.6	78.7	74.9	68.13	67.50
1. .7 .7 1. 1. 1.	67.9	30.6	75.2	77.6	78.6	74.8	68.10	67.46
1. .6 .6 1. 1. 1.	67.7	30.4	75.2	77.6	78.5	74.8	68.00	67.36
1. .5 .5 1. 1. 1.	67.6	30.0	75.1	77.5	78.4	74.7	67.88	67.23
1. .4 .4 1. 1. 1.	67.5	29.7	75.1	77.5	78.3	74.7	67.79	67.13
1. .3 .3 1. 1. 1.	67.3	29.0	74.9	77.5	78.2	74.6	67.61	66.93
1. .2 .2 1. 1. 1.	67.1	28.6	74.7	77.5	78.2	74.5	67.43	66.75
1. .1 .1 1. 1. 1.	66.8	28.1	74.4	77.4	78.2	74.3	67.20	66.53
1. .0 .0 1. 1. 1.	66.4	27.8	73.9	77.4	78.3	74.2	66.96	66.32
1. 1. 1. .9 .9 .9	68.5	30.3	74.1	77.9	76.1	74.9	67.93	66.96
1. 1. 1. .8 .8 .8	68.7	28.6	72.9	78.4	72.6	75.3	67.49	66.06
1. 1. 1. .7 .7 .7	69.2	26.7	71.8	78.4	68.9	75.5	67.01	65.08
1. 1. 1. .6 .6 .6	69.5	23.9	70.4	78.5	64.0	75.7	66.25	63.68
1. 1. 1. .5 .5 .5	70.2	20.8	68.9	78.4	59.8	75.9	65.54	62.35
1. 1. 1. .4 .4 .4	70.9	16.9	67.0	78.3	55.6	76.3	64.71	60.83
1. 1. .3 .3 .3 .3	71.6	12.5	64.9	78.0	52.0	76.5	63.77	59.24
1. 1. 1. .2 .2 .2	72.4	8.1	62.0	77.8	47.9	76.7	62.72	57.48
1. 1. 1. .1 .1 .1	72.5	4.0	58.1	77.4	44.5	76.8	61.41	55.55
1. 1. 1. .0 .0 .0	70.8	1.9	52.9	77.1	41.6	76.9	59.83	53.52
1. 1. 1. .0 1. 1.	70.4	13.4	71.7	75.7	59.8	75.4	64.45	61.06
1. .9 .9 .9 .9 .9	68.3	30.1	74.1	78.0	76.2	75.1	67.93	66.96
1. .8 .8 .9 .9 .9	68.3	29.6	74.3	78.0	76.2	75.1	67.90	66.93
1. .8 .8 1. .9 .9	68.0	31.3	74.2	77.9	77.9	75.0	68.13	67.39
1. .8 .8 1. .9 1.	67.7	31.0	74.8	77.8	77.9	74.9	68.03	67.34
1. .8 .8 1. .8 1.	67.7	30.8	74.7	78.0	77.3	74.9	68.00	67.23
1. .8 .8 .8 .8 .8	68.9	28.0	73.2	78.5	72.6	75.4	67.55	66.10
# of pixels	8744	4960	4932	4881	2856	11340	37713	37713

The columns labeled m h t e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

reliability measure, the sources were ranked as: 1) ABMSS, 2) elevation, 3) aspect, 4) SAR sh, 5) SAR st and 6) slope. The equivocation measure (shown in Tables 4.47 and 4.48) ranked the sources somewhat differently. The equivocation ranking was: 1) ABMSS, 2) elevation, 3) aspect, 4) slope, 5) SAR sh, 6) SAR st. In the experiments, weights were selected to reflect the rankings implied by the reliability measures.

Classifying all the data sources in Table 4.45 (training) with equal weights gave a significant improvement in both overall and average accuracies as compared to best single-source classification (ABMSS). The overall accuracy was increased to 70.26% (or by 20.42%) and the average accuracy was increased to 69.89% (or by 19.36%). By changing the weights, the overall accuracy could only be improved to 70.40% and the average accuracy was increased to 69.95%. This was achieved by a weighting suggested by the equivocation measure (weights: all sources 1, except the SAR sources were weighted 0.8).

The results in Table 4.45 are interesting. Removing the SAR sources (1.0,0.0,0.0,1.0,1.0,1.0) reduced the classification accuracy only slightly (OA: 69.54%, AVE: 69.26%); removing the elevation source (1.0,1.0,1.0,0.0,1.0,1.0) had a much more significant effect on the results (OA: 66.56%, AVE: 63.75%). Thus the results showed that it was helpful to use the elevation source in classification even though that source had some class-specific dependence to the ABMSS data.

Looking at the test results in Table 4.46 a similar performance was seen as in single-source classifications of training data. For most of the data sources the accuracies were predictably a little lower than in the training case.

Table 4.47

The Equivocations of the Gaussian Data Sources.

Sensor	ABMSS	SAR Shallow	SAR Steep
Equivocation	1.141	1.621	1.636

Table 4.48

The Equivocations of the Non-Gaussian Data Sources with Regard to the Three Modeling Methods Used.

Data Source	Histogram Estimation	Maximum Penalized Likelihood Method	Parzen Estimation
Elevation	1.430	1.430	1.429
Slope	1.620	1.620	1.626
Aspect	1.532	1.539	1.557

The accuracies of the SAR sources were almost the same for training and test data. Also, these sources had higher accuracies for test data than the aspect source. The similarity of training and test results indicates that the training sample was apparently representative.

When all the data sources in Table 4.46 were classified with equal weights, the overall and average accuracies improved substantially in comparison to the ABMSS classification (OA: 48.34%, AVE: 49.10%). The overall accuracy increased to 68.19% or by 19.85%. The average accuracy improved to 67.59% or by 18.49%. When the weights were changed, neither higher overall nor average accuracies could be reached. Several of the weights showed similar performance to the equal weights, but none was higher. The result of discarding the elevation source (1.0,1.0,1.0,0.0,1.0,1.0) was again significantly lower (OA: 64.45%, AVE: 61.06%) than when equal weights were used. This result, along with the similar training result, showed that the elevation source should be included in the multisource classification even though it had significant correlation with the ABMSS data. The results in Tables 4.45 and 4.46 showed that the SMC method outperforms the ML method (Tables 4.43 and 4.44) both in terms of classification accuracy and classification time. The SMC was significantly faster, needing only 402 CPU sec (training and test) for the six-source composite, whereas the ML method needed 1095 CPU sec.

The results using the LOP are shown in Tables 4.49 and 4.50. These results were somewhat similar to the LOP results for the Colorado data. When all the data sources were classified with equal weights (training), the overall and average classification accuracies were lower as compared to the

Table 4.49

Linear Opinion Pool Applied in Classification of Anderson River Data: Training Samples. Topographic Sources were Modeled by Histogram Approach.

	Percent Agreement with Reference for Class							
	1	2	3	4	5	6	OA	AVE
				Single Sources				
ABMSS	13.3	4.5	83.6	84.7	48.6	68.5	49.84	50.53
SAR sh	45.0	2.4	12.0	7.6	0.0	78.2	36.81	24.19
SAR st	35.5	1.3	4.0	12.9	1.3	86.0	36.57	23.50
Elevation	22.0	18.3	44.3	48.0	53.0	57.2	40.75	40.47
Slope	33.8	0.5	8.2	2.6	51.9	67.2	33.44	27.37
Aspect	42.6	25.0	52.2	17.7	17.4	51.0	38.94	34.32
m h t e s a				Multiple Sources				
1. 1. 1. 1. 1. 1.	55.0	0.0	0.0	49.6	0.0	95.3	47.84	33.32
1. .9 .9 1. 1. 1.	56.0	0.0	0.0	52.8	0.0	95.2	48.46	34.01
1. .8 .8 1. 1. 1.	56.6	0.0	0.0	55.7	0.0	95.2	48.96	34.59
1. .7 .7 1. 1. 1.	58.2	0.0	0.0	60.5	0.0	94.9	49.87	35.60
1. .6 .6 1. 1. 1.	59.2	0.0	0.0	64.4	0.3	94.6	50.54	36.42
1. .5 .5 1. 1. 1.	60.9	0.0	0.0	68.5	1.9	94.3	51.47	37.58
1. .4 .4 1. 1. 1.	62.2	0.0	0.0	71.0	4.7	93.8	52.18	38.63
1. .3 .3 1. 1. 1.	65.2	0.0	0.0	73.2	9.8	93.2	53.35	40.23
1. .2 .2 1. 1. 1.	68.7	0.0	0.0	74.2	12.0	92.6	54.29	41.24
1. .1 .1 1. 1. 1.	69.5	0.0	0.0	74.9	16.4	91.1	54.45	41.99
1. .0 .0 1. 1. 1.	70.8	0.2	0.0	76.2	19.9	90.4	54.98	42.90
1. 1. 1. .9 .9 .9	55.9	0.0	0.0	53.5	0.0	95.1	48.48	34.08
1. 1. 1. .8 .8 .8	57.5	0.0	0.0	58.5	0.0	94.7	49.37	35.11
1. 1. 1. .7 .7 .7	58.7	0.0	0.0	62.0	0.0	94.4	50.01	35.84
1. 1. 1. .6 .6 .6	59.1	0.0	0.0	65.3	0.3	93.9	50.42	36.44
1. 1. 1. .5 .5 .5	60.6	0.0	0.0	68.3	2.2	93.4	51.13	37.41
1. 1. 1. .4 .4 .4	62.6	0.0	0.0	70.3	7.9	92.9	52.16	38.96
1. 1. 1. .3 .3 .3	64.3	0.0	0.0	71.4	14.8	92.9	53.19	40.56
1. 1. 1. .2 .2 .2	66.0	0.0	0.0	72.7	18.6	91.9	53.76	41.54
1. 1. 1. .1 .1 .1	67.5	0.0	0.0	73.6	21.1	90.5	53.97	42.11
1. 1. 1. .0 .0 .0	68.2	0.0	0.0	73.1	24.3	89.4	54.00	42.50
1. 1. 1. .0 1. 1.	58.6	0.0	0.0	30.8	0.0	94.9	46.12	30.72
1. .9 .9 .9 .9 .9	56.7	0.0	0.0	57.6	0.0	94.8	49.10	34.85
1. .8 .8 .9 .9 .9	58.4	0.0	0.0	60.7	0.0	94.6	49.84	35.62
1. .8 .8 1. .9 .9	57.8	0.0	0.0	60.7	0.0	94.7	49.73	35.53
1. .8 .8 1. .9 1.	57.0	0.0	0.0	58.7	0.0	94.9	49.34	35.09
1. .8 .8 1. .8 1.	57.8	0.0	0.0	60.9	0.0	94.8	49.77	35.57
1. .8 .8 .8 .8 .8	60.7	0.0	0.0	66.1	0.0	94.0	50.87	36.78
# of pixels	971	551	548	542	317	1260	4189	4189

The columns labeled m h t e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

CPU time for training and classification: 376 sec.

Table 4.50

Linear Opinion Pool Applied in Classification of Anderson River Data: Test Samples. Topographic Sources were Modeled by Histogram Approach.

	Percent Agreement with Reference for Class							
	1	2	3	4	5	6	OA	AVE
				Single Sources				
ABMSS	12.4	4.1	81.0	80.5	49.5	67.1	48.34	49.10
SAR sh	44.4	1.9	13.1	7.9	0.0	77.6	36.62	24.15
SAR st	34.2	0.9	3.8	13.4	0.7	85.5	36.02	23.07
Elevation	18.3	15.9	43.1	47.3	50.9	55.3	38.56	38.47
Slope	32.1	0.4	6.9	1.8	50.9	64.9	32.03	26.17
Aspect	39.0	22.7	43.9	13.5	11.3	46.5	34.36	29.48
m h t e s a				Multiple Sources				
1. 1. 1. 1. 1. 1.	51.0	0.0	0.0	48.8	0.0	95.7	46.91	32.58
1. .9 .9 1. 1. 1.	51.7	0.0	0.0	50.8	0.0	95.5	47.29	33.01
1. .8 .8 1. 1. 1.	52.8	0.0	0.0	53.6	0.0	95.3	47.83	33.61
1. .7 .7 1. 1. 1.	53.9	0.0	0.0	56.8	0.1	95.0	48.42	34.30
1. .6 .6 1. 1. 1.	55.1	0.0	0.0	60.0	0.3	94.7	49.05	35.03
1. .5 .5 1. 1. 1.	56.8	0.0	0.0	63.1	0.9	94.3	49.76	35.85
1. .4 .4 1. 1. 1.	59.0	0.0	0.0	66.7	2.2	93.5	50.59	36.90
1. .3 .3 1. 1. 1.	61.0	0.0	0.0	69.6	4.4	92.8	51.37	37.95
1. .2 .2 1. 1. 1.	63.4	0.0	0.0	71.5	7.9	91.7	52.13	39.09
1. .1 .1 1. 1. 1.	65.6	0.0	0.0	73.1	12.0	90.4	52.76	40.19
1. .0 .0 1. 1. 1.	67.7	0.0	0.0	74.4	16.1	88.9	53.30	41.20
1. 1. 1. .9 .9 .9	52.7	0.0	0.0	51.2	0.0	95.6	47.58	33.25
1. 1. 1. .8 .8 .8	54.4	0.0	0.0	55.1	0.0	95.2	48.39	34.13
1. 1. 1. .7 .7 .7	56.3	0.0	0.0	58.9	0.0	94.9	49.20	35.02
1. 1. 1. .6 .6 .6	57.5	0.0	0.0	62.3	0.4	94.4	49.80	35.76
1. 1. 1. .5 .5 .5	59.6	0.0	0.0	65.2	2.1	93.8	50.63	36.79
1. 1. 1. .4 .4 .4	61.5	0.0	0.0	67.5	4.9	93.1	51.36	37.84
1. 1. 1. .3 .3 .3	63.6	0.0	0.0	69.5	8.5	92.3	52.14	38.98
1. 1. 1. .2 .2 .2	65.7	0.0	0.0	71.3	12.8	91.4	52.91	40.19
1. 1. 1. .1 .1 .1	67.3	0.0	0.0	72.6	16.7	90.3	53.44	41.16
1. 1. 1. .0 .0 .0	68.9	0.0	0.0	73.1	20.8	89.3	53.89	42.03
1. 1. 1. .0 1. 1.	56.8	0.0	0.0	26.1	0.0	95.1	45.14	29.66
1. .9 .9 .9 .9 .9	53.8	0.0	0.0	54.5	0.0	95.3	48.16	33.92
1. .8 .8 .9 .9 .9	54.9	0.0	0.0	57.5	0.1	94.9	48.71	34.56
1. .8 .8 1. .9 .9	54.3	0.0	0.0	56.9	0.1	95.1	48.53	34.38
1. .8 .8 1. .9 1.	53.4	0.0	0.0	55.4	0.0	95.1	48.15	33.98
1. .8 .8 1. .8 1.	54.3	0.0	0.0	57.3	0.1	94.9	48.56	34.44
1. .8 .8 .8 .8 .8	57.2	0.0	0.0	61.4	0.4	94.5	49.65	35.59
# of pixels	8744	4960	4932	4881	2856	11340	37713	37713

The columns labeled m h t e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

best single-source classification. The highest overall accuracies in Table 4.49 were achieved when the SAR sources were discarded altogether (weights: 1.0,0.0,0.0,1.0,1.0,1.0). This best result was an overall accuracy of 54.98% and average accuracy of 42.90%, significantly worse than the results achieved by the SMC and the ML methods. Another "good" result was achieved when the topographic sources were discarded (weights: 1.0,1.0,1.0,0.0,0.0,0.0). By discarding these three sources, the LOP gave overall accuracy of 54.00% and average accuracy of 42.50%. These best two results showed that the LOP tended to give the ABMSS source something close to dictatorship. The LOP was especially poor in terms of average accuracy. It did not distinguish well between information classes, and three of them were most of the time not classified correctly at all.

The test results for the LOP (Table 4.50) were similar to the training results. The major difference was that the highest overall and average accuracies were now achieved when the topographic sources were discardeded (weights: 1.0,1.0,1.0,0.0,0.0,0.0). The best overall accuracy was 53.89% and the highest average accuracy was 42.03%. The results in Tables 4.49 and 4.50 show clearly that not much can be expected from the LOP in classification of the Anderson River data.

b) Topographic Data Modeled by Maximum Penalized Likelihood Method

The topographic data were now modeled by the maximum penalized likelihood method using the IMSL program D3SPL. The smoothing parameter (γ) giving the highest classification accuracies for training and test data was chosen as the smoothing parameter to be used in the experiments reported here. The smoothing parameter that gave the best results for all the sources was $\gamma=1.0$. By looking at Tables 4.51 (training) and 4.52 (test) the single-source classification results using the maximum penalized likelihood method are seen to be very similar to the histogram results in Tables 4.45 and 4.46. The histogram approach showed a little better accuracy for training data, but the maximum likelihood method was slightly better in overall classification accuracy of test data. The reliability measure using overall classification accuracy ranked the sources in the same way as for its counterpart with the histogram estimation. The equivocation reliability measure (see Tables 4.47 and 4.48) also ranked the sources in the same way as the equivocation for the histogram estimation.

Looking at the SMC classification of training data in Table 4.51, it is seen that the highest overall and average classification accuracies were achieved when all the sources were combined with equal weights. The overall accuracy (70.47%) and average accuracy (70.05%) were a little higher than were achieved with the histogram approach in Table 4.45. Several good results with the different weights are reported in Table 4.51 but none are better than those achieved with equal weights. The test results are shown in Table 4.52. The highest accuracies there, as in the previous table, were achieved when all the sources had equal weights. The overall accuracy (68.20%) was just above the accuracy with the histogram approach in Table 4.46, but the average accuracy (67.48%) was slightly less than with the histogram approach. For the Anderson River data, these results indicate that there is not much difference in using the maximum penalized likelihood method rather than the

Table 4.51

Statistical Multisource Classification of Anderson River Data: Training Samples. Topographic Sources were Modeled by Maximum Penalized Likelihood Method.

	Percent Agreement with Reference for Class							
	1	2	3	4	5	6	OA	AVE
				Single Sources				
ABMSS	13.3	4.5	83.6	84.7	48.6	68.5	49.84	50.53
SAR sh	45.0	2.4	12.0	7.6	0.0	78.2	36.81	24.19
SAR st	35.5	1.3	4.0	12.9	1.3	86.0	36.57	23.50
Elevation	18.6	16.3	47.3	48.2	48.6	59.3	40.39	39.71
Slope	32.5	0.5	8.2	2.4	52.1	68.2	33.44	27.32
Aspect	39.2	22.0	51.5	18.8	17.4	54.8	38.96	33.94
m h t e s a				Multiple Sources				
1. 1. 1. 1. 1. 1.	70.8	35.8	78.6	79.0	80.4	75.7	70.47	70.05
1. .9 .9 1. 1. 1.	71.0	34.5	79.0	78.6	79.8	75.8	70.33	69.78
1. .8 .8 1. 1. 1.	70.8	33.6	78.6	78.6	79.5	76.0	70.14	69.50
1. .7 .7 1. 1. 1.	71.0	33.6	78.8	79.0	79.2	76.0	70.26	69.59
1. .6 .6 1. 1. 1.	71.0	33.2	78.6	79.0	79.8	76.2	70.28	69.63
1. .5 .5 1. 1. 1.	70.6	32.1	79.4	78.8	79.8	76.0	70.06	69.45
1. .4 .4 1. 1. 1.	70.3	31.8	80.3	78.8	80.1	75.7	70.02	69.50
1. .3 .3 1. 1. 1.	70.2	31.8	81.8	78.8	79.5	75.6	70.09	69.60
1. .2 .2 1. 1. 1.	70.2	31.8	81.4	78.6	79.5	75.4	69.97	69.48
1. .1 .1 1. 1. 1.	69.7	30.9	81.6	78.6	79.8	75.4	69.78	69.32
1. .0 .0 1. 1. 1.	69.4	30.9	81.0	78.6	78.9	74.9	69.42	68.95
1. 1. 1. .9 .9 .9	71.1	32.5	77.9	79.3	77.9	75.9	69.92	69.10
1. 1. 1. .8 .8 .8	71.7	31.0	77.0	79.3	74.4	76.2	69.59	68.28
1. 1. 1. .7 .7 .7	72.0	29.4	76.1	79.5	69.7	76.2	68.99	67.15
1. 1. 1. .6 .6 .6	72.2	26.9	75.9	78.8	64.0	76.3	68.18	65.68
1. 1. 1. .5 .5 .5	72.7	24.0	74.5	78.6	60.3	76.4	67.46	64.40
1. 1. 1. .4 .4 .4	72.9	20.1	72.6	78.8	57.4	76.3	66.56	63.04
1. 1. 1. .3 .3 .3	73.8	16.0	70.3	78.8	52.4	76.5	65.58	61.29
1. 1. 1. .2 .2 .2	73.4	9.6	66.1	78.8	48.6	76.5	63.81	58.83
1. 1. 1. .1 .1 .1	72.0	4.7	61.7	78.2	45.1	76.5	61.92	56.37
1. 1. 1. .0 .0 .0	70.2	3.6	55.5	77.7	42.0	76.5	60.25	54.25
1. 1. 1. .0 1. 1.	72.3	16.0	76.3	78.0	62.5	76.0	66.51	63.50
1. .9 .9 .9 .9 .9	71.4	32.3	78.5	79.3	77.9	76.0	70.06	69.22
1. .8 .8 .9 .9 .9	71.0	32.1	78.2	79.3	77.6	76.2	69.97	69.08
1. .8 .8 1. .9 .9	71.0	33.2	78.6	79.2	79.2	76.0	70.21	69.53
1. .8 .8 1. .9 1.	70.6	33.2	78.6	79.3	79.2	76.0	70.16	69.51
1. .8 .8 1. .8 1.	70.8	33.4	78.5	79.5	78.5	76.1	70.18	69.47
1. .8 .8 .8 .8 .8	71.9	30.1	77.6	79.2	73.8	76.3	69.56	68.15
# of pixels	971	551	548	542	317	1260	4189	4189

The columns labeled m h t e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

CPU time for training and classification: 926 sec.

Table 4.52

Statistical Multisource Classification of Anderson River Data: Test Samples. Topographic Sources were Modeled by Maximum Penalized Likelihood Method.

	Percent Agreement with Reference for Class							
	1	2	3	4	5	6	OA	AVE
				Single Sources				
ABMSS	12.4	4.1	81.0	80.5	49.5	67.1	48.34	49.10
SAR sh	44.4	1.9	13.1	7.9	0.0	77.6	36.62	24.15
SAR st	34.2	0.9	3.8	13.4	0.7	85.5	36.02	23.07
Elevation	15.6	14.4	46.8	47.8	47.1	57.3	38.64	38.18
Slope	30.9	0.4	6.9	2.0	50.9	65.5	31.94	26.11
Aspect	35.9	19.9	43.1	15.2	11.7	50.2	34.54	29.34
m h t e s a				Multiple Sources				
1. 1. 1. 1. 1. 1.	68.8	31.3	74.3	77.6	78.2	74.7	68.20	67.48
1. .9 .9 1. 1. 1.	68.6	30.8	74.2	77.7	78.0	74.8	68.12	67.36
1. .8 .8 1. 1. 1.	68.6	30.3	74.1	77.8	77.9	74.9	68.07	67.27
1. .7 .7 1. 1. 1.	68.5	29.9	74.2	77.8	77.9	74.9	67.99	67.18
1. .6 .6 1. 1. 1.	68.4	29.6	74.1	77.8	77.9	74.8	67.91	67.10
1. .5 .5 1. 1. 1.	68.3	29.5	74.1	77.8	77.9	74.7	67.87	67.08
1. .4 .4 1. 1. 1.	68.1	29.0	74.3	77.8	77.8	74.7	67.74	66.94
1. .3 .3 1. 1. 1.	68.1	28.5	74.1	77.9	77.8	74.7	67.65	66.84
1. .2 .2 1. 1. 1.	67.9	27.9	73.7	77.8	77.7	74.5	67.41	66.60
1. .1 .1 1. 1. 1.	67.8	27.4	73.2	77.7	77.8	74.3	67.19	66.36
1. .0 .0 1. 1. 1.	67.3	26.8	72.9	77.6	77.6	74.3	66.92	66.09
1. 1. 1. .9 .9 .9	69.0	29.8	73.3	78.1	75.5	75.0	67.87	66.77
1. 1. 1. .8 .8 .8	69.1	28.5	72.3	78.4	72.1	75.3	67.48	65.94
1. 1. 1. .7 .7 .7	69.4	26.4	71.3	78.5	68.3	75.5	66.93	64.92
1. 1. 1. .6 .6 .6	70.0	23.5	69.8	78.5	63.6	75.8	66.22	63.53
1. 1. 1. .5 .5 .5	70.7	20.2	68.6	78.4	59.6	76.0	65.54	62.25
1. 1. 1. .4 .4 .4	71.4	16.6	66.5	78.3	55.3	76.3	64.71	60.74
1. 1. 1. .3 .3 .3	71.9	12.1	64.5	78.0	51.8	76.6	63.73	59.15
1. 1. 1. .2 .2 .2	72.7	7.7	61.7	77.8	47.8	76.7	62.69	57.40
1. 1. 1. .1 .1 .1	72.6	3.9	57.8	77.4	44.5	76.7	61.37	55.50
1. 1. 1. .0 .0 .0	70.8	1.9	52.9	77.1	41.6	76.9	59.83	53.53
1. 1. 1. .0 1. 1.	71.1	12.6	71.1	75.8	59.3	75.4	64.41	60.88
1. .9 .9 .9 .9 .9	68.8	29.4	73.2	78.2	75.7	75.2	67.84	66.74
1. .8 .8 .9 .9 .9	69.1	29.0	73.3	78.1	75.8	75.2	67.87	66.75
1. .8 .8 1. .9 .9	68.6	30.6	73.4	78.0	77.3	75.0	68.04	67.15
1. .8 .8 1. .9 1.	68.3	30.2	74.0	77.9	77.6	74.9	67.97	67.15
1. .8 .8 1. .8 1.	68.4	30.2	73.9	78.3	76.6	74.9	67.95	67.05
1. .8 .8 .8 .8 .8	69.5	27.3	72.2	78.6	71.7	75.4	67.42	65.78
# of pixels	8744	4960	4932	4881	2856	11340	37713	37713

The columns labeled m h t e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

histogram approach for modeling the topographic data.

The results using the LOP with the maximum penalized likelihood method are shown in Tables 4.53 (training) and 4.54 (test). The results using the maximum penalized likelihood method were for the most part slightly better than the results with the histograms (Tables 4.49 and 4.50). The weaknesses of the LOP were evident regardless of the density estimation method used. The LOP did an extremely poor job in classifying classes 2, 3 and 5. The highest accuracies for training (Table 4.49) were reached with the same weights as with the histograms (discard the SAR sources; weights: 1.0,0.0,0.0,1.0,1.0,1.0). The "best" overall and average accuracies for training data with the maximum penalized likelihood method (OA: 55.10%, AVE: 42.98%) were higher than the ones with the histogram approach. The "best" result for test data (Table 4.54) was the same as with the histogram approach. This "best" test result was reached when the topographic sources were discarded; varying the density estimation method had no effect.

c) Topographic Data Modeled by Parzen Density Estimation

Finally, the topographic data sources were modeled by Parzen density estimation using a Gaussian kernel function. The following smoothing parameters gave the best results and were consequently used:

1) Elevation data: $\sigma = 0.5$

2) Slope data: $\sigma = 0.75$

3) Aspect data: $\sigma = 1.0$

Table 4.53

Linear Opinion Pool Applied in Classification of Anderson River Data: Training Samples. Topographic Sources were Modeled by Maximum Penalized Likelihood Method.

	Percent Agreement with Reference for Class							
	1	2	3	4	5	6	OA	AVE
				Single Sources				
ABMSS	13.3	4.5	83.6	84.7	48.6	68.5	49.84	50.53
SAR sh	45.0	2.4	12.0	7.6	0.0	78.2	36.81	24.19
SAR st	35.5	1.3	4.0	12.9	1.3	86.0	36.57	23.50
Elevation	18.6	16.3	47.3	48.2	48.6	59.3	40.39	39.71
Slope	32.5	0.5	8.2	2.4	52.1	68.2	33.44	27.32
Aspect	39.2	22.0	51.5	18.8	17.4	54.8	38.96	33.94
m h t e s a				Multiple Sources				
1. 1. 1. 1. 1. 1.	55.2	0.0	0.0	49.8	0.0	95.5	47.96	33.42
1. .9 .9 1. 1. 1.	56.7	0.0	0.0	53.5	0.0	95.4	48.77	34.27
1. .8 .8 1. 1. 1.	57.6	0.0	0.0	55.9	0.0	95.2	49.22	34.79
1. .7 .7 1. 1. 1.	58.4	0.0	0.0	60.3	0.0	94.8	49.87	35.59
1. .6 .6 1. 1. 1.	59.5	0.0	0.0	64.2	0.0	94.4	50.51	36.36
1. .5 .5 1. 1. 1.	60.8	0.0	0.0	68.3	1.6	94.4	51.42	37.50
1. .4 .4 1. 1. 1.	62.6	0.0	0.0	71.4	4.4	93.8	52.30	38.71
1. .3 .3 1. 1. 1.	65.8	0.0	0.0	73.1	9.5	93.4	53.52	40.29
1. .2 .2 1. 1. 1.	69.3	0.0	0.0	74.2	12.0	92.4	54.36	41.31
1. .1 .1 1. 1. 1.	70.5	0.2	0.0	74.9	16.4	91.2	54.74	42.20
1. .0 .0 1. 1. 1.	71.8	0.2	0.0	76.4	19.6	90.0	55.10	42.98
1. 1. 1. .9 .9 .9	56.4	0.0	0.0	53.7	0.0	95.3	48.70	34.24
1. 1. 1. .8 .8 .8	58.3	0.0	0.0	58.9	0.0	94.7	49.61	35.30
1. 1. 1. .7 .7 .7	59.1	0.0	0.0	62.2	0.0	94.4	50.16	35.96
1. 1. 1. .6 .6 .6	60.8	0.0	0.0	65.5	0.3	93.9	50.82	36.74
1. 1. 1. .5 .5 .5	61.2	0.0	0.0	68.5	2.2	93.5	51.32	37.55
1. 1. 1. .4 .4 .4	62.8	0.0	0.0	70.3	7.9	92.9	52.21	38.99
1. 1. 1. .3 .3 .3	64.4	0.0	0.0	71.4	14.8	92.8	53.19	40.56
1. 1. 1. .2 .2 .2	66.1	0.0	0.0	72.7	18.6	92.0	53.81	41.57
1. 1. 1. .1 .1 .1	67.6	0.0	0.0	73.6	21.1	90.5	54.00	42.13
1. 1. 1. .0 .0 .0	68.2	0.0	0.0	73.1	24.3	89.4	54.00	42.50
1. 1. 1. .0 1. 1.	59.1	0.0	0.0	30.1	0.0	94.8	46.12	30.67
1. .9 .9 .9 .9 .9	57.7	0.0	0.0	57.7	0.0	94.8	49.37	35.04
1. .8 .8 .9 .9 .9	58.8	0.0	0.0	61.4	0.0	94.5	50.01	35.79
1. .8 .8 1. .9 .9	58.1	0.0	0.0	60.9	0.0	94.7	49.82	35.61
1. .8 .8 1. .9 1.	58.0	0.0	0.0	58.9	0.0	95.1	49.65	35.32
1. .8 .8 1. .8 1.	58.4	0.0	0.0	61.1	0.0	94.8	49.96	35.72
1. .8 .8 .8 .8 .8	61.2	0.0	0.0	65.9	0.0	94.0	50.99	36.84
# of pixels	971	551	548	542	317	1260	4189	4189

The columns labeled m h t e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

CPU time for training and classification: 900 sec.

Table 4.54

Linear Opinion Pool Applied in Classification of Anderson River Data: Test Samples. Topographic Sources were Modeled by Maximum Penalized Likelihood Method.

	Percent Agreement with Reference for Class							
	1	2	3	4	5	6	OA	AVE
				Single Sources				
ABMSS	12.4	4.1	81.0	80.5	49.5	67.1	48.34	49.10
SAR sh	44.4	1.9	13.1	7.9	0.0	77.6	36.62	24.15
SAR st	34.2	0.9	3.8	13.4	0.7	85.5	36.02	23.07
Elevation	15.6	14.4	46.8	47.8	47.1	57.3	38.64	38.18
Slope	30.9	0.4	6.9	2.0	50.9	65.5	31.94	26.11
Aspect	35.9	19.9	43.1	15.2	11.7	50.2	34.54	29.34
m h t e s a				Multiple Sources				
1. 1. 1. 1. 1. 1.	51.3	0.0	0.0	48.9	0.0	95.7	47.00	32.65
1. .9 .9 1. 1. 1.	52.0	0.0	0.0	50.8	0.0	95.5	47.36	33.06
1. .8 .8 1. 1. 1.	53.0	0.0	0.0	53.6	0.0	95.2	47.86	33.64
1. .7 .7 1. 1. 1.	54.3	0.0	0.0	56.8	0.1	94.9	48.49	34.35
1. .6 .6 1. 1. 1.	55.3	0.0	0.0	60.2	0.3	94.7	49.12	35.09
1. .5 .5 1. 1. 1.	57.0	0.0	0.0	63.4	0.8	94.2	49.80	35.90
1. .4 .4 1. 1. 1.	58.9	0.0	0.0	66.8	2.0	93.5	50.56	36.86
1. .3 .3 1. 1. 1.	61.4	0.0	0.0	69.7	4.2	92.7	51.44	38.00
1. .2 .2 1. 1. 1.	63.6	0.0	0.0	71.6	7.7	91.6	52.14	39.09
1. .1 .1 1. 1. 1.	65.9	0.0	0.0	73.2	11.8	90.2	52.78	40.19
1. .0 .0 1. 1. 1.	68.0	0.0	0.0	74.5	16.0	88.6	53.26	41.18
1. 1. 1. .9 .9 .9	53.0	0.0	0.0	51.2	0.0	95.6	47.67	33.31
1. 1. 1. .8 .8 .8	54.8	0.0	0.0	55.1	0.0	95.3	48.48	34.19
1. 1. 1. .7 .7 .7	56.7	0.0	0.0	58.5	0.0	94.8	49.25	35.02
1. 1. 1. .6 .6 .6	58.1	0.0	0.0	62.4	0.4	94.4	49.95	35.87
1. 1. 1. .5 .5 .5	59.8	0.0	0.0	65.3	2.0	93.8	50.69	36.83
1. 1. 1. .4 .4 .4	61.7	0.0	0.0	67.6	4.9	93.1	51.43	37.88
1. 1. 1. .3 .3 .3	63.7	0.0	0.0	69.6	8.5	92.3	52.18	39.01
1. 1. 1. .2 .2 .2	65.8	0.0	0.0	71.3	12.7	91.4	52.92	40.19
1. 1. 1. .1 .1 .1	67.4	0.0	0.0	72.6	16.6	90.3	53.44	41.15
1. 1. 1. .0 .0 .0	68.9	0.0	0.0	73.1	20.8	89.3	53.89	42.03
1. 1. 1. .0 1. 1.	57.0	0.0	0.0	26.1	0.0	95.1	45.19	29.70
1. .9 .9 .9 .9 .9	54.0	0.0	0.0	54.5	0.0	95.3	48.22	33.96
1. .8 .8 .9 .9 .9	55.1	0.0	0.0	57.7	0.1	94.9	48.79	34.63
1. .8 .8 1. .9 .9	54.2	0.0	0.0	57.0	0.0	95.0	48.52	34.37
1. .8 .8 1. .9 1.	53.8	0.0	0.0	55.4	0.0	95.1	48.23	34.04
1. .8 .8 1. .8 1.	54.5	0.0	0.0	57.2	0.1	94.9	48.57	34.44
1. .8 .8 .8 .8 .8	57.6	0.0	0.0	61.4	0.4	94.5	49.74	35.65
# of pixels	8744	4960	4932	4881	2856	11340	37713	37713

The columns labeled m h t e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

The results of the single-source Parzen density classifications are shown in Tables 4.55 and 4.56. By looking at the training results (Table 4.55) and comparing them to the results for the other density estimation methods (Tables 4.45 and 4.51) it is seen that the Parzen density estimation does not perform as well in classification accuracy of training data (similar to the Colorado experiment). In contrast the test results (Table 4.56) using the Parzen density method outperformed the histogram (Table 4.46) and the maximum penalized likelihood estimates (Table 4.52). For example for the aspect data the Parzen density estimation improved the overall accuracy of test data by just under 2.0% as compared to the other methods. Two percent increase in accuracy for these data is noteworthy

The reliability measure based on the overall classification accuracy ranked the data sources in the same order as it had for the other density estimation methods. However, looking at the equivocations in Tables 4.47 and 4.48 it can be seen that the equivocation ranked the sources in the following manner: 1) ABMSS, 2) elevation, 3) aspect, 4) SAR sh, 5) slope and 6) SAR st. The poor classification accuracy of training data with Parzen density estimation moved the slope data down one spot in the ranking; the overall classification accuracy measure still ranked the slope data as the worst data source.

The results using SMC are also shown in Tables 4.55 (training) and 4.56 (test). The training results showed that when all the data sources were given equal weights, overall accuracy of 69.32% and average accuracy of 68.62% were achieved. Both of these accuracies were lower than the ones reached with the other density estimation methods. The overall accuracy was

Table 4.55

Statistical Multisource Classification of Anderson River Data: Training Samples. Topographic Sources were Modeled by Parzen Density Estimation.

	Percent Agreement with Reference for Class							
	1	2	3	4	5	6	OA	AVE
	Single Sources							
ABMSS	13.3	4.5	83.6	84.7	48.6	68.5	49.84	50.53
SAR sh	45.0	2.4	12.0	7.6	0.0	78.2	36.81	24.19
SAR st	35.5	1.3	4.0	12.9	1.3	86.0	36.57	23.50
Elevation	13.2	14.5	50.7	48.0	48.6	61.0	39.84	39.33
Slope	36.4	0.4	0.0	1.5	42.3	71.8	33.47	25.40
Aspect	39.8	23.8	46.4	19.6	6.0	54.0	37.65	31.60
m h t e s a	Multiple Sources							
1. 1. 1. 1. 1. 1.	68.7	34.3	77.4	79.5	76.3	75.5	69.32	68.62
1. .9 .9 1. 1. 1.	68.9	32.8	77.9	79.5	76.3	75.6	69.28	68.51
1. .8 .8 1. 1. 1.	68.9	33.2	78.5	79.5	75.7	75.8	69.42	68.60
1. .7 .7 1. 1. 1.	68.9	33.0	78.8	79.3	75.7	75.7	69.40	68.59
1. .6 .6 1. 1. 1.	68.6	31.9	78.8	79.2	75.4	75.6	69.09	68.24
1. .5 .5 1. 1. 1.	68.9	31.8	78.6	79.3	75.4	75.6	69.16	68.28
1. .4 .4 1. 1. 1.	68.6	31.2	78.6	79.5	75.1	75.6	69.01	68.11
1. .3 .3 1. 1. 1.	69.4	30.5	80.1	79.0	75.1	75.5	69.18	68.26
1. .2 .2 1. 1. 1.	69.0	29.2	79.4	79.0	74.8	75.1	68.68	67.73
1. .1 .1 1. 1. 1.	69.2	28.7	79.4	78.8	74.4	74.8	68.54	67.56
1. .0 .0 1. 1. 1.	69.0	28.9	79.7	78.8	74.1	74.8	68.51	67.55
1. 1. 1. .9 .9 .9	69.1	33.2	76.1	79.5	72.2	75.9	68.92	66.76
1. 1. 1. .8 .8 .8	69.9	31.6	75.0	79.3	68.1	76.0	68.44	66.66
1. 1. 1. .7 .7 .7	70.6	29.9	74.8	79.2	63.1	76.2	68.04	65.64
1. 1. 1. .6 .6 .6	71.4	26.3	74.6	78.8	59.6	76.1	67.37	64.47
1. 1. 1. .5 .5 .5	72.2	22.3	73.9	78.8	56.5	76.7	66.75	63.32
1. 1. 1. .4 .4 .4	72.4	19.1	72.3	78.8	54.3	76.3	65.98	62.17
1. 1. 1. .3 .3 .3	73.0	14.2	68.1	79.2	51.1	76.3	64.74	60.29
1. 1. 1. .2 .2 .2	72.3	9.1	65.0	79.0	47.9	76.4	63.28	58.28
1. 1. 1. .1 .1 .1	72.1	5.1	61.1	78.2	44.5	76.7	61.92	56.28
1. 1. 1. .0 .0 .0	70.2	3.6	55.5	77.7	42.0	76.5	60.25	54.25
1. 1. 1. .0 1. 1.	72.1	13.8	75.5	77.7	54.6	75.8	65.39	61.58
1. .9 .9 .9 .9 .9	69.9	32.3	76.1	79.5	72.9	75.8	69.01	67.75
1. .8 .8 .9 .9 .9	69.7	32.1	77.2	79.5	72.9	76.0	69.13	67.90
1. .8 .8 1. .9 .9	69.7	32.1	77.2	79.5	72.9	76.0	69.13	67.90
1. .8 .8 1. .9 1.	68.8	33.2	78.3	79.9	75.4	75.8	69.40	68.56
1. .8 .8 1. .8 1.	68.9	33.6	78.1	79.7	73.5	75.8	69.28	68.26
1. .8 .8 .8 .8 .8	71.2	31.2	76.1	79.2	67.5	76.0	68.78	66.86
# of pixels	971	551	548	542	317	1260	4189	4189

The columns labeled m h t e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

CPU time for training and classification: 8479 sec.

Table 4.56

Statistical Multisource Classification of Anderson River Data: Test Samples. Topographic Sources were Modeled by Parzen Density Estimation.

	Percent Agreement with Reference for Class							
	1	2	3	4	5	6	OA	AVE
	Single Sources							
ABMSS	12.4	4.1	81.0	80.5	49.5	67.1	48.34	49.10
SAR sh	44.4	1.9	13.1	7.9	0.0	77.6	36.62	24.15
SAR st	34.2	0.9	3.8	13.4	0.7	85.5	36.02	23.07
Elevation	12.0	13.3	51.1	47.3	47.1	59.6	38.82	38.40
Slope	35.5	0.3	0.0	0.9	42.7	70.3	32.76	24.95
Aspect	39.5	23.3	43.7	17.6	3.2	52.4	36.19	29.95
m h t e s a	Multiple Sources							
1. 1. 1. 1. 1. 1.	68.9	32.4	75.5	78.5	75.6	74.9	68.51	67.63
1. .9 .9 1. 1. 1.	69.0	32.0	75.9	78.7	75.5	75.0	68.57	67.67
1. .8 .8 1. 1. 1.	69.0	31.8	75.9	78.6	75.5	75.1	68.58	67.65
1. .7 .7 1. 1. 1.	68.7	31.6	75.9	78.6	75.6	75.1	68.51	67.60
1. .6 .6 1. 1. 1.	68.7	31.2	76.1	78.5	75.9	75.1	68.47	67.59
1. .5 .5 1. 1. 1.	68.6	30.9	76.3	78.6	75.8	75.0	68.41	67.54
1. .4 .4 1. 1. 1.	68.6	30.2	76.2	78.5	75.4	74.9	68.24	67.31
1. .3 .3 1. 1. 1.	68.5	29.5	76.2	78.6	75.4	74.8	68.09	67.16
1. .2 .2 1. 1. 1.	68.1	28.7	76.0	78.7	75.3	74.6	67.84	66.91
1. .1 .1 1. 1. 1.	68.2	28.2	75.7	78.7	75.1	74.5	67.68	66.73
1. .0 .0 1. 1. 1.	67.8	27.5	75.1	78.6	75.1	74.3	67.36	66.40
1. 1. 1. .9 .9 .9	69.3	30.7	74.3	78.8	72.3	75.1	68.10	66.76
1. 1. 1. .8 .8 .8	69.6	29.0	73.4	78.8	68.3	75.4	67.57	65.73
1. 1. 1. .7 .7 .7	69.9	26.7	72.4	78.8	64.8	75.7	67.05	64.71
1. 1. 1. .6 .6 .6	70.3	24.2	70.7	78.6	60.8	75.9	66.32	63.41
1. 1. 1. .5 .5 .5	70.8	21.2	67.1	78.5	57.1	76.3	65.62	62.13
1. 1. 1. .4 .4 .4	71.3	16.7	67.2	78.3	53.9	76.4	64.72	60.64
1. 1. 1. .3 .3 .3	72.0	12.3	64.6	78.1	50.2	76.6	63.69	58.96
1. 1. 1. .2 .2 .2	72.8	7.9	61.9	77.8	46.8	76.8	62.71	57.32
1. 1. 1. .1 .1 .1	72.4	4.2	57.9	77.4	44.2	76.8	61.37	55.49
1. 1. 1. .0 .0 .0	70.8	1.9	52.9	77.1	41.6	76.9	59.83	53.52
1. 1. 1. .0 1. 1.	72.0	11.5	72.2	76.8	54.6	75.5	64.42	60.42
1. .9 .9 .9 .9 .9	69.4	30.4	74.6	78.9	72.2	75.3	68.18	66.80
1. .8 .8 .9 .9 .9	69.4	30.0	75.1	79.0	72.0	75.3	68.16	66.77
1. .8 .8 1. .9 .9	69.1	31.5	75.1	78.9	74.1	75.3	68.42	67.31
1. .8 .8 1. .9 1.	68.8	31.6	75.6	78.8	74.2	75.1	68.41	67.37
1. .8 .8 1. .8 1.	68.7	31.6	75.5	79.0	72.7	75.1	68.27	67.09
1. .8 .8 .8 .8 .8	70.0	27.9	73.5	78.8	68.1	75.6	67.58	65.64
# of pixels	8744	4960	4932	4881	2856	11340	37713	37713

The columns labeled m h t e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

increased slightly in Table 4.55 by lowering the weights on the SAR sources to 0.8 and keeping the weights of the other sources at 1. The highest overall accuracy (69.42%) was still lower than the "best" results for training data achieved by the other density estimation methods. The reason for this low accuracy was clearly that the Parzen estimation was poorer in classifying the training data than the other methods. Looking at the test results in Table 4.56 it can be seen that the Parzen density estimation gave the highest overall and average accuracies of test data. When the sources were combined with equal weights, the overall accuracy was improved to 68.51% (histogram: 68.19%, maximum penalized likelihood method: 68.20%) and the average accuracy was increased to 67.63% (histogram: 67.59%, maximum penalized likelihood method: 67.48%). When the weights of the SAR data sources were decreased to 0.8, without changing the weights of the other sources, the overall and average accuracies both improved slightly (OA: 68.58%, AVE: 67.65%) as compared to the equal weights result. This overall accuracy was the highest test result achieved in the all the SMC experiments for the Anderson River data. Therefore, it can be concluded from these results that the SMC generalizes well when Parzen density estimation is used to model the non-Gaussian data sources.

The results using the LOP with Parzen density estimation are shown in Tables 4.57 (training) and 4.58 (test). As a consequence of the poor training performance by the Parzen density estimation, the training accuracies using the LOP in Table 4.57 were worse than those obtained with the other density estimation methods. In contrast the test accuracies using Parzen density estimation were slightly better than the ones with the other methods.

Table 4.57

Linear Opinion Pool Applied in Classification of Anderson River Data: Training Samples. Topographic Sources were Modeled by Parzen Density Estimation.

	Percent Agreement with Reference for Class							
	1	2	3	4	5	6	OA	AVE
				Single Sources				
ABMSS	13.3	4.5	83.6	84.7	48.6	68.5	49.84	50.53
SAR sh	45.0	2.4	12.0	7.6	0.0	78.2	36.81	24.19
SAR st	35.5	1.3	4.0	12.9	1.3	86.0	36.57	23.50
Elevation	13.2	14.5	50.7	48.0	48.6	61.0	39.84	39.33
Slope	36.4	0.4	0.0	1.5	42.3	71.8	33.47	25.40
Aspect	39.8	23.8	46.4	19.6	6.0	54.0	37.65	31.60
m h t e s a				Multiple Sources				
1. 1. 1. 1. 1. 1.	53.1	0.0	0.0	49.8	0.0	95.9	47.60	33.14
1. .9 .9 1. 1. 1.	54.1	0.0	0.0	52.0	0.0	95.8	48.08	33.65
1. .8 .8 1. 1. 1.	54.9	0.0	0.0	57.4	0.0	95.6	48.89	34.64
1. .7 .7 1. 1. 1.	55.8	0.0	0.0	60.3	0.0	95.2	49.37	35.22
1. .6 .6 1. 1. 1.	57.3	0.0	0.0	64.8	0.0	95.2	50.27	36.20
1. .5 .5 1. 1. 1.	58.8	0.0	0.0	67.9	0.3	94.7	50.92	36.95
1. .4 .4 1. 1. 1.	60.8	0.0	0.0	71.6	0.9	94.0	51.71	37.89
1. .3 .3 1. 1. 1.	63.0	0.0	0.0	72.9	5.0	93.5	52.54	39.07
1. .2 .2 1. 1. 1.	65.3	0.0	0.0	74.0	9.8	92.6	53.31	40.28
1. .1 .1 1. 1. 1.	67.6	0.2	0.0	75.1	13.9	91.6	54.00	41.38
1. .0 .0 1. 1. 1.	68.7	0.2	0.0	76.6	17.7	89.9	54.24	42.17
1. 1. 1. .9 .9 .9	54.2	0.0	0.0	54.2	0.0	95.2	48.22	33.94
1. 1. 1. .8 .8 .8	56.0	0.0	0.0	60.0	0.0	94.8	49.25	35.12
1. 1. 1. .7 .7 .7	57.8	0.0	0.0	62.9	0.0	94.3	49.89	35.83
1. 1. 1. .6 .6 .6	58.7	0.0	0.0	65.9	0.0	93.9	50.37	36.41
1. 1. 1. .5 .5 .5	60.2	0.0	0.0	68.5	1.6	93.3	51.01	37.27
1. 1. 1. .4 .4 .4	62.9	0.0	0.0	70.1	7.9	92.8	52.16	38.95
1. 1. 1. .3 .3 .3	64.8	0.0	0.0	71.0	13.9	92.5	53.06	40.36
1. 1. 1. .2 .2 .2	66.0	0.0	0.0	72.9	18.6	91.7	53.74	41.54
1. 1. 1. .1 .1 .1	67.3	0.0	0.0	73.1	22.1	90.6	53.97	42.17
1. 1. 1. .0 .0 .0	68.2	0.0	0.0	73.1	24.3	89.4	54.00	42.50
1. 1. 1. .0 1. 1.	58.2	0.0	0.0	30.8	0.0	94.7	45.95	30.61
1. .9 .9 .9 .9 .9	55.3	0.0	0.0	58.9	0.0	95.2	49.08	34.90
1. .8 .8 .9 .9 .9	57.2	0.0	0.0	62.2	0.0	94.6	49.75	35.66
1. .8 .8 1. .9 .9	55.7	0.0	0.0	61.3	0.0	95.0	49.42	35.33
1. .8 .8 1. .9 1.	55.6	0.0	0.0	59.0	0.0	95.5	49.25	35.02
1. .8 .8 1. .8 1.	56.2	0.0	0.0	61.4	0.0	95.0	49.56	35.44
1. .8 .8 .8 .8 .8	58.2	0.0	0.0	66.1	0.0	94.0	50.30	36.37
# of pixels	971	551	548	542	317	1260	4189	4189

The columns labeled m h t e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

CPU time for training and classification: 8453 sec.

Table 4.58

Linear Opinion Pool Applied in Classification of Anderson River Data: Test Samples. Topographic Sources were Modeled by Parzen Density Estimation.

	Percent Agreement with Reference for Class							
	1	2	3	4	5	6	OA	AVE
	Single Sources							
ABMSS	12.4	4.1	81.0	80.5	49.5	67.1	48.34	49.10
SAR sh	44.4	1.9	13.1	7.9	0.0	77.6	36.62	24.15
SAR st	34.2	0.9	3.8	13.4	0.7	85.5	36.02	23.07
Elevation	12.0	13.3	51.1	47.3	47.1	59.6	38.82	38.40
Slope	35.5	0.3	0.0	0.9	42.7	70.3	32.76	24.95
Aspect	39.5	23.3	43.7	17.6	3.2	52.4	36.19	29.95
m h t e s a	Multiple Sources							
1. 1. 1. 1. 1. 1.	52.4	0.0	0.0	49.6	0.0	95.9	47.39	32.98
1. .9 .9 1. 1. 1.	53.1	0.0	0.0	51.8	0.0	95.6	47.77	33.42
1. .8 .8 1. 1. 1.	54.0	0.0	0.0	54.3	0.0	95.4	48.25	33.96
1. .7 .7 1. 1. 1.	55.2	0.0	0.0	57.3	0.0	95.2	48.84	34.62
1. .6 .6 1. 1. 1.	56.5	0.0	0.0	60.7	0.0	95.0	49.50	35.36
1. .5 .5 1. 1. 1.	57.8	0.0	0.0	64.5	0.2	94.4	50.18	36.18
1. .4 .4 1. 1. 1.	59.7	0.0	0.0	67.7	1.0	94.0	50.95	37.07
1. .3 .3 1. 1. 1.	61.9	0.0	0.0	70.1	2.8	93.3	51.67	38.00
1. .2 .2 1. 1. 1.	64.1	0.0	0.0	72.2	5.9	92.2	52.37	39.07
1. .1 .1 1. 1. 1.	66.3	0.1	0.0	73.9	9.8	90.7	52.96	40.13
1. .0 .0 1. 1. 1.	68.1	0.2	0.0	75.0	14.3	88.9	53.34	41.09
1. 1. 1. .9 .9 .9	54.1	0.0	0.0	52.5	0.0	95.7	48.11	33.71
1. 1. 1. .8 .8 .8	55.8	0.0	0.0	56.2	0.0	95.2	48.85	34.54
1. 1. 1. .7 .7 .7	57.4	0.0	0.0	60.0	0.0	94.8	49.57	35.37
1. 1. 1. .6 .6 .6	58.8	0.0	0.0	63.1	0.1	94.2	50.14	36.05
1. 1. 1. .5 .5 .5	60.5	0.0	0.0	66.0	1.8	93.6	50.87	36.99
1. 1. 1. .4 .4 .4	62.6	0.0	0.0	68.0	4.7	93.0	51.60	38.03
1. 1. 1. .3 .3 .3	64.6	0.0	0.0	69.5	8.5	92.1	52.33	39.14
1. 1. 1. .2 .2 .2	66.5	0.0	0.0	71.3	12.6	91.3	53.06	40.29
1. 1. 1. .1 .1 .1	68.1	0.0	0.0	72.3	16.7	90.4	53.59	41.25
1. 1. 1. .0 .0 .0	68.9	0.0	0.0	73.1	20.8	89.3	53.89	42.03
1. 1. 1. .0 1. 1.	57.6	0.0	0.0	28.6	0.0	95.1	45.66	30.22
1. .9 .9 .9 .9 .9	55.1	0.0	0.0	55.4	0.0	95.4	48.64	34.32
1. .8 .8 .9 .9 .9	56.0	0.0	0.0	58.3	0.0	95.1	49.13	34.91
1. .8 .8 1. .9 .9	55.4	0.0	0.0	57.6	0.0	95.2	48.90	34.68
1. .8 .8 1. .9 1.	54.9	0.0	0.0	56.3	0.0	95.3	48.67	34.41
1. .8 .8 1. .8 1.	55.6	0.0	0.0	58.1	0.0	95.1	49.01	34.80
1. .8 .8 .8 .8 .8	58.2	0.0	0.0	62.8	0.1	94.7	50.08	35.95
# of pixels	8744	4960	4932	4881	2856	11340	37713	37713

The columns labeled m h t e s a indicate the weights applied to the sources (in the same order as the single source classifications above).

However, the highest overall and average accuracies of test data were reached when the topographic sources were discarded, exactly the same result as for the other density estimation methods.

d) General Comments on the Statistical Methods

The SMC was clearly the best statistical method used. The LOP, on the other hand, did not perform well at all. The three density estimation methods showed different characteristics. The histogram was the best method in terms of classification accuracy of training data. The maximum penalized likelihood method and the Parzen density estimation showed better performance in classification accuracy of test data. The Parzen density estimation gave the best overall classification accuracy of test data for the combined sources. However, the Parzen density estimation was computationally more intensive than the other density estimation methods as seen in Table 4.60. It took fifteen times longer to train and classify the data using this method as compared to the maximum penalized likelihood method and 1347 times longer as compared to the histogram method. The maximum penalized likelihood method and the Parzen density estimation were equally fast for the Colorado data, but for the Colorado data the test data size was smaller. Here the test pixels were 37713 as compared to only 1011 for the Colorado data. The computational complexity of the Parzen estimator is a shortcoming to be taken into account.

The SMC method was faster than the ML classifier when either the histogram or the maximum penalized likelihood methods were used for density estimation. The SMC also outperformed the ML in terms of classification

Table 4.59

Source-Specific CPU Times (in Sec) for Training Plus Classification of Gaussian Data Sources.

Sensor	ABMSS	SAR Shallow	SAR Steep
# of channels	11	4	4
CPU time	198	42	42

Table 4.60

Source-Specific CPU Times (in Sec) for Training Plus Classification of Non-Gaussian Data Sources with Regard to Different Modeling Methods.

Method	Histogram Estimation	Maximum Penalized Likelihood Method	Parzen Estimation
CPU time	2	176	2694

accuracy. However, the reliability factor mechanism did not help much in the SMC classification of the Anderson River data except when the Parzen density estimator was used. The reasons why the results could not be improved for the other density estimators with different weighting are unclear. The Anderson River data are very hard to classify accurately and the classifiers might need all the information they can get.

The LOP method showed very poor performance both in terms of overall and average accuracies. The LOP was seen to be of very questionable value as a multisource classification tool. As stated in Chapter 2, the LOP has in general more tendency to result in multimodal distribution than a logarithmic opinion pool (SMC). Because of the multimodality of the LOP it needs agreeable sources to perform well, i.e., sources which tend to make the same source-specific decisions for most of the input data. The sources used in the multisource classification of both the Anderson River and the Colorado data cannot be considered agreeable.

4.3.2 Results: Neural Network Methods

The CGLC and CGBP were trained with Gray-coded input data. The Anderson River data has 22 data channels. Each channel was coded with 8 bits and therefore, 176 (or 8*22) input neurons were used for both networks. The data were trained on the 9 data classes discussed in the beginning of Section 4.3. Therefore, 9 output neurons were selected. The convergence criterion for the training procedures was selected the same as in the Colorado experiments (gradient of the error function has to be less than 0.0001 for the procedure to converge).

The results using the CGLC are shown in Tables 4.61 (training) and 4.62 (test). After 295 iterations, the training procedure was stopped since the error function did not decrease further. The highest overall accuracy of training data was achieved after 250 iterations (OA: 73.55%, Ave: 72.48%). These results were significantly better than the ones reached by the statistical methods. The best test result using the CGLC was achieved after 295 iterations. There the CGLC gave overall accuracy of 67.88% for test data and 66.48% average accuracy. The SMC with all density estimation techniques achieved better results for test data (histogram method; OA: 68.13%, AVE: 67.39%).

The CGBP was tested extensively with three layers of neurons since adding more layers did not improve the classification accuracy. The CGBP was implemented with 25 hidden neurons. Adding more hidden neurons did not increase the classification accuracy. The results of the CGBP experiment are shown in Tables 4.63 (training) and 4.64 (test). The CGBP showed excellent performance in classification of training data. When the training procedure stopped (the error function did not decrease further) after 1417 iterations, the overall accuracy had reached 99.47% and the average accuracy 99.43%. Obviously the CGBP outperformed all the other methods in classification of training data. However, the CGBP did not do much better in testing than the CGLC. The highest accuracies of test data were reached after only 200 iterations (OA: 67.95%, AVE: 66.60%). These accuracies were lower than the ones achieved by the SMC method with any of the three density estimation approaches. After 200 iterations, the test performance of the CGBP fell off significantly. The test accuracies continued to decrease until the

Table 4.61

Conjugate Gradient Linear Classifier Applied in Classification of the Anderson River Data Set: Training Samples.

Number of	CPU	Percent Agreement with Reference for Class							
iterations	time	1	2	3	4	5	6	OA	AVE
50	1447	65.6	45.6	71.0	70.7	78.9	82.6	70.45	69.07
100	2209	70.2	43.6	72.6	73.1	83.9	84.0	72.57	71.23
150	2923	69.7	44.5	72.3	73.6	85.5	84.6	72.91	71.70
200	3787	69.4	43.4	72.1	74.4	88.0	85.5	73.22	72.13
250	4488	70.1	45.2	72.3	74.0	88.0	85.3	73.55	72.48
295	5129	69.4	45.2	72.3	74.5	87.7	85.6	73.50	72.45
# of pixels		971	551	548	542	317	1260	4189	4189

Table 4.62

Conjugate Gradient Linear Classifier Applied in Classification of the Anderson River Data Set: Test Samples.

Number of	Percent Agreement with Reference for Class							
iterations	1	2	3	4	5	6	OA	AVE
50	61.5	36.9	67.0	67.8	72.5	79.9	66.17	64.27
100	63.8	35.5	69.2	68.4	79.2	81.3	67.82	66.23
150	63.4	37.1	68.8	68.1	80.0	81.0	67.80	66.40
200	63.4	38.0	68.8	68.2	79.9	80.4	67.87	66.45
250	63.5	37.9	68.5	67.9	79.9	80.7	67.74	66.40
295	63.5	38.2	68.7	68.1	79.8	80.7	67.88	66.48
# of pixels	8744	4960	4932	4881	2856	11340	37713	37713

Table 4.63

Conjugate Gradient Backpropagation Applied in Classification of the Anderson River Data Set: Training Samples.

Number of	CPU	Percent Agreement with Reference for Class							
iterations	time	1	2	3	4	5	6	OA	AVE
50	3780	58.1	34.1	67.9	64.0	76.0	83.4	65.96	63.92
100	6173	68.0	45.6	71.4	70.7	83.3	83.9	71.76	70.48
150	8607	73.7	47.2	77.0	73.2	85.5	86.0	75.17	73.77
200	10941	76.2	57.4	79.9	73.4	89.3	88.8	78.63	77.50
250	13401	82.2	69.3	82.5	76.8	90.9	90.4	82.96	82.02
300	15554	85.4	76.2	86.3	80.8	93.4	91.9	86.27	85.67
350	19625	89.9	82.0	90.1	81.0	95.6	94.0	89.40	88.77
400	20435	93.4	84.6	93.4	82.7	95.9	94.8	91.45	90.80
600	29767	98.4	95.1	99.5	87.1	99.7	98.1	96.66	96.32
900	44296	99.7	99.3	99.8	90.6	99.7	99.5	98.42	98.10
1200	58623	99.7	99.6	100.0	97.4	100.0	99.7	99.45	99.40
1417	68951	99.7	99.6	100.0	97.6	100.0	99.7	99.47	99.43
# of pixels		971	551	548	542	317	1260	4189	4189

Table 4.64

Conjugate Gradient Backpropagation Applied in Classification of the Anderson River Data Set: Test Samples.

Number of	Percent Agreement with Reference for Class							
iterations	1	2	3	4	5	6	OA	AVE
50	55.8	27.6	64.5	61.1	74.6	81.3	63.03	60.82
100	62.3	35.8	67.5	67.7	79.0	81.4	67.20	65.62
150	64.5	35.6	69.2	68.4	77.0	81.1	67.74	65.97
200	62.7	43.6	67.3	68.1	77.5	80.4	67.95	66.60
250	62.8	45.5	64.6	65.3	74.6	79.3	66.95	65.53
300	62.2	44.5	62.5	64.8	71.9	78.7	65.95	64.10
350	61.0	43.8	62.3	63.0	71.8	77.6	64.96	63.25
400	61.8	42.4	61.8	62.3	69.2	77.3	64.56	62.47
600	57.6	38.3	57.3	61.2	65.1	73.8	60.93	58.88
900	55.2	36.8	55.0	63.3	61.7	69.9	58.73	56.98
1200	53.5	35.8	54.1	63.7	62.2	69.4	58.00	56.45
1417	53.8	35.4	54.3	63.6	62.0	69.3	58.00	56.40
# of pixels	8744	4960	4932	4881	2856	11340	37713	37713

training procedure was stopped. After 1417 iterations the overall accuracy of test data was only 58.00% and the average accuracy only 56.40%. Obviously the training procedure of the CGBP had the problem of overtraining. When it stopped it usually showed excellent performance for training data, but it did not do well for test data. To do well for test data it has to be stopped earlier. When to stop the training is a major problem. In this regard, the CGLC was a better choice in the classification of Anderson River data. When the CGLC training procedure of the CGLC stopped, it produced results close to its best training *and* test results.

The training procedure for the CGBP was also more time consuming than for the CGLC. The hidden neurons were the obvious reason for this. After 200 iterations the CGBP had needed 10941 CPU sec and after 1417 iterations it had needed 68951 CPU sec. However, the CGLC needed 5129 CPU sec for 295 iterations. Also, the CGBP needed 1362 CPU sec in classification of the data but the CGLC needed 622 sec. In comparison, the SMC classified the data in only 107 CPU sec and was trained in 402 (histogram approach), 926 (maximum penalized likelihood method) or 8453 (Parzen density estimation) sec.

The best classification results in the experiment on Anderson River data are shown in Figure 4.9. Looking at this figure it is seen that the SMC achieved higher overall accuracy in classification of test data as compared to the neural networks although the neural networks achieved higher training accuracies. Thus, the SMC classifier outperformed the neural networks in this experiment both in terms of classification accuracy of test data and speed (excluding Parzen density estimation).

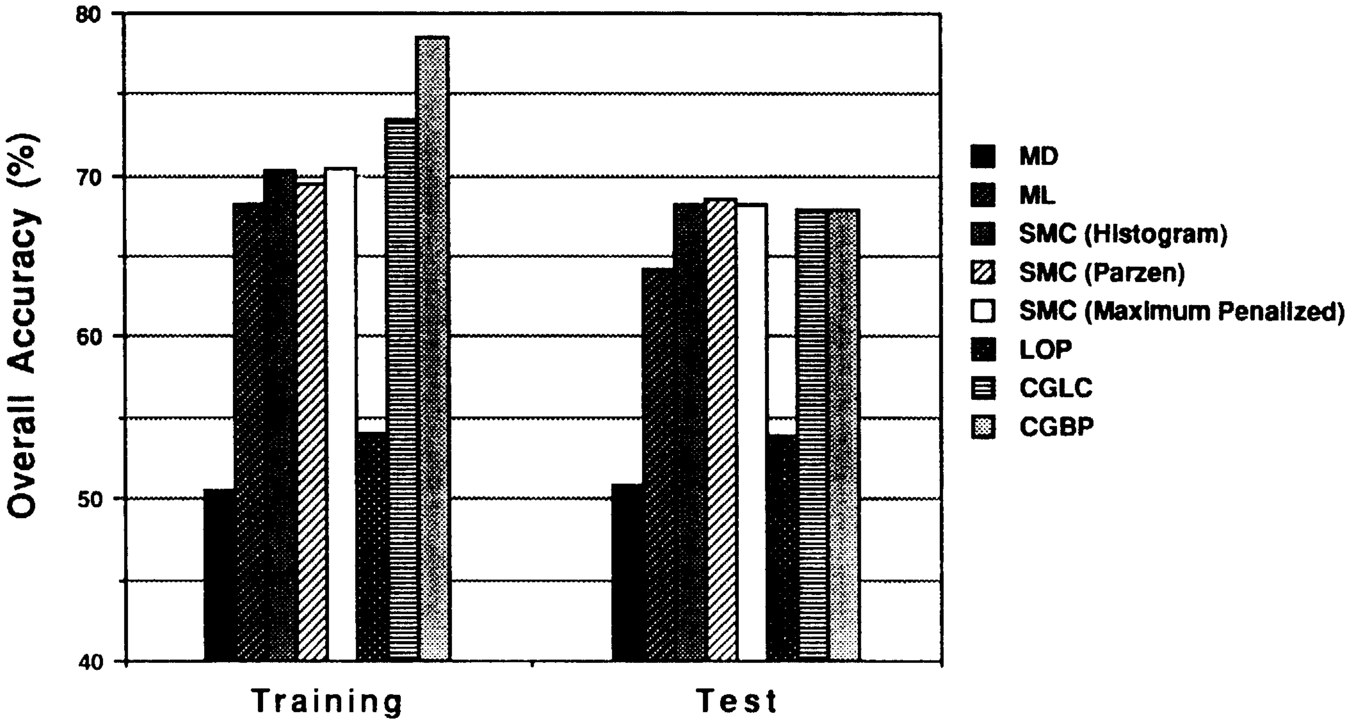


Figure 4.9 Summary of Best Classification Results for Experiment on Anderson River Data.

4.4 Experiments with Simulated HIRIS Data

This experiment investigated how well the statistical methods and the neural network models perform as classifiers of very-high-dimensional data (data that have many features, possibly hundreds of them). In these experiments the very-high-dimensional data were simulated High Resolution Imaging Spectrometer (HIRIS) data. The HIRIS instrument is planned to be a part of a cluster of scientific instruments forming the Earth Observing System (EOS). A simulation program called RSSIM [84] was used to simulate the data.

The simulated data used in the experiments were Gaussian distributed, which is one of the reasons why multivariate statistical approaches are used for the classification. However, a problem with using conventional multivariate statistical approaches for classification of multidimensional data is that these methods rely on having nonsingular (invertible) class-specific covariance matrices. When n features are used, the training samples for each class need to include at least n+1 *different* samples so that the matrices are nonsingular. Therefore, the covariance matrices may be singular in high-dimensional cases involving limited training samples.

The RSSIM simulation program generated 201 spectral bands of HIRIS data. The HIRIS data were simulated based on statistics from Earth surface reflectance measurements from a site in Finney County, Kansas, on May 3, 1977. A total of 1551 observations were combined from three information classes: winter wheat, summer fallow, and an "unknown" class. Each class consisted of 675 samples. The information classes were assumed to be Gaussian distributed.

For these experiments, three feature sets (20-, 40- and 60-dimensional) were extracted from the 201 data channels. Each feature set consisted of data channels uniformly spaced over the HIRIS spectral range (0.4 μm to 2.4 μm) excluding the water absorption bands. Also, the 20-dimensional data set was selected as a subset of the 40-dimensional data set and the 40 dimensional data set was selected as a subset of the 60-dimensional data set. Thus the higher-dimensional data sets added features to the 20-dimensional data set.

Experiments were conducted using both the statistical algorithms (MD, ML, SMC and LOP) and the neural network methods (CGBP and CGLC). To see how sample size affected the performance of all the algorithms, the experiments were conducted for 100, 200, 300, 400, 500 and 600 training samples per class. The sample size was in each case the same for all the classes. Therefore, for each classification the overall and average accuracies were identical.

4.4.1 20-Dimensional Data

The JM distance separabilities (maximum of 1.41421) for the 20-dimensional data are shown in Table 4.65. The data were relatively separable according to the average JM distance separability. However, classes 2 (summer fallow) and 3 (unknown) were not as distinguishable from each other as both of them were from class 1 (winter wheat).

The results of the experiments with the 20-dimensional data are shown in Tables 4.66 (MD training), 4.67 (MD test), 4.68 (ML training), 4.69 (ML test), 4.70 (CGBP training), 4.71 (CGBP test), 4.72 (CGLC training) and 4.73 (CGLC test). The results are also summarized in Figures 4.10 (training) and

Table 4.65

Pairwise JM Distances for the 20-Dimensional Simulated HIRIS Data.

Class #	2	3
1	1.40120	1.36444
2	-	1.07504
Average:	1.280277	

C.3

Table 4.66

Minimum Euclidean Distance Classifier Applied to 20-Dimensional Simulated HIRIS Data: Training Samples.

# of Training	CPU	Percent Agreement with Reference for Class			
Samples	Time	1	2	3	OA
100	2	85.0	48.0	54.0	62.33
200	2	84.5	48.5	54.0	62.33
300	2	85.7	50.0	58.7	64.78
400	2	84.8	54.0	59.5	66.08
500	2	86.0	51.2	61.8	66.33
600	2	85.2	49.5	58.8	64.50

Table 4.67

Minimum Euclidean Distance Classifier Applied to 20-Dimensional Simulated HIRIS Data: Test Samples.

# of Training	Percent Agreement with Reference for Class			
Samples	1	2	3	OA
100	83.7	47.1	59.1	63.30
200	83.6	46.9	60.0	63.51
300	84.0	51.2	56.8	64.00
400	80.7	44.7	70.2	65.21
500	74.3	46.3	68.0	62.86
600	81.3	58.7	46.7	62.22

Table 4.68

Maximum Likelihood Method for Gaussian Data Applied to 20-Dimensional Simulated HIRIS Data: Training Samples.

# of Training Samples	CPU Time	Percent Agreement with Reference for Class			
		1	2	3	OA
100	15	100.0	98.0	90.0	96.00
200	15	100.0	95.0	88.0	94.33
300	16	99.7	88.3	86.0	91.33
400	18	99.0	91.3	87.8	92.67
500	18	99.6	90.2	86.0	91.93
600	18	99.6	89.2	84.8	91.22

Table 4.69

Maximum Likelihood Method for Gaussian Data Applied to 20-Dimensional Simulated HIRIS Data: Test Samples.

# of Training Samples	Percent Agreement with Reference for Class			
	1	2	3	OA
100	94.8	62.8	74.3	77.28
200	95.2	65.1	71.6	77.26
300	97.9	86.1	82.4	88.80
400	96.7	81.5	78.9	85.70
500	96.6	81.7	81.7	86.67
600	94.7	88.0	80.0	87.56

Table 4.70

Conjugate Gradient Backpropagation Applied to 20-Dimensional Simulated HIRIS Data: Training Samples.

Sample size	Number of iterations	CPU time	Percent Agreement with Reference for Class			
			1	2	3	OA
100	118	357	100.0	100.0	100.0	100.00
200	168	871	100.0	100.0	100.0	100.00
300	195	1396	100.0	100.0	100.0	100.00
400	258	2451	100.0	100.0	100.0	100.00
500	324	3890	100.0	100.0	100.0	100.00
600	350	4922	100.0	100.0	100.0	100.00

Table 4.71

Conjugate Gradient Backpropagation Applied to 20-Dimensional Simulated HIRIS Data: Test Samples.

Sample size	Number of iterations	Percent Agreement with Reference for Class			
		1	2	3	OA
100	357	82.6	53.6	49.6	61.91
200	168	82.5	52.4	54.7	63.23
300	195	87.5	60.8	57.1	68.44
400	258	88.0	60.0	53.8	67.27
500	324	85.1	60.0	48.6	64.57
600	350	86.7	58.7	49.3	64.89

Table 4.72

Conjugate Gradient Linear Classifier Applied to 20-Dimensional Simulated HIRIS Data: Training Samples.

Sample size	Number of iterations	CPU time	Percent Agreement with Reference for Class			
			1	2	3	OA
100	309	190	100.0	100.0	100.0	100.00
200	516	533	100.0	92.0	92.5	94.83
300	431	431	100.0	82.7	81.3	88.00
400	442	821	99.5	82.8	79.0	87.08
500	226	542	98.6	79.8	75.2	84.53
600	507	1364	98.8	78.0	73.5	83.44

Table 4.73

Conjugate Gradient Linear Classifier Applied to 20-Dimensional Simulated HIRIS Data: Test Samples.

Sample size	Number of iterations	Percent Agreement with Reference for Class			
		1	2	3	OA
100	309	80.3	55.5	46.1	60.64
200	516	86.7	57.3	53.3	65.75
300	431	87.7	62.7	57.6	69.33
400	442	88.0	62.5	53.1	67.88
500	226	88.0	56.6	52.6	65.71
600	507	89.3	64.0	52.0	68.44

4.11 (test). The classification accuracy of the MD algorithm (Table 4.66 (training) and 4.67 (test)) was poor, and using a larger sample size did not improve its accuracy. However, the MD algorithm was extremely fast in classification.

The ML method (Table 4.68 (training) and 4.69 (test)) showed the best performance overall of all the methods. Larger sample size did help with this algorithm: the accuracy of the test data increased significantly when 300 or more samples per class were used for training compared to when fewer samples were used.

The 3-layer CGBP neural network (Tables 4.70 (training) and 4.71 (test)) was trained with Gray-coded binary input data (240 input neurons). Fifteen hidden neurons were used since the classification performance of the network did not improve with more hidden neurons. As in all the neural network experiments in this section, three output neurons were used (the number of classes). Also, all the neural networks were considered to have converged when the gradient of the error function was less than 0.0001. The neural networks converged in each case. The CGBP neural network was always trained to perfection for the 20-dimensional data regardless of sample size. However, for test data it did not do very well. Its overall test accuracy varied, but without a clear indication that the CGBP does better with a large training sample than with a smaller training set. For this method, the training time grew rapidly with sample size requiring 1.37 hours of CPU time for the largest sample size (272 times longer than for the ML method). Compared to the CGBP the ML method was very fast and its training time remained almost constant regardless of the size of the sample.

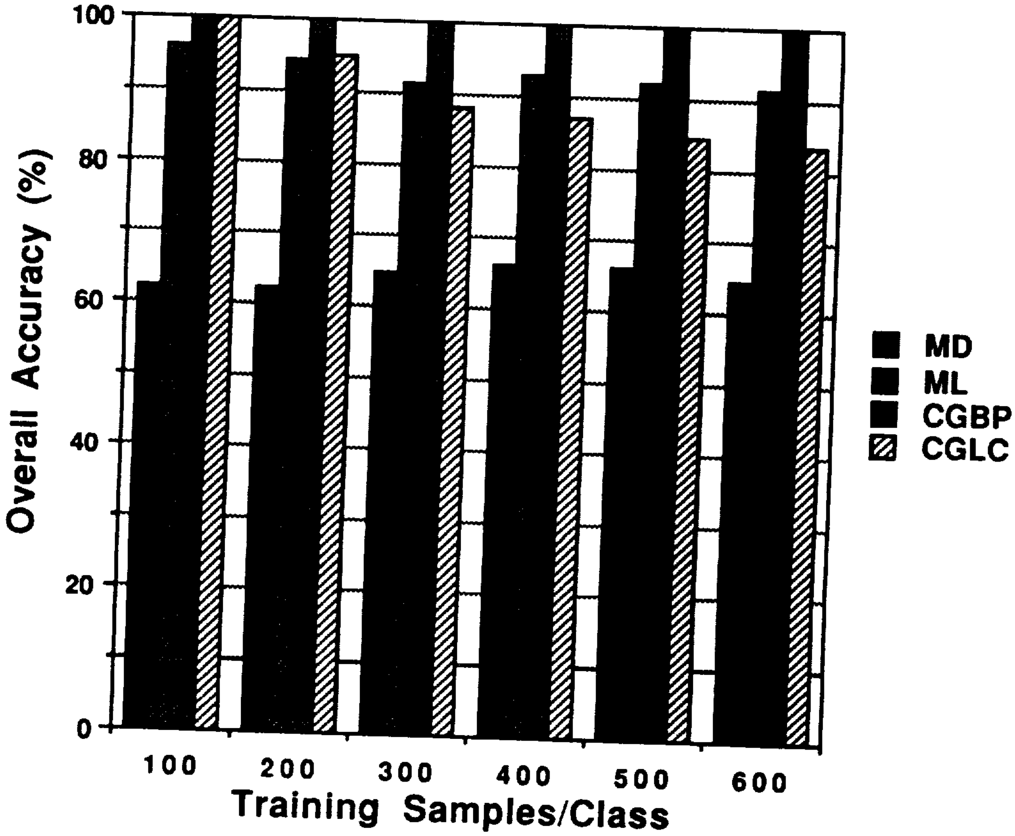


Figure 4.10 Classification of Training Data (20 Dimensions)

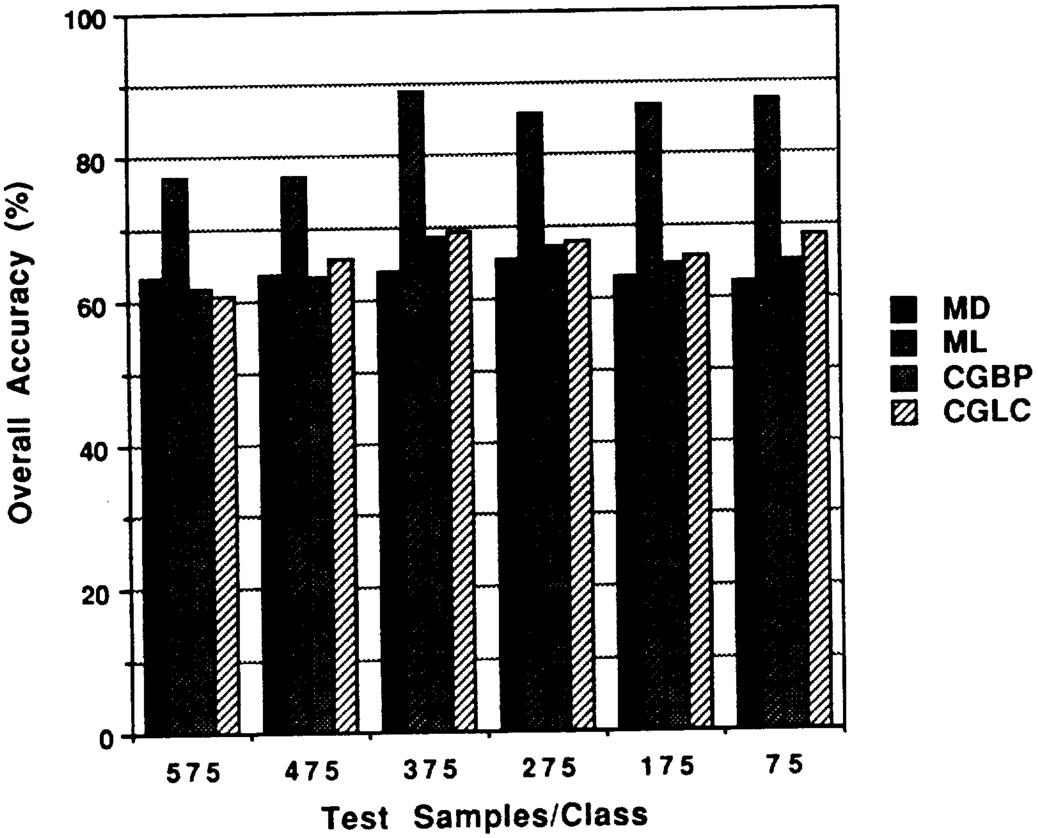


Figure 4.11 Classification of Test Data (20 Dimensions)

The CGLC (Tables 4.72 (training) and 4.73 (test)) was in the same way as the CGBP trained with Gray-coded binary input data (240 input neurons). The CGLC did rather well in training. With 100 training samples per class it was perfect but with increased sample size it always did worse. For the test data, it showed performance similar to the CGBP. The CGLC is not as time consuming during training as the CGBP (because of the hidden neurons in the CGBP). The CGBP required from 1.5 to 7 times more time to train and classify the data than the CGLC in this experiment. Thus the CGLC is a better alternative for the 20-dimensional data in this experiment.

4.4.2 40-Dimensional Data

The 40-dimensional data are relatively separable, as shown in Table 4.74. Predictably the average JM distance increased when 20 features were added to the 20-dimensional data in Section 4.4.1. The results for classification of the 40-dimensional data are shown in Tables 4.75 (MD training), 4.76 (MD test), 4.77 (ML training), 4.78 (ML test), 4.79 (CGBP training), 4.80 (CGBP test), 4.81 (CGLC training) and 4.82 (CGLC test). The results are also summarized in Figures 4.12 (training) and 4.13 (test). The performance of the MD algorithm (Table 4.75 (training) and 4.76 (test)) was very similar to the classification result using the 20-dimensional data. Classification time increased about a factor of 2 when 20 dimensions were added, but the MD algorithm was, as expected, much faster than all other methods.

The accuracy of the ML method (Tables 4.77 (training) and 4.78 (test)) increased when 40 dimensions were used instead of 20, but it took about 3.9 times longer in training and classification than for the 20-dimensional data.

Table 4.74

Pairwise JM Distances for the 40-Dimensional Simulated HIRIS Data.

Class #	2	3
1	1.41189	1.40192
2	-	1.32275
Average:	1.378855	

Table 4.75

Minimum Euclidean Distance Classifier Applied to 40-Dimensional Simulated HIRIS Data: Training Samples.

# of Training Samples	CPU Time	Percent Agreement with Reference for Class			
		1	2	3	OA
100	4	84.0	47.0	56.0	62.33
200	4	84.0	49.0	55.0	62.67
300	4	85.0	50.0	59.3	64.78
400	4	84.0	54.8	60.3	66.33
500	4	85.4	51.8	61.8	66.33
600	4	84.8	50.3	59.8	65.00

Table 4.76

Minimum Euclidean Distance Classifier Applied to 40-Dimensional Simulated HIRIS Data: Test Samples.

# of Training Samples	Percent Agreement with Reference for Class			
	1	2	3	OA
100	83.1	48.9	58.8	63.59
200	82.9	48.2	59.6	63.58
300	83.5	51.2	58.1	64.27
400	80.4	44.4	69.8	64.85
500	73.7	46.3	68.6	62.86
600	80.0	58.7	49.3	62.67

Table 4.77

Maximum Likelihood Method for Gaussian Data Applied to 40-Dimensional Simulated HIRIS Data: Training Samples.

# of Training Samples	CPU Time	Percent Agreement with Reference for Class 1	2	3	OA
100	61	100.0	100.0	100.0	100.00
200	61	100.0	100.0	99.0	99.67
300	62	100.0	97.3	97.3	98.22
400	62	100.0	97.8	97.8	98.50
500	67	99.8	97.6	96.8	98.07
600	75	100.0	97.2	96.5	97.89

Table 4.78

Maximum Likelihood Method for Gaussian Data Applied to 40-Dimensional Simulated HIRIS Data: Test Samples.

# of Training Samples	Percent Agreement with Reference for Class 1	2	3	OA
100	90.8	50.3	77.0	72.70
200	92.6	55.8	73.5	73.96
300	98.1	92.5	93.1	94.58
400	97.8	91.6	92.7	94.06
500	97.7	92.0	92.6	94.10
600	94.7	93.3	93.3	93.78

Table 4.79

Conjugate Gradient Backpropagation Applied to 40-Dimensional Simulated HIRIS Data: Training Samples.

Sample size	Number of iterations	CPU time	Percent Agreement with Reference for Class 1	2	3	OA
100	64	485	100.0	100.0	100.0	100.00
200	150	1548	100.0	100.0	100.0	100.00
300	374	4889	100.0	100.0	100.0	100.00
400	274	5225	100.0	100.0	100.0	100.00
500	264	6386	100.0	100.0	100.0	100.00
600	524	14899	100.0	100.0	100.0	100.00

Table 4.80

Conjugate Gradient Backpropagation Applied to 40-Dimensional Simulated HIRIS Data: Test Samples.

Sample size	Number of iterations	Percent Agreement with Reference for Class 1	2	3	OA
100	64	87.1	57.6	54.6	66.43
200	150	83.8	56.4	48.2	62.81
300	374	85.6	61.6	60.5	69.24
400	274	86.2	57.1	95.6	67.52
500	264	83.4	54.9	57.7	65.33
600	524	85.3	68.0	60.0	71.11

Table 4.81

Conjugate Gradient Linear Classifier Applied to 40-Dimensional Simulated HIRIS Data: Training Samples.

Sample size	Number of iterations	CPU time	Percent Agreement with Reference for Class			
			1	2	3	OA
100	146	194	100.0	100.0	100.0	100.00
200	469	898	100.0	100.0	100.0	100.00
300	903	2461	99.7	99.3	99.7	99.56
400	650	2293	100.0	98.0	97.0	98.33
500	629	2657	100.0	89.2	89.4	92.87
600	492	2492	100.0	87.2	85.3	90.83

Table 4.82

Conjugate Gradient Linear Classifier Applied to 40-Dimensional Simulated HIRIS Data: Test Samples.

Sample size	Number of iterations	Percent Agreement with Reference for Class			
		1	2	3	OA
100	146	85.7	58.1	48.7	64.17
200	469	82.5	50.7	48.2	60.49
300	903	81.1	61.9	53.3	65.42
400	650	86.9	63.6	56.7	69.09
500	629	88.6	60.6	54.9	68.00
600	492	90.7	64.0	50.7	68.44

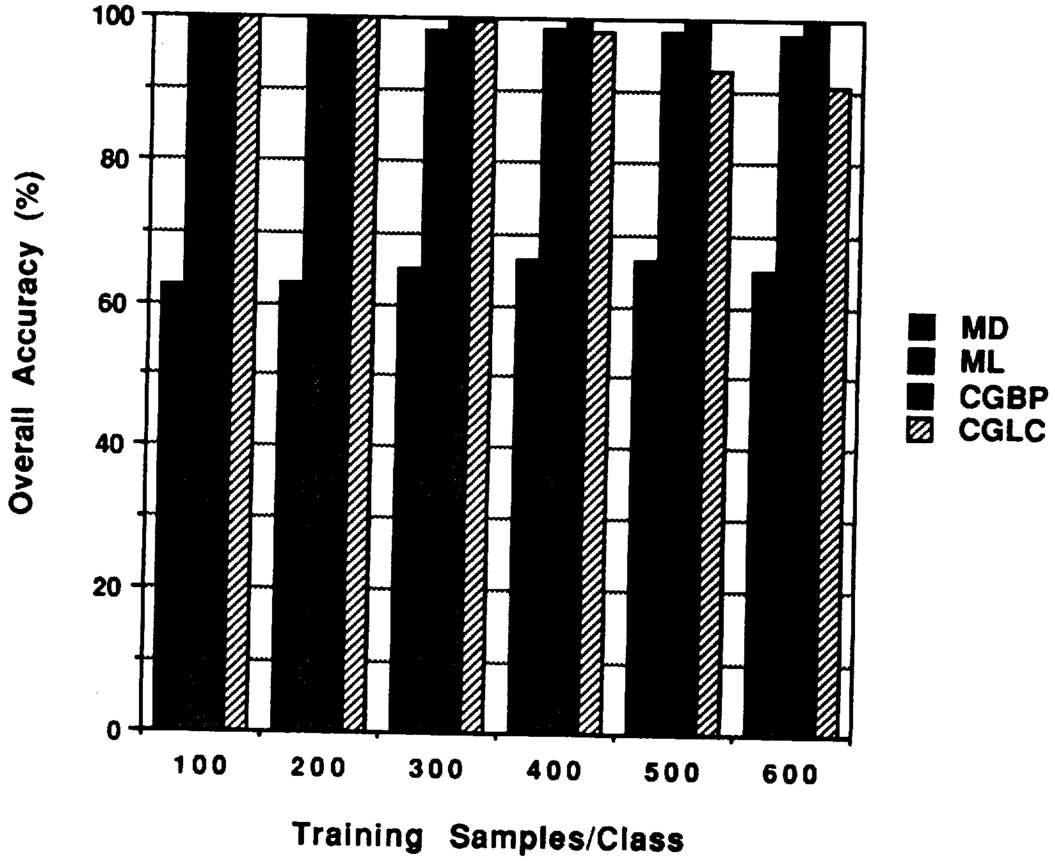


Figure 4.12 Classification of Training Data (40 Dimensions)

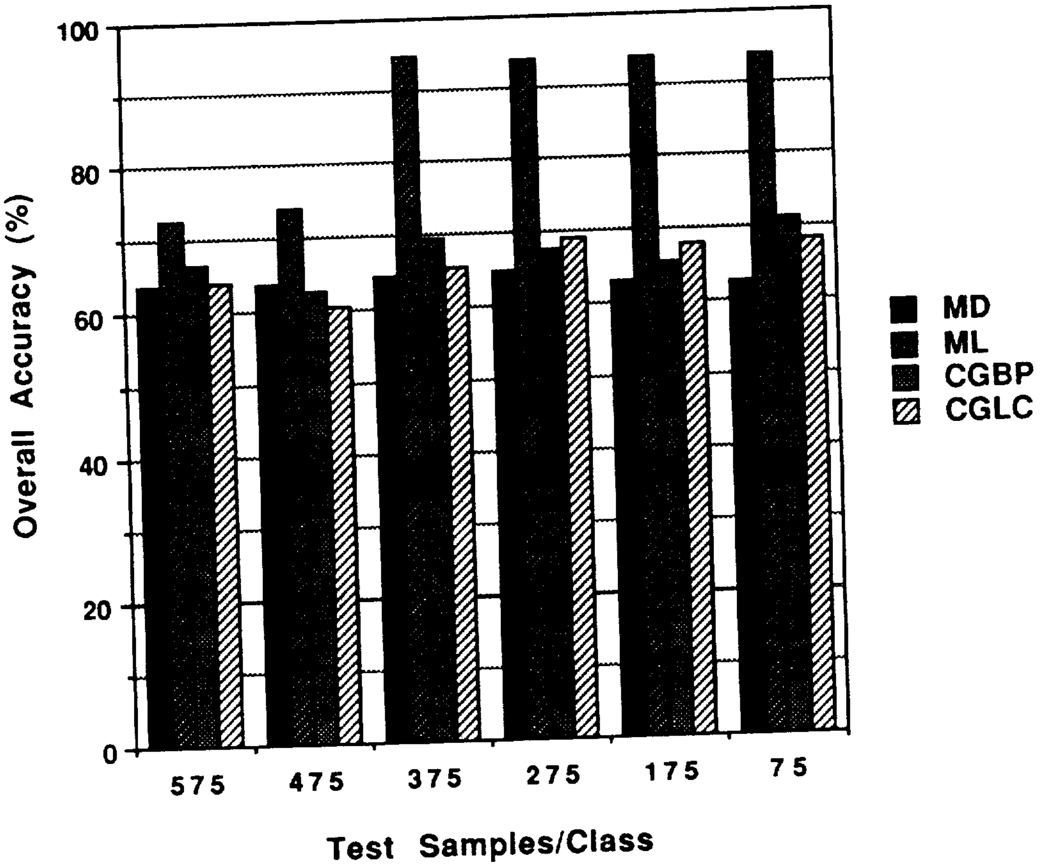


Figure 4.13 Classification of Test Data (40 Dimensions)

The classification accuracy of training data was nearly perfect for all sample sizes. As for the 20-dimensional data the accuracy of test data improved significantly when 300 or more training samples per class were used. Thus, overall the performance of the ML method was very good for the 40-dimensional data set.

The CGBP neural network (Tables 4.79 (training) and 4.80 (test)) was trained with 480 input neurons and 15 hidden neurons. It was again trained to perfection for every sample size and again the training time grew with increasing sample size. The training and classification of the 40-dimensional data took up to 3 times longer than for the 20-dimensional data. For 600 samples per class the neural net converged in just over 4 hours of CPU time (200 times longer than the ML method). However, the classification accuracy of the test samples was not improved greatly for the 40-dimensional data. The most dramatic improvement was for 600 training samples per class.

The CGLC (480 input neurons) (Tables 4.81 (training) and 4.82 (test)) showed an improvement in terms of accuracy of training data when 40 dimensions were used instead of 20. As in the case of the 20-dimensional data the accuracy of training data decreased with increased sample size. The classification accuracy of test data was similar to the 20-dimensional case. The CGLC took up to 5 times longer to converge for 40 dimensions as compared to 20 dimensions. However, it was in most cases more than two times faster than the CGBP and gave similar classification results.

4.4.3 60-Dimensional Data

The results of classification of the 60-dimensional data are summarized in Figures 4.14 (training) and 4.15 (test). In classification of 60-dimensional data the MD algorithm (Tables 4.83 (training) and 4.84 (test)) showed a very similar performance to classification of the other high-dimensional data sets. It was about 3 times slower than in classification of the 20-dimensional data.

The ML method could not be applied to the 60-dimensional data since the covariance matrices were singular. The SMC and the LOP were used instead. In order to use the SMC algorithm, the data had be split into two or more independent data sources. The correlations between the spectral channels can be visualized as shown in Figure 4.16; the brightness indicates the correlation. The lighter the tone, the more correlated are the spectral bands. (The black regions from 1.35 μm to 1.47 μm and 1.81 μm and 1.97 μm are the water absorption bands.) By looking at Figure 4.16, it was determined that the spectral region from 0.7 μm to 1.35 μm was uncorrelated from the other spectral bands. Twenty data channels were in the spectral region from 0.7 μm to 1.35 μm, which was treated as data source #1. Source #2 consisted of the other 40 data channels. The information classes were modeled by the Gaussian distribution in both data sources. The JM distance separabilities of the data sources are shown in Tables 4.85 (source #1) and 4.86 (source #2). The information classes in data sources were relatively separable but the classes in source #2 had a higher average JM distance than the classes in source #1.

The results of the SMC classifications with respect to different sample sizes and various source-specific weights are shown in Tables 4.87 through

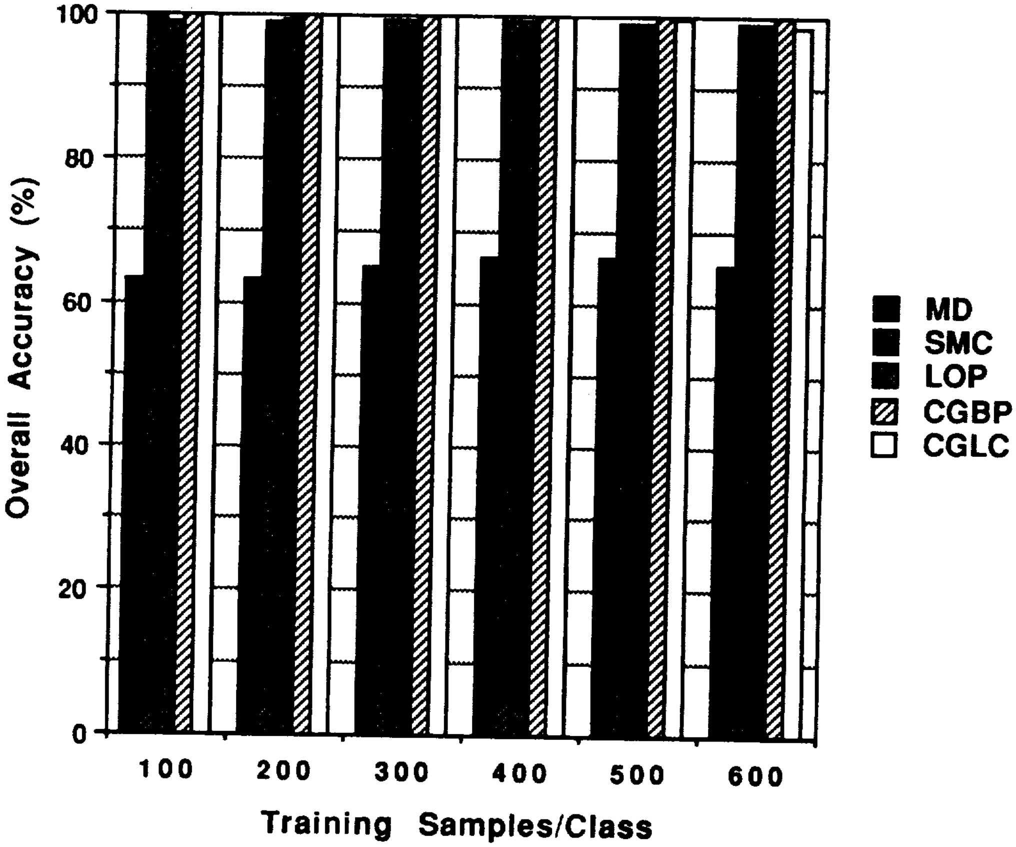


Figure 4.14 Classification of Training Data (60 Dimensions)

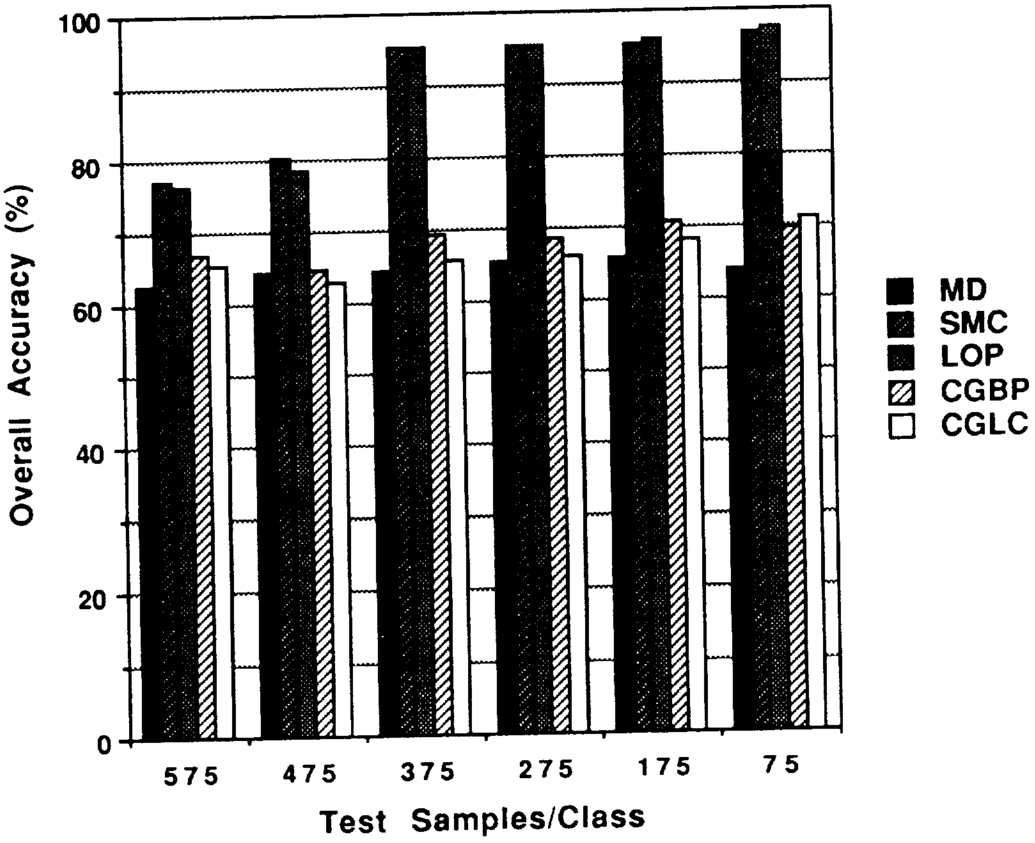


Figure 4.15 Classification of Test Data (60 Dimensions)

Table 4.83

Minimum Euclidean Distance Classifier Applied to 60-Dimensional Simulated HIRIS Data: Training Samples.

# of Training Samples	CPU Time	Percent Agreement with Reference for Class 1	2	3	OA
100	5	83.0	50.0	57.0	63.33
200	6	84.0	51.0	55.0	63.33
300	6	85.3	50.7	59.0	65.00
400	6	83.8	55.8	60.3	66.58
500	6	85.2	53.0	61.4	66.53
600	6	85.2	51.5	59.8	65.50

Table 4.84

Minimum Euclidean Distance Classifier Applied to 60-Dimensional Simulated HIRIS Data: Test Samples.

# of Training Samples	Percent Agreement with Reference for Class 1	2	3	OA
100	83.5	50.1	59.0	64.17
200	83.6	49.5	59.8	64.28
300	84.0	53.9	58.4	65.42
400	81.1	45.8	70.5	65.82
500	74.3	48.0	69.1	63.81
600	80.0	64.0	48.0	64.00

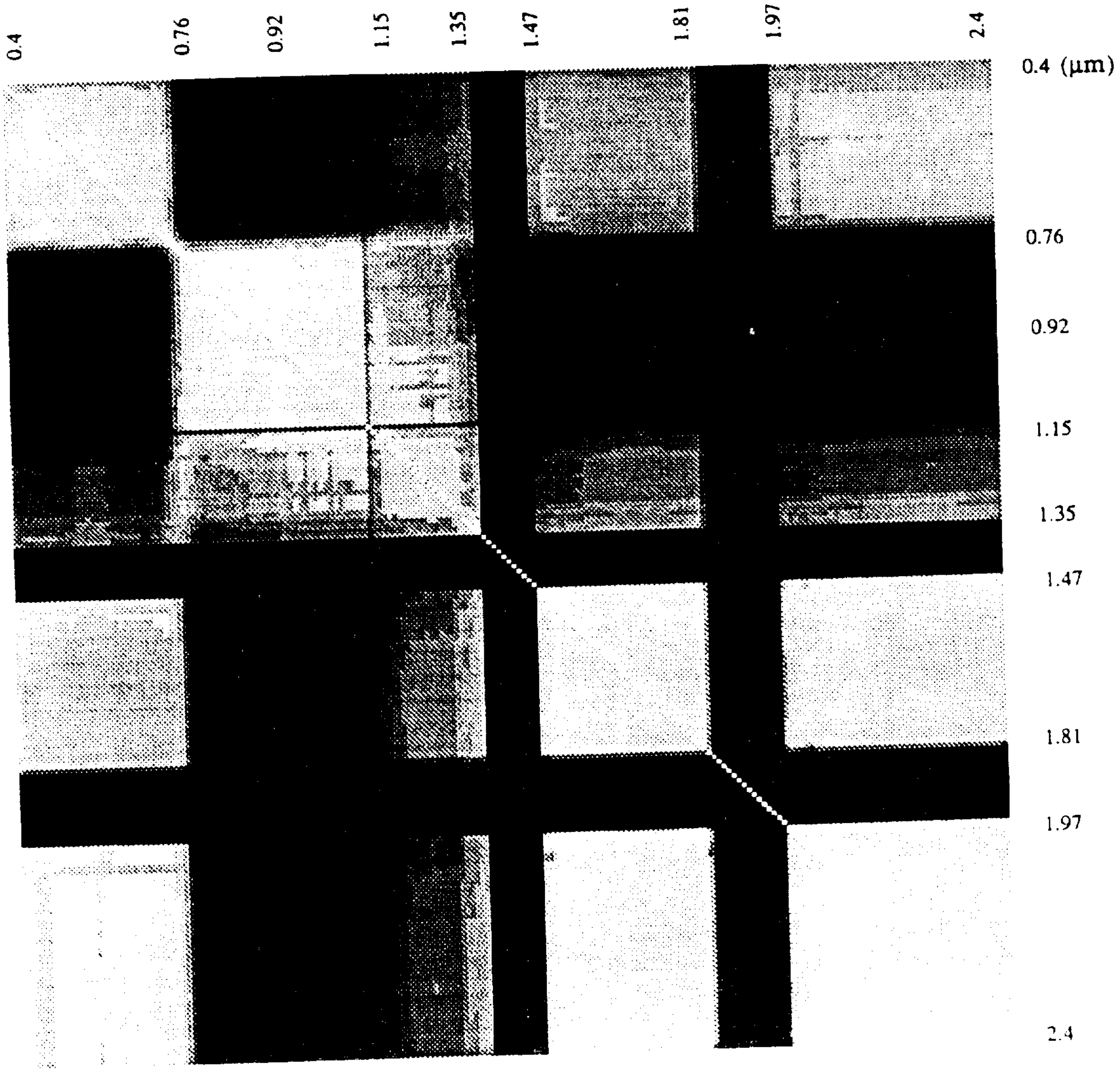


Figure 4.16 Global Statistical Correlation Coefficient Image of HIRIS Data Set

Table 4.85

Pairwise JM Distances for Data Source #1.

Class #	2	3
1	1.31192	1.24447
2	-	0.96362
Average:	1.173336	

Table 4.86

Pairwise JM Distances for Data Source #2.

Class #	2	3
1	1.40908	1.39607
2	-	1.33653
Average:	1.380562	

4.92. Ranking of the sources according to the source-specific reliability measures based on the classification accuracy of training data, the equivocation measure (Table 4.93) and JM distance separability (Table 4.94) agreed in all cases, regardless of sample size. The reliability measures always estimated source #2 as more reliable than source #1. Using these reliability measures to weight the data sources in combination gave the highest accuracies of training data for sample sizes up to 300 training samples per class (Tables 4.87, 4.88 and 4.89). However, the same weights did not achieve the best accuracies for test data. The differences were significant for 100 and 200 samples per class, where the "best" results were reached when source #1 got the weight 1.0 and source #2 was weighted by either 0.1 or 0.2. These unexpected results suggest that the data sources were undertrained with only 100 and 200 samples per class. When 300 samples per class were used (Table 4.89) the highest test accuracy was reached when source #1 was weighted by 1.0 and source #2 by 0.9. However, several other weights gave excellent accuracies as shown in Table 4.89. The SMC gave the best test performance when 400 or more training samples were used for each class (Tables 4.90, 4.91 and 4.92) and source # 2 was given more weight than source #1. Using 400 or more training samples for the high-dimensional data was sufficient. In most cases several weight combinations could achieve the highest accuracies.

The results using the LOP (Tables 4.95 through 4.100) were very similar to the SMC results. Both the SMC and LOP were excellent in the classification of the 60-dimensional data set. For both methods the classification accuracy of test samples increased with the number of training samples used. Both of these algorithms were very fast, with a slight edge to the LOP which uses

Table 4.87

Statistical Multisource Classification of Simulated HIRIS Data (100 Training and 575 Test Samples per Class).

Percent Agreement with Reference for Class								
	Training				Testing			
	1	2	3	OA	1	2	3	OA
	Single Sources							
Source #1	99.0	93.0	96.0	96.00	85.9	67.3	57.7	70.32
Source #2	100.0	100.0	100.0	100.00	83.3	47.8	82.1	71.07
s1 s2	Multiple Sources							
1. 1.	100.0	100.0	100.0	100.00	89.6	51.0	83.1	74.55
1. .9	100.0	100.0	100.0	100.00	89.7	50.8	83.3	74.61
1. .8	100.0	100.0	100.0	100.00	96.6	51.3	83.3	75.07
1. .7	100.0	100.0	100.0	100.00	91.1	51.3	83.5	75.30
1. .6	100.0	100.0	100.0	100.00	90.6	52.3	83.7	75.53
1. .5	100.0	100.0	100.0	100.00	91.1	52.9	83.0	75.48
1. .4	100.0	100.0	100.0	100.00	91.1	54.6	83.1	76.29
1. .3	100.0	100.0	100.0	100.00	91.3	55.3	83.5	76.70
1. .2	100.0	100.0	100.0	100.00	90.1	64.2	77.6	77.28
1. .1	100.0	100.0	100.0	100.00	90.1	64.2	77.6	77.28
1. .0	99.0	93.0	96.0	96.00	85.9	67.3	57.7	70.32
.9 1.	100.0	100.0	100.0	100.00	89.6	50.4	82.8	74.26
.8 1.	100.0	100.0	100.0	100.00	89.6	50.3	82.6	74.14
.7 1.	100.0	100.0	100.0	100.00	88.7	50.1	82.8	73.86
.6 1.	100.0	100.0	100.0	100.00	88.3	49.9	83.0	73.74
.5 1.	100.0	100.0	100.0	100.00	87.8	49.7	83.0	73.51
.4 1.	100.0	100.0	100.0	100.00	87.3	49.4	82.8	73.16
.3 1.	100.0	100.0	100.0	100.00	87.0	48.7	82.6	72.77
.2 1.	100.0	100.0	100.0	100.00	86.1	48.3	81.9	72.12
.1 1.	100.0	100.0	100.0	100.00	84.5	48.0	82.1	71.54
.0 1.	100.0	100.0	100.0	100.00	83.3	47.8	82.1	71.07
# of pixels	100	100	100	300	575	575	575	1725

The columns labeled s1 and s2 indicate the weights applied to sources 1 (s1) and 2 (s2).

CPU time for training and classification: 81 sec.

Table 4.88

Statistical Multisource Classification of Simulated HIRIS Data (200 Training and 475 Test Samples per Class).

	Percent Agreement with Reference for Class							
	Training				Testing			
	1	2	3	OA	1	2	3	OA
	Single Sources							
source #1	97.5	91.0	92.0	93.50	88.2	70.3	57.3	71.93
source #2	100.0	100.0	99.5	99.83	88.0	53.1	80.4	73.82
sl s2	Multiple Sources							
1. 1.	100.0	100.0	99.5	99.83	89.1	55.8	82.3	75.75
1. .9	100.0	100.0	99.5	99.83	89.1	56.0	82.3	75.79
1. .8	100.0	100.0	99.5	99.83	89.5	55.8	82.5	75.93
1. .7	100.0	100.0	99.5	99.83	89.7	56.0	82.5	76.07
1. .6	100.0	100.0	99.5	99.83	89.9	58.1	82.5	76.84
1. .5	100.0	100.0	99.5	99.83	90.5	60.0	83.4	77.96
1. .4	100.0	99.5	99.0	99.50	90.3	62.3	84.0	78.88
1. .3	100.0	99.0	99.0	99.33	90.7	63.6	84.4	79.58
1. .2	100.0	97.5	99.0	98.83	90.7	71.2	76.4	79.37
1. .1	100.0	96.0	98.5	98.17	90.5	71.2	76.4	79.37
1. .0	97.5	91.0	92.0	93.50	88.2	70.3	57.3	71.93
.9 1.	100.0	100.0	99.5	99.83	89.5	55.4	82.3	75.72
.8 1.	100.0	100.0	99.5	99.83	89.3	55.4	82.5	75.72
.7 1.	100.0	100.0	99.5	99.83	89.1	55.4	82.3	75.79
.6 1.	100.0	100.0	99.5	99.83	88.9	54.9	82.3	75.37
.5 1.	100.0	100.0	99.5	99.83	88.6	54.5	82.1	75.09
.4 1.	100.0	100.0	99.5	99.83	88.2	54.3	81.7	74.74
.3 1.	100.0	100.0	99.5	99.83	88.2	53.5	81.7	74.46
.2 1.	100.0	100.0	99.5	99.83	88.0	53.1	81.3	74.10
.1 1.	100.0	100.0	99.5	99.83	87.8	52.8	80.8	73.82
.0 1.	100.0	100.0	99.5	99.83	88.0	53.1	80.4	73.82
# of pixels	200	200	200	600	475	475	475	1425

The columns labeled s1 and s2 indicate the weights applied to sources 1 (s1) and 2 (s2).

CPU time for training and classification: 85 sec.

Table 4.89

Statistical Multisource Classification of Simulated HIRIS Data (300 Training and 375 Test Samples per Class).

	Percent Agreement with Reference for Class							
	Training				Testing			
	1	2	3	OA	1	2	3	OA
	Single Sources							
source #1	96.3	86.7	89.3	90.78	90.7	82.4	79.5	84.18
source #2	100.0	99.0	98.3	99.11	96.0	94.4	94.1	94.84
sl s2	Multiple Sources							
1. 1.	100.0	99.0	98.7	99.22	96.5	96.0	95.2	95.91
1. .9	100.0	98.7	99.3	99.33	97.1	96.3	95.2	96.18
1. .8	100.0	98.0	99.3	99.11	96.0	96.3	95.2	95.82
1. .7	99.7	98.0	99.0	98.89	94.7	95.7	93.8	94.76
1. .6	98.7	95.7	97.7	97.33	94.1	93.3	91.7	93.07
1. .5	98.3	95.0	96.7	96.67	93.3	92.3	89.1	91.56
1. .4	97.3	94.0	95.0	95.44	92.8	90.1	87.7	90.22
1. .3	97.0	91.3	93.3	93.89	92.3	87.5	86.4	88.71
1. .2	96.7	90.7	92.3	93.22	91.7	86.1	83.7	87.20
1. .1	96.3	89.0	90.7	92.00	91.2	84.3	81.9	85.78
1. .0	96.3	87.0	88.7	90.67	90.7	82.4	79.5	84.18
.9 1.	100.0	99.0	98.7	99.22	96.8	95.7	94.9	95.82
.8 1.	100.0	99.0	98.7	99.22	96.5	95.5	94.9	95.64
.7 1.	100.0	99.3	98.7	99.33	96.5	95.5	94.7	95.56
.6 1.	100.0	99.3	98.7	99.33	96.5	95.5	94.4	95.47
.5 1.	100.0	99.3	98.7	99.33	96.3	94.9	94.4	95.29
.4 1.	100.0	99.3	98.7	99.33	96.3	94.9	94.4	95.20
.3 1.	100.0	99.3	98.7	99.33	96.3	94.9	94.4	95.20
.2 1.	100.0	99.0	98.7	99.22	96.0	94.4	94.4	94.93
.1 1.	100.0	99.0	98.7	99.22	96.0	94.7	94.4	95.02
.0 1.	100.0	99.0	98.3	99.11	96.0	94.4	94.1	94.84
# of pixels	300	300	300	900	375	375	375	1125

The columns labeled sl and s2 indicate the weights applied to sources 1 (sl) and 2 (s2).

CPU time for training and classification: 87 sec.

Table 4.90

Statistical Multisource Classification of Simulated HIRIS Data (400 Training and 275 Test Samples per Class).

	Percent Agreement with Reference for Class							
	Training				Testing			
	1	2	3	OA	1	2	3	OA
	Single Sources							
source #1	96.8	87.3	87.3	90.42	83.6	82.2	71.3	79.03
source #2	100.0	99.0	98.0	99.00	94.2	94.2	96.7	95.03
s1 s2	Multiple Sources							
1. 1.	99.8	99.3	98.8	99.25	94.2	94.5	96.4	95.03
1. .9	99.8	99.0	99.0	99.25	93.8	94.5	95.6	94.67
1. .8	99.8	99.0	99.0	99.25	93.5	94.9	95.6	94.67
1. .7	99.8	99.0	99.0	99.25	92.7	94.5	95.6	94.30
1. .6	99.3	98.5	99.0	98.92	90.9	94.9	96.0	93.94
1. .5	99.0	98.3	99.0	98.75	90.2	93.8	96.0	93.33
1. .4	98.5	97.8	98.8	98.33	88.4	92.4	95.6	92.12
1. .3	98.5	95.5	97.5	97.17	85.5	91.6	89.8	89.70
1. .2	98.5	93.0	95.5	95.67	85.5	88.7	88.4	87.52
1. .1	97.3	90.5	92.5	93.42	84.4	85.5	82.2	84.00
1. .0	96.8	87.5	87.3	90.42	83.6	82.2	71.3	79.03
.9 1.	99.8	99.3	98.8	99.25	94.5	94.9	96.4	95.27
.8 1.	99.8	99.3	99.0	99.33	94.5	95.3	96.4	95.39
.7 1.	99.8	99.3	99.0	99.33	94.5	94.9	96.4	95.27
.6 1.	100.0	99.3	99.0	99.42	94.5	94.5	96.0	95.03
.5 1.	100.0	99.3	98.8	99.33	94.5	94.5	96.0	95.03
.4 1.	100.0	99.3	98.8	99.33	94.5	94.5	96.0	95.03
.3 1.	100.0	99.3	98.8	99.33	95.3	94.5	96.4	95.39
.2 1.	100.0	99.3	98.5	99.25	94.9	94.5	97.1	95.27
.1 1.	100.0	99.0	98.3	99.08	94.2	94.5	97.1	95.27
.0 1.	100.0	99.0	98.0	99.00	94.2	94.2	96.7	95.03
# of pixels	400	400	400	1200	275	275	275	825

The columns labeled s1 and s2 indicate the weights applied to sources 1 (s1) and 2 (s2).

CPU time for training and classification: 90 sec.

Table 4.91

Statistical Multisource Classification of Simulated HIRIS Data (500 Training and 175 Test Samples per Class).

	Percent Agreement with Reference for Class							
	Training				Testing			
	1	2	3	OA	1	2	3	OA
	Single Sources							
source #1	95.8	88.0	84.8	89.53	78.9	78.3	73.7	76.95
source #2	99.8	98.4	96.8	98.33	94.3	94.3	98.9	95.81
sl s2	Multiple Sources							
1. 1.	99.6	98.8	98.2	98.87	92.6	95.4	97.7	95.23
1. .9	99.6	98.8	98.4	98.93	93.1	95.4	97.7	95.43
1. .8	99.6	98.6	98.2	98.80	93.1	94.9	97.7	95.24
1. .7	99.2	98.8	98.4	98.80	93.1	96.0	97.7	95.62
1. .6	99.0	98.6	98.4	98.67	90.3	96.0	97.7	94.67
1. .5	98.4	98.0	98.4	98.27	88.0	95.4	97.7	93.71
1. .4	98.0	96.8	97.8	97.53	87.4	93.7	97.1	92.76
1. .3	97.8	95.0	96.4	96.40	85.7	92.6	91.4	89.90
1. .2	97.6	92.4	94.0	94.67	84.6	88.6	89.7	87.62
1. .1	96.6	90.6	90.4	92.53	82.9	83.4	83.4	83.24
1. .0	96.2	88.0	84.8	89.53	78.9	78.3	73.7	76.95
.9 1.	99.6	98.8	98.0	98.80	92.6	95.4	97.7	95.23
.8 1.	99.6	98.8	98.0	98.80	93.1	94.9	97.7	95.24
.7 1.	99.6	98.8	97.8	98.67	94.3	94.9	97.7	95.62
.6 1.	99.6	98.6	97.8	98.67	94.9	94.9	98.9	96.19
.5 1.	99.6	98.6	97.8	98.67	94.3	94.9	99.4	96.19
.4 1.	99.6	98.6	97.8	98.67	94.3	94.9	99.4	96.19
.3 1.	99.6	98.6	97.8	98.67	94.3	94.9	99.4	96.19
.2 1.	99.6	98.6	97.4	98.53	93.7	94.3	99.4	95.81
.1 1.	99.8	98.4	97.2	98.47	94.3	94.3	99.4	96.00
.0 1.	99.8	98.4	96.8	98.33	94.3	94.3	98.9	95.81
# of pixels	500	500	500	1500	175	175	175	525

The columns labeled sl and s2 indicate the weights applied to sources 1 (sl) and 2 (s2).

CPU time for training and classification: 90 sec.

Table 4.92

Statistical Multisource Classification of Simulated HIRIS Data (600 Training and 75 Test Samples per Class).

Percent Agreement with Reference for Class								
	Training				Testing			
	1	2	3	OA	1	2	3	OA
	Single Sources							
source #1	95.2	86.5	85.2	88.94	72.0	78.7	73.3	74.67
source #2	100.0	98.3	96.8	98.39	92.0	97.3	97.3	95.56
s1 s2	Multiple Sources							
1. 1.	99.5	98.5	98.3	98.78	92.0	100.0	98.7	96.89
1. .9	99.5	98.3	98.5	98.78	92.0	100.0	98.7	96.89
1. .8	99.5	98.5	98.5	98.83	92.0	100.0	98.7	96.89
1. .7	99.3	98.7	98.3	98.78	92.0	97.3	98.7	96.00
1. .6	99.3	98.5	98.2	98.39	89.3	97.3	98.7	95.11
1. .5	99.0	98.0	98.2	98.39	85.3	97.3	98.7	93.78
1. .4	98.2	96.7	97.8	97.56	82.7	93.3	96.0	90.67
1. .3	97.5	95.3	96.2	96.33	80.0	92.0	93.3	88.44
1. .2	97.2	92.7	94.0	94.61	76.0	88.0	96.7	84.89
1. .1	96.3	90.3	91.3	92.67	73.3	85.3	88.0	82.22
1. .0	95.2	86.5	85.2	88.94	72.0	80.0	78.7	74.67
.9 1.	99.5	98.3	98.3	98.72	92.0	100.0	98.7	96.89
.8 1.	99.7	98.3	98.3	98.78	92.0	100.0	98.7	96.89
.7 1.	99.7	98.3	98.2	98.72	92.0	100.0	98.7	96.89
.6 1.	99.8	98.3	98.0	98.72	92.0	100.0	98.7	96.89
.5 1.	99.8	98.3	98.0	98.72	92.0	100.0	98.7	96.89
.4 1.	99.8	98.3	98.0	98.72	92.0	100.0	98.7	96.89
.3 1.	99.8	98.3	97.7	98.61	90.7	100.0	98.7	96.44
.2 1.	99.8	98.3	97.5	98.56	90.7	98.7	98.7	96.00
.1 1.	100.0	98.3	97.0	98.44	92.0	98.7	97.3	96.00
.0 1.	100.0	98.3	96.8	98.39	92.0	97.3	97.3	95.56
# of pixels	600	600	600	1800	75	75	75	225

The columns labeled s1 and s2 indicate the weights applied to sources 1 (s1) and 2 (s2).

CPU time for training and classification: 90 sec.

Table 4.93

Source-Specific Equvivocations for Simulated HIRIS Data Versus Number of Training Samples.

Equivocation		
Training Samples	Source #	
	1	2
100	0.2581	0.0000
200	0.3325	0.0105
300	0.4057	0.0637
400	0.3958	0.0686
500	0.4154	0.0947
600	0.4356	0.0981

Table 4.94

Source-Specific JM Distances for Simulated HIRIS Data Versus Number of Training Samples.

JM Distancc		
Training Samples	Source #	
	1	2
100	1.313423	1.413598
200	1.273153	1.410736
300	1.208442	1.392419
400	1.234641	1.389006
500	1.214116	1.386011
600	1.195292	1.383060

Table 4.95

Linear Opinion Pool Applied in Classification of Simulated HIRIS Data (100 Training and 575 Test Samples per Class).

Percent Agreement with Reference for Class								
	Training				Testing			
	1	2	3	OA	1	2	3	OA
	Single Sources							
source #1	99.0	93.0	96.0	96.00	85.9	67.3	57.7	70.32
source #2	100.0	100.0	100.0	100.00	83.3	47.8	82.1	71.07
s1 s2	Multiple Sources							
1. 1.	100.0	100.0	100.0	100.00	89.6	50.6	82.3	74.14
1. .9	100.0	100.0	100.0	100.00	91.5	61.0	79.8	77.45
1. .8	100.0	100.0	99.0	99.67	90.8	65.9	73.0	76.58
1. .7	100.0	98.0	99.0	99.00	90.4	67.5	71.0	76.29
1. .6	100.0	97.0	99.0	98.67	89.4	67.3	69.2	75.30
1. .5	99.0	97.0	99.0	98.33	88.2	67.8	66.6	74.20
1. .4	99.0	97.0	99.0	98.33	88.2	67.8	66.6	74.20
1. .3	99.0	96.0	99.0	98.00	87.5	67.1	62.2	72.29
1. .2	99.0	95.0	97.0	97.00	87.0	67.0	60.3	71.42
1. .1	99.0	94.0	96.0	96.33	86.3	66.6	59.0	70.61
1. .0	99.0	93.0	96.0	96.00	85.9	67.3	57.7	70.32
.9 1.	100.0	100.0	100.0	100.00	85.7	50.3	82.1	72.70
.8 1.	100.0	100.0	100.0	100.00	85.4	49.9	81.9	72.41
.7 1.	100.0	100.0	100.0	100.00	84.7	49.7	82.1	72.17
.6 1.	100.0	100.0	100.0	100.00	84.9	49.6	81.9	72.12
.5 1.	100.0	100.0	100.0	100.00	84.3	49.4	81.7	71.83
.4 1.	100.0	100.0	100.0	100.00	84.0	48.9	81.9	71.59
.3 1.	100.0	100.0	100.0	100.00	84.0	48.3	81.9	71.42
.2 1.	100.0	100.0	100.0	100.00	83.8	48.0	81.9	71.25
.1 1.	100.0	100.0	100.0	100.00	83.8	47.8	82.1	71.25
.0 1.	100.0	100.0	100.0	100.00	83.3	47.8	82.1	71.07
# of pixels	100	100	100	300	575	575	575	1725

The columns labeled s1 and s2 indicate the weights applied to sources 1 (s1) and 2 (s2).

CPU time for training and classification: 81 sec.

Table 4.96

Linear Opinion Pool Applied in Classification of Simulated HIRIS Data (200 Training and 475 Test Samples per Class).

Percent Agreement with Reference for Class								
	Training				Testing			
	1	2	3	OA	1	2	3	OA
	Single Sources							
source #1	97.5	91.0	92.0	93.50	88.2	70.3	57.3	71.93
source #2	100.0	100.0	99.5	99.83	88.0	53.1	80.4	73.82
s1 s2	Multiple Sources							
1. 1.	100.0	100.0	99.5	99.83	89.3	54.5	81.9	75.09
1. .9	100.0	100.0	99.5	99.83	92.0	55.2	80.6	75.93
1. .8	100.0	99.5	98.5	99.33	90.7	72.0	72.4	78.67
1. .7	100.0	98.5	98.5	99.00	90.7	72.0	72.4	78.39
1. .6	100.0	98.5	98.0	98.83	90.5	72.4	71.2	78.04
1. .5	100.0	97.0	97.0	98.00	90.1	71.8	67.6	76.49
1. .4	98.5	97.0	97.0	97.50	89.5	71.2	65.5	75.37
1. .3	98.0	96.0	96.0	96.67	89.3	71.4	63.2	74.60
1. .2	98.0	94.5	95.0	95.83	88.8	71.6	60.4	73.75
1. .1	97.5	92.5	92.5	94.17	88.8	70.3	59.2	73.19
1. .0	97.5	91.0	92.0	93.50	88.2	70.3	57.3	71.93
.9 1.	100.0	100.0	99.5	99.83	88.8	54.5	81.9	75.09
.8 1.	100.0	100.0	99.5	99.83	88.8	54.3	82.1	75.09
.7 1.	100.0	100.0	99.5	99.83	88.2	54.1	81.9	74.74
.6 1.	100.0	100.0	99.5	99.83	88.2	53.9	81.9	74.67
.5 1.	100.0	100.0	99.5	99.83	88.0	53.9	81.5	74.46
.4 1.	100.0	100.0	99.5	99.83	88.0	53.9	81.3	74.18
.3 1.	100.0	100.0	99.5	99.83	88.0	53.1	80.8	73.96
.2 1.	100.0	100.0	99.5	99.83	88.0	52.8	81.1	73.96
.1 1.	100.0	100.0	99.5	99.83	88.0	53.1	80.6	73.89
.0 1.	100.0	100.0	99.5	99.83	88.0	53.1	80.4	73.82
# of pixels	200	200	200	600	475	475	475	1425

The columns labeled s1 and s2 indicate the weights applied to sources 1 (s1) and 2 (s2).

CPU time for training and classification: 84 sec.

Table 4.97

Linear Opinion Pool Applied in Classification of Simulated HIRIS Data (300 Training and 375 Test Samples per Class).

Percent Agreement with Reference for Class								
	Training				Testing			
	1	2	3	OA	1	2	3	OA
	Single Sources							
source #1	96.3	86.7	89.3	90.78	90.7	82.4	79.5	84.18
source #2	100.0	99.0	98.3	99.11	96.0	94.4	94.1	94.84
s1 s2	Multiple Sources							
1. 1.	99.7	99.0	99.0	99.22	96.5	96.0	95.2	95.91
1. .9	99.7	98.7	99.0	99.11	96.5	96.0	95.2	95.91
1. .8	99.7	98.7	99.3	99.22	96.8	96.3	95.2	96.09
1. .7	99.7	98.3	99.3	99.11	96.5	96.3	95.2	96.00
1. .6	99.7	97.3	99.3	99.11	95.7	96.0	95.5	95.73
1. .5	99.7	97.3	99.3	98.78	94.4	96.0	95.7	95.38
1. .4	99.0	97.0	99.7	98.56	93.9	94.4	95.5	94.58
1. .3	98.3	94.7	98.3	97.11	93.1	92.0	92.5	92.53
1. .2	98.0	93.7	95.0	95.56	92.5	88.0	89.1	90.13
1. .1	97.0	90.7	92.7	93.44	92.0	86.9	86.9	88.62
1. .0	96.3	86.7	89.3	90.78	90.7	82.4	79.5	84.18
.9 1.	99.7	99.0	98.7	99.11	96.3	96.3	94.9	95.82
.8 1.	99.7	99.3	98.7	99.22	96.3	95.7	94.9	95.64
.7 1.	99.7	99.3	98.7	99.22	96.3	95.5	94.9	95.56
.6 1.	99.7	99.3	98.7	99.22	96.5	95.5	94.9	95.64
.5 1.	100.0	99.3	98.7	99.33	96.5	95.2	94.7	95.47
.4 1.	100.0	99.3	98.7	99.33	96.3	95.2	94.4	95.29
.3 1.	100.0	99.3	98.7	99.33	96.3	95.2	94.4	95.29
.2 1.	100.0	99.0	98.7	99.22	96.3	94.7	94.4	95.11
.1 1.	100.0	99.0	98.7	99.22	96.0	94.7	94.4	95.02
.0 1.	100.0	99.0	98.3	99.11	96.0	94.4	94.1	94.84
# of pixels	300	300	300	900	375	375	375	1125

The columns labeled s1 and s2 indicate the weights applied to sources 1 (s1) and 2 (s2).

CPU time for training and classification: 86 sec.

Table 4.98

Linear Opinion Pool Applied in Classification of Simulated HIRIS Data (400 Training and 275 Test Samples per Class).

	Percent Agreement with Reference for Class							
	Training				Testing			
	1	2	3	OA	1	2	3	OA
	Single Sources							
source #1	96.8	87.3	87.3	90.42	83.6	82.2	71.3	79.03
source #2	100.0	99.0	98.0	99.00	94.2	94.2	96.7	95.03
s1 s2	Multiple Sources							
1. 1.	99.8	99.3	98.8	99.25	95.3	94.9	96.0	95.39
1. .9	99.8	99.0	98.8	99.17	95.3	94.5	95.3	95.03
1. .8	99.5	99.0	99.0	99.17	94.2	94.2	95.3	94.55
1. .7	99.0	99.0	98.3	99.75	91.3	93.5	92.0	92.24
1. .6	98.8	97.0	97.3	97.67	89.5	92.0	89.5	90.30
1. .5	98.8	95.5	96.3	96.83	86.5	90.5	88.7	88.61
1. .4	98.5	93.3	93.8	95.17	85.1	89.8	86.2	87.03
1. .3	97.8	92.5	92.0	94.08	84.7	87.3	82.5	84.85
1. .2	97.3	90.5	91.0	92.92	84.4	84.7	79.3	82.79
1. .1	97.3	89.0	89.8	92.00	83.6	85.5	75.3	80.48
1. .0	96.8	87.3	87.3	90.42	83.6	82.2	71.3	79.03
.9 1.	100.0	99.3	98.8	99.33	95.3	94.9	96.0	95.39
.8 1.	100.0	99.3	99.0	99.42	95.6	94.9	96.0	95.52
.7 1.	100.0	99.3	99.0	99.42	96.0	94.5	96.0	95.52
.6 1.	100.0	99.3	98.8	99.33	95.6	94.2	96.4	95.27
.5 1.	100.0	99.3	98.8	99.33	95.6	94.5	96.0	95.52
.4 1.	100.0	99.3	98.5	99.25	95.3	94.5	96.4	95.36
.3 1.	100.0	99.3	98.5	99.25	94.9	94.5	96.4	95.27
.2 1.	100.0	99.3	98.5	99.25	94.9	94.5	96.4	95.27
.1 1.	100.0	99.0	98.3	99.08	94.2	94.2	96.7	95.03
.0 1.	100.0	99.0	98.0	99.00	94.2	94.2	96.7	95.03
# of pixels	400	400	400	1200	275	275	275	825

The columns labeled s1 and s2 indicate the weights applied to sources 1 (s1) and 2 (s2).

CPU time for training and classification: 89 sec.

Table 4.99

Linear Opinion Pool Applied in Classification of Simulated HIRIS Data (500 Training and 175 Test Samples per Class).

	Percent Agreement with Reference for Class							
	Training				Testing			
	1	2	3	OA	1	2	3	OA
	Single Sources							
source #1	95.8	88.0	84.8	89.53	78.9	78.3	73.7	76.95
source #2	99.8	98.4	96.8	98.33	94.3	94.3	98.9	95.81
s1 s2	Multiple Sources							
1. 1.	99.6	98.8	98.0	98.80	95.4	95.4	97.7	96.19
1. .9	99.6	98.8	98.2	98.87	94.9	95.4	97.7	96.00
1. .8	99.6	98.8	98.2	98.87	93.7	95.4	97.7	95.62
1. .7	98.6	98.6	98.0	98.40	91.4	94.3	96.0	93.90
1. .6	98.2	97.2	98.4	97.27	89.7	93.7	92.0	91.81
1. .5	98.2	95.4	95.6	96.40	86.3	92.6	90.3	89.71
1. .4	97.6	93.2	93.8	93.87	84.6	89.1	89.1	87.62
1. .3	97.0	91.8	92.8	93.87	82.9	86.9	85.7	85.14
1. .2	96.8	91.0	89.8	92.53	82.9	84.0	82.9	83.24
1. .1	96.4	89.8	87.6	91.27	80.0	80.6	76.6	79.05
1. .0	95.8	88.0	84.8	89.53	78.9	78.3	73.7	76.95
.9 1.	99.6	98.8	98.0	98.80	94.9	95.4	97.7	96.00
.8 1.	99.6	98.8	98.0	98.80	95.4	94.9	98.3	96.19
.7 1.	99.6	98.8	97.8	98.67	94.9	94.9	98.3	96.00
.6 1.	99.6	98.6	97.6	98.67	94.3	94.3	98.9	96.00
.5 1.	99.6	98.6	97.6	98.60	94.3	94.9	99.4	96.19
.4 1.	99.6	98.6	97.4	98.53	94.3	94.9	99.4	96.19
.3 1.	99.6	98.6	97.4	98.53	94.3	94.3	99.4	96.00
.2 1.	99.6	98.6	97.4	98.60	94.3	94.3	99.4	96.00
.1 1.	99.8	98.4	97.2	98.47	94.3	94.3	99.4	96.00
.0 1.	99.8	98.4	96.8	98.33	94.3	94.3	98.9	95.81
# of pixels	500	500	500	1500	175	175	175	525

The columns labeled s1 and s2 indicate the weights applied to sources 1 (s1) and 2 (s2).

CPU time for training and classification: 90 sec.

Table 4.100

Linear Opinion Pool Applied in Classification of Simulated HIRIS Data (600 Training and 75 Test Samples per Class).

Percent Agreement with Reference for Class								
	Training				Testing			
	1	2	3	OA	1	2	3	OA
	Single Sources							
source #1	95.2	86.5	85.2	88.94	72.0	78.7	73.3	74.67
source #2	100.0	98.3	96.8	98.39	92.0	97.3	97.3	95.56
s1 s2	Multiple Sources							
1. 1.	99.7	98.5	98.3	98.83	93.3	100.0	98.7	97.33
1. .9	99.5	98.5	98.5	98.83	93.3	100.0	98.7	97.33
1. .8	99.5	98.3	98.5	98.78	92.0	97.3	98.7	96.00
1. .7	99.2	98.5	98.2	98.61	85.3	97.3	97.3	93.33
1. .6	98.3	97.5	96.5	97.44	81.3	93.3	93.3	89.33
1. .5	97.8	96.3	95.2	96.44	80.0	92.0	92.0	88.00
1. .4	97.2	94.3	94.0	95.17	78.7	89.3	90.7	86.22
1. .3	96.8	91.8	92.2	93.61	74.7	84.0	88.0	82.22
1. .2	96.3	90.2	90.7	92.39	73.3	82.7	85.3	80.44
1. .1	95.5	89.2	87.8	90.83	73.3	81.3	81.3	78.67
1. .0	95.2	86.5	85.2	88.94	72.0	78.7	73.3	74.67
.9 1.	99.8	98.3	98.3	98.83	93.3	100.0	98.7	97.33
.8 1.	99.8	98.3	98.3	98.78	93.3	100.0	98.7	97.33
.7 1.	99.8	98.3	98.0	98.72	93.3	100.0	98.7	97.33
.6 1.	99.8	98.3	97.8	98.67	92.0	100.0	98.7	96.89
.5 1.	99.8	98.3	97.8	98.67	90.7	100.0	98.7	96.44
.4 1.	99.8	98.3	97.7	98.61	90.7	100.0	98.7	96.44
.3 1.	99.8	98.3	97.2	98.56	90.7	100.0	98.7	96.44
.2 1.	99.8	98.3	97.2	98.44	92.0	98.7	98.7	96.44
.1 1.	100.0	98.3	96.8	98.39	92.0	98.7	97.3	96.00
.0 1.	100.0	98.3	96.8	98.39	92.0	97.3	97.3	95.56
# of pixels	600	600	600	1800	75	75	75	225

The columns labeled s1 and s2 indicate the weights applied to sources 1 (s1) and 2 (s2).

CPU time for training and classification: 90 sec.

addition rather than multiplication in its global membership function. As compared to the 40-dimensional ML classification, these two methods were about 25% slower (Figure 4.17). It is worth noting that a ML classification of 60-dimensional data would have been still slower. Also, classification using the LOP and the SMC improved in terms of accuracy as compared to the ML classification of 40-dimensional data.

The CGBP neural network (720 input neurons, 20 hidden neurons) was trained to perfection for the 60-dimensional data (Tables 4.101 (training) and 4.102 (test)). In terms of accuracy of classification of test data, it was a little better than for the lower-dimensional cases. Also, a sample size of 300 or larger increased the overall accuracy for test data. The CGBP converged slowly. As with the other experiments its time to convergence grew rapidly with the number of training samples used. For 600 training samples per class, the algorithm converged in 3.65 CPU hours. The LOP and the SMC were 146 times faster. If compared to the 40-dimensional case, the CGBP was about 1.2 times slower in training and classification of the 60-dimensional data (Figure 4.18). In the 60-dimensional case the algorithm needed fewer iterations than for the 40-dimensional data.

As the dimensionality grew the CGLC (720 input neurons) did better in classification of training data (Table 4.103). In classification of test samples (Table 4.104), the CGLC was a little better than for the 40-dimensional data. The CGLC was about two times faster than the CGBP algorithm in classification of 60-dimensional data but the time to convergence also grew rapidly with the sample size. Oddly enough the training times for the 40 and 60 dimensions were almost the same for the CGLC with 300 training samples

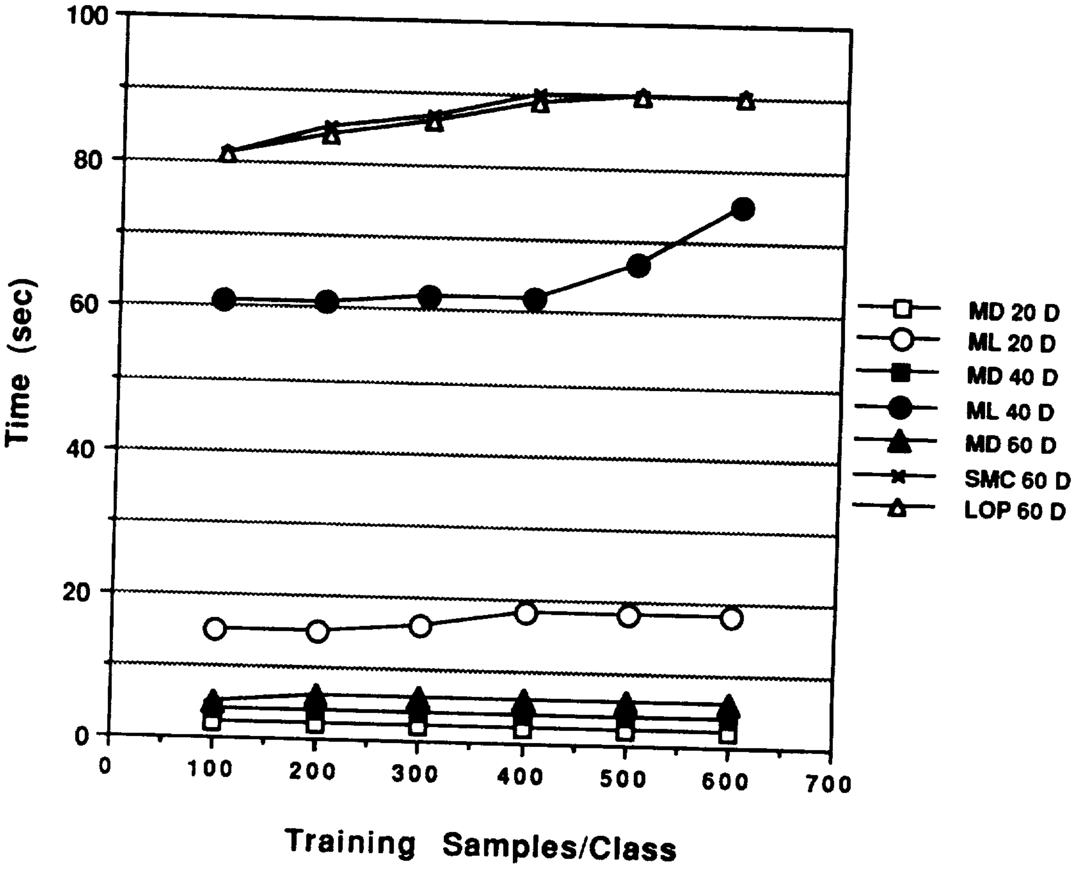


Figure 4.17 Statistical Methods: Training Plus Classification Time versus Training Sample Size

Table 4.101

Conjugate Gradient Backpropagation Applied to 60-Dimensional Simulated HIRIS Data: Training Samples.

Sample size	Number of iterations	CPU time	Percent Agreement with Reference for Class			
			1	2	3	OA
100	59	650	100.0	100.0	100.0	100.00
200	91	1921	100.0	100.0	100.0	100.00
300	183	4696	100.0	100.0	100.0	100.00
400	169	5969	100.0	100.0	100.0	100.00
500	172	7622	100.0	100.0	100.0	100.00
600	250	13174	100.0	100.0	100.0	100.00

Table 4.102

Conjugate Gradient Backpropagation Applied to 60-Dimensional Simulated HIRIS Data: Test Samples.

Sample size	Number of iterations	Percent Agreement with Reference for Class			
		1	2	3	OA
100	59	89.7	57.9	52.5	66.72
200	91	89.3	57.5	46.9	64.56
300	183	89.1	62.7	56.5	69.42
400	169	88.0	55.6	61.8	68.48
500	172	86.9	59.4	65.7	70.67
600	250	92.0	60.0	57.3	69.78

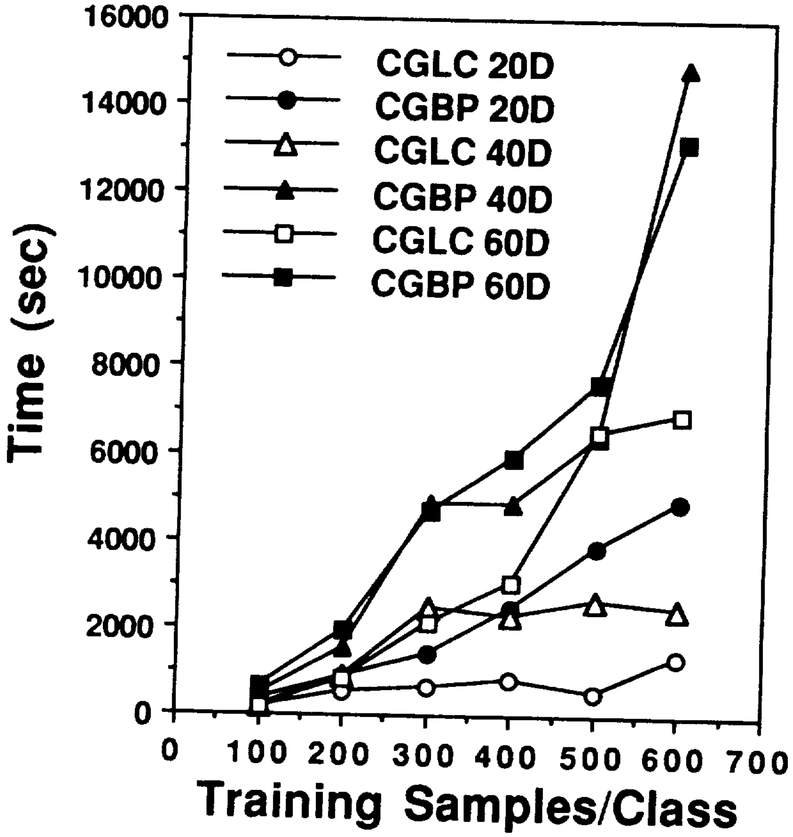


Figure 4.18 Neural Network Models: Training Plus Classification Time versus Training Sample Size

Table 4.103

Conjugate Gradient Linear Classifier Applied to 60-Dimensional Simulated HIRIS Data: Training Samples.

Sample size	Number of iterations	CPU time	Percent Agreement with Reference for Class 1	2	3	OA
100	102	201	100.0	100.0	100.0	100.00
200	246	843	100.0	100.0	100.0	100.00
300	517	2140	100.0	100.0	99.7	99.89
400	565	3030	100.0	100.0	100.0	100.00
500	1041	6511	100.0	99.8	99.4	99.73
600	857	6931	100.0	98.0	98.2	98.72

Table 4.104

Conjugate Gradient Linear Classifier Applied to 60-Dimensional Simulated HIRIS Data: Test Samples.

Sample size	Number of iterations	Percent Agreement with Reference for Class 1	2	3	OA
100	102	87.3	61.6	46.8	65.22
200	246	84.8	56.8	47.2	62.95
300	517	84.8	55.2	57.3	65.78
400	565	84.4	55.3	58.2	65.94
500	1041	86.9	61.7	56.6	68.38
600	857	88.0	66.7	58.7	71.11

or less. With a larger sample size, the 40-dimensional classification was about two times faster (Figure 4.18).

4.4.4 Summary

The Statistical methods were consistently superior to the neural network methods in the classifications of very-high-dimensional data performed here. The ML method, when applicable, was clearly the best, both fast and accurate, in classification of the 20- and 40-dimensional data sets. It could not be applied for the 60-dimensional data because of a singular covariance matrix. In that case the SMC and the LOP outperformed the minimum distance and neural network methods. In fact, these two methods must be considered desirable alternatives for classification of very-high-dimensional data. If the high-dimensional data can be split into two or more independent data sources, the SMC and the LOP can be very accurate and extremely fast. They are faster in classification than the ML method and can also be applied in classification of multitype data when the ML method is not appropriate. Also, in these experiments the LOP showed a far better performance than in the classifications of the multisource data in Sections 4.2 and 4.3. The apparent reason is that the two HIRIS data sources were rather agreeable. When this is the case the LOP can provide very good performance.

The MD classifier showed very poor performance. It is very fast but cannot discriminate the data adequately. Since it does not use any second order statistics, it is bound to perform poorly in classification of high-dimensional data [85]. Also, it shows saturation, i.e., above a certain number of dimensions its classification accuracy does not increase. In the experiments, the MD classification accuracy did not improve for data sets more complex than the 20-dimensional data.

Of the neural network methods applied, CGBP showed excellent performance in classification of training data. However, its classification accuracy for test data did not go much over 70%. The CGBP was very slow in training and increasing the number of training samples slowed the training process markedly. In contrast increasing the number of training samples did not significantly improve the classification accuracy of test data. It seems evident that CGBP needs to have seen almost every sample during training to be able to classify them correctly during testing.

Training of the CGBP is more efficient than conventional backpropagation and requires fewer parameter selections. However, as in the conventional backpropagation, the number of hidden neurons must be selected empirically. We selected the lowest number of hidden neurons which gave 100% accuracy during training. Use of too many hidden neurons makes the neural network computationally complex and can degrade its performance (analogous to the Hughes phenomenon [29]).

The CGLC uses no hidden neurons, and in the experiments with high-dimensional data it did not do much worse than the CGBP. The relatively good performance of the CGLC indicates good separability of the data. The CGLC was not as accurate as CGBP in classifying training data but achieved similar accuracies in classifying test data. The CGLC converged faster than CGBP, so it seems to be the better alternative for classification of very-high-dimensional data.

In defense of the neural network methods, it can be said that the maximum likelihood method had an unfair advantage since the simulated data were generated to be Gaussian. Neural networks are easy to implement and do not need any prior information about the data whereas a suitable statistical model has to be available for the ML method. Also, neural network methods were shown earlier to have potential in classifying difficult multitype data sets. However, the neural networks do not have as much ability to generalize as the statistical methods, which was evident in the test data results. These methods will not be comparable to the statistical methods in terms of speed unless implemented on parallel machines. Currently their computation time increases very rapidly with an increased number of training samples in contrast to the statistical methods which require almost no increased time when the training sample size increases.

CI:APTER 5

CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

5.1 Conclusions

This empirical evaluation of statistical methods and neural networks for classification of both multisource remote sensing/geographic data and very-high-dimensional data has revealed some striking differences.

The neural network models, the CGLC and the CGBP, showed good performance as pattern recognition methods for multisource remotely sensed data. Both neural networks were superior to the statistical methods used in terms of classification accuracy of training data. However, in classification of test data better results were achieved with statistical methods. Also, the neural network models have an overtraining problem. If their training procedure goes through *too many* learning cycles, the neural networks will get too specific in classifying the train ng data and give less than optimal results for test data. This overtraining problem is a shortcoming that has to be considered in the application of neural networks for classification.

The neural network models have the advantage that they are distribution-free and therefore no knowledge is needed about the underlying statistical distributions of the data. This is an obvious advantage over most statistical methods requiring modeling of the data, which is difficult when there is no prior knowledge of the distribution functions or when the data are non-Gaussian. It also avoids the problem of determining how much influence a source should have in the classification, which is necessary for both the SMC and LOP methods.

However, the neural networks, especially the CGBP, are computationally complex. When the sample size was large in the experiments, the training time could be very long. The experiments also showed how important the representation of the data is when using a neural network. To perform well the neural network models must be trained using representative training samples. Any trainable classifier needs to be trained using representative training samples but the neural networks are more sensitive to this than are the statistical methods. If the neural networks are trained with representative training samples the results showed that a two-layer or a three-layer net can do almost as well as the statistical methods in multisource classification of test samples. However, the neural network methods were clearly inferior to the statistical methods in the classification of the very-high-dimensional (simulated) HIRIS data. It was known beforehand that the HIRIS data were Gaussian; they were simulated that way. Therefore, the neural network methods did not have much chance of doing better than the statistical methods. The neural network models are more appropriate when the data are of multiple types and cannot be modeled by a convenient multivariate statistical model.

The SMC method worked well for combining multispectral and topographic data. The classification of four and six data sources gave significant improvement in overall and average classification accuracies as compared to single source classification. Using different levels of weights for different sources also showed promise in the experiments in terms of increase in overall classification accuracy.

Three different modeling methods were used in the experiments for density estimation of non-Gaussian data sources. The Parzen density estimation showed very good test performance in terms of overall classification accuracy. However, the Parzen density estimation was more time consuming than the other methods (histogram approach and maximum penalized likelihood method) when the sample size was large. The maximum penalized likelihood method also gave very good test accuracy. Both the Parzen density estimation and the maximum penalized likelihood method are useful alternatives for modeling of non-Gaussian data in multisource classification.

The SMC algorithm requires representative training samples but tends not to be as sensitive to their being representative as are the neural network models. The SMC algorithm outperformed the neural networks in classifying test data since it was provided with more prior knowledge in the form of the statistical model(s) for the data. Carefully modeled density functions make the statistical approach more capable of generalizing to samples not seen during training. Also, the neural network models require computationally expensive iterative training in contrast to the SMC algorithm. On the other hand, significantly more insight and effort are required on the part of the analyst to use the SMC. Also, when the Parzen density estimation is used with the SMC, the training time of the SMC can become computationally intensive in its own right.

The LOP did not do well at all in the multisource classification of multisource remote sensing and geographic data. The LOP is appealing because of its simplicity but it is not appropriate for classification of multisource data. It was clearly inferior to the SMC in classification of these data. However, in classification of the very-high-dimensional data, both the LOP and SMC algorithms showed excellent performance. Both methods were faster than the conventional ML classifiers and can always be used in contrast to the ML which shows singularity problems with limited number of training samples. The reason for the good performance of the LOP in the high-dimensional classification was that the two data sources were rather agreeable and had high source-specific accuracies. That was not the case for the sources in the multisource classification experiments. When the data sources are relatively agreeable the LOP can do well in classification and improve the overall accuracy as compared to the single source classifications.

The three suggested reliability measures were employed as ranking criteria for the data sources in the SMC and LOP classifications. These worked well for the SMC in all cases where sample sizes were adequate. The ranking criteria also worked well for the LOP in the classification of very-high-dimensional data. They could not help in classifications of multitype remote sensing and geographic data because the sources were not agreeable and the LOP tended toward dictatorship of the best source. It is very hard to determine the optimum weights for both the SMC and the LOP. That problem is still being investigated. With both optimum weighting and optimum data modeling the SMC will certainly give an excellent performance in classification of multisource remote sensing and geographic data.

In general the main advantage statistical classification algorithms have over the neural network models is that if the distribution functions of the information classes *are known* these methods can perform very accurately. But for those cases, as for instance in multisource classification, in which we do not know the distribution functions, neural network models can be more appropriate, although at considerable computational expense.

There are several problems related to both the statistical and neural network approaches in multisource classification which need further work. Suggestions for future research directions in this area are discussed next.

5.2 Future Research Directions

The most important problem with the statistical methods is weight selection. As observed previously, it is very hard to find optimum weights for the statistical multisource classifiers. One general approach for determining weights appears to be the use of optimization techniques similar to the mathematical programming methods suggested in Sections 2.3.4 and 2.3.5. These methods need more research to be applicable for optimum weight selection.

As discussed in Chapter 3, it is very difficult to implement statistics explicitly in neural networks. Therefore, it is very hard to combine the statistical consensus theory approaches and the neural networks models. However, one possibility for a *consensual neural network* is the *stage-wise* neural network algorithm described as follows. This network does not use prior statistical information but is somewhat analogous to the statistical consensus approaches.

In the stage-wise neural network a single-stage neural network is trained for a fixed number of iterations or until the training procedure converges. When training of the first stage has finished, the classification error for that stage is computed. Then another stage is created. The input data to the second stage are obtained by non-linearly transforming the original input vectors. The second stage is trained in a similar fashion to the first stage. When the training of the second stage has finished, the consensus from both stages is computed by taking the weighted sum (using stage-specific weights) of output activities from the stages. The stage-specific weights can, e.g., be selected based on the overall classification accuracies of each stage. Then the *consensual classification error* for the consensual neural network is computed using both stages. If the consensual classification error is lower than the classification error for the first stage, a new stage is created and trained in a similar way to the second stage, but with another set of non-linearly transformed input data. After training of this stage has finished, the consensus and the consensual error are computed for the output activities from all the stages.

Stages are added in the consensual neural network as long as the consensual classification error decreases. If the consensual classification error is not decreasing, the training is stopped. Testing can be done by applying all the stages in parallel.

The consensual neural network algorithm combines the information from various different "sources." In contrast to the data sources usually referred to in multisource classification, the "sources" here consist of non-linearly transformed data which have been transformed several times from the raw data. In neural networks it is very important to find the "best" representation of input data and the consensual neural network attempts to average over the results from several input representations. Also, in the consensual neural network, testing can be done in parallel between all the stages, which makes this method attractive for implementation on parallel machines.

This type of consensual neural network may be a desirable alternative for multisource classification. However, it needs further work in terms of guidance of weight-selection for the sources and selection of the best non-linear transformation.

LIST OF REFERENCES

[1] P.H. Swain, J.A. Richards and T. Lee, "Multisource Data Analysis in Remote Sensing and Geographic Information Processing," *Proceedings of the 11th International Symposium on Machine Processing of Remotely Sensed Data 1985*, West Lafayette, Indiana, pp. 211-217, June 1985.

[2] T. Lee, J.A. Richards and P.H. Swain, "Probabilistic and Evidential Approaches for Multisource Data Analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. GE-25, no. 3, pp. 283-293, May 1987.

[3] A.H. Strahler and N.A. Bryant, "Improving Forest Cover Classification Accuracy from Landsat by Incorporating Topographic Information," *Proceedings Twelfth International Symposium on Remote Sensing of the Environment*, Environmental Institute of Michigan, pp. 927-942, April 1978.

[4] J. Franklin, T.L. Logan, C.E. Woodcock and A.H. Strahler, "Coniferous Forest Classification and Inventory Using Landsat and Digital Terrain Data," *IEEE Transactions on Geoscience and Remote Sensing*, vol GE-25, no. 1, pp. 139-149, 1986.

[5] A.R. Jones, J.J. Settle and B.K. Wyatt, "Use of Digital Terrain Data in the Interpretation of SPOT-1 HRV Multispectral Imagery," *International Journal of Remote Sensing*, vol. 9, no. 4, pp. 669-682, 1988.

[6] C.F. Hutchinson, "Techniques for Combining Landsat and Ancillary Data for Digital Classification Improvement," *Photogrammetric Engineering and Remote Sensing*, vol. 48, no. 1, pp. 123-130, 1982.

[7] R.M. Hoffer, M.D. Fleming, L.A. Bartolucci, S.M. Davis, R.F. Nelson, *Digital Processing of Landsat MSS and Topographic Data to Improve Capabilities for Computerized Mapping of Forest Cover Types*, LARS Technical Report 011579, Laboratory for Applications of Remote Sensing in cooperation with Department of Forestry and Natural Resources, Purdue University, W. Lafayette, IN 47906, 1979.

[8] H. Kim and P.H. Swain, "Multisource Data Analysis in Remote Sensing and Geographic Information Systems Based on Shafer's Theory of Evidence," *Proceedings IGARSS '89*, IGARSS '89 12th Canadian Symposium on Remote Sensing, vol. 2, pp. 829-832, 1989.

[9] J.A. Richards, D.A. Landgrebe and P.H. Swain, "A Means for Utilizing Ancillary Information in Multispectral Classification," *Remote Sensing of Environment*, vol. 12, pp. 463-477, 1982.

[10] R.M. Hoffer and staff, "Computer-Aided Analysis of Skylab Multispectral Scanner Data in Mountainous Terrain for Land Use, Forestry, Water Resources and Geological Applications," *LARS Information Note 121275*, Laboratory for Applications of Remote Sensing, Purdue University, W. Lafayette, IN 47907, 1975.

[11] A. Torchinsky, *Real Variables*, Addison-Wesley Publishing Company, Redwood City, California, 1988.

[12] S. French, "Group Consensus Probability Distributions: A Critical Survey," in *Bayesian Statistics 2*, J.M. Bernardo, M.H. DeGroot, D.V. Lindley, A.F.M. Smith (eds.), North Holland, New York, New York, 1985.

[13] K.J. McConway, *The Combination of Experts' Opinions in Probability Assessment: Some Theoretical Considerations*, Ph.D. Thesis, University College, London, 1980.

[14] C.G. Wagner, "Allocation, Lehrer Models, and the Consensus of Probabilities," *Theory and Decision*, vol. 14, pp. 207-220, 1982.

[15] R.F. Bordley and R.W. Wolff, "On the Aggregation of Individual Probability Estimates," *Management Sciences*, vol. 27, pp. 959-964, 1981.

[16] K.J. McConway, "Marginalization and Linear Opinion Pools," *Journal of the American Statistical Association*, vol. 76, pp. 410-414, 1981.

[17] C. Berenstein, L.N. Kanal and D. Lavine, "Consensus Rules," in *Uncertainty in Artificial Intelligence*, L.N. Kanal and J.F. Lemmer (eds.), North Holland, New York, New York, 1986.

[18] M. Stone, "The Opinion Pool," *Annals Mathematical Statistics*, 32, pp. 1339-1342, 1961.

[19] N. Dalkey, *Studies in the Quality of Life*, Lexington Books, Lexington, MA, 1972.

[20] J.V. Zidek, *Multi-Bayesianity*, Technical Report no. 05, University of British Columbia, Vancouver, Canada, 1984.

[21] R.L. Winkler, "The Consensus of Subjective Probability Distributions", *Management Science*, vol. 15, no. 2, pp. B-61 - B-75, Oct. 1968.

[22] R.L. Winkler, "Combining Probability Distributions from Dependent Information Sources," *Management Sciences*, vol. 27, pp. 479-488, 1981.

[23] R.L. Winkler, "The Quantification of Judgement: Some Methodological Suggestions," *Journal of the American Statistical Association*, vol. 62, no. 320, pp. 1105-1120, 1967.

[24] R.L. Winkler, "Scoring Rules and the Evaluation of Probability Assessors," *Journal of the American Statistical Association*, vol. 64, pp. 1073-1078, 1969

[25] C. Genest and J.V. Zidek, "Combining Probability Distributions: A Critique and and Annotated Bibliography," *Statistical Science*, vol. 1., no. 1, pp. 114-118, 1986.

[26] M. Bacharach, *Bayesian Dialogues*, unpublished manuscript, Christ Church, Oxford, 1973.

[27] C. Genest, K.J. McConway and M.J. Schervish, "Characterization of Externally Bayesian Pooling Operators," *The Annals of Statistics*, vol. 14, no. 2, pp. 487-501, 1986.

[28] R.F Bordley, *Studies in Mathematical Group Decision Theory*, Ph.D. thesis, University of California, Berkeley, California, 1979.

[29] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd edition, Academic Press, New York, 1990.

[30] P.H. Swain, "Fundamentals of Pattern Recognition in Remote Sensing," in *Remote Sensing - The Quantitative Approach*, edited by P.H. Swain and S. Davis, McGraw-Hill Book Company, New York, 1978.

[31] J.A. Richards, *Remote Sensing Digital Image Analysis - An Introduction*, Springer-Verlag, Berlin, W. Germany, 1986.

[32] S.J. Whitsitt and D.A. Landgrebe, *Error Estimation and Separability Measures in Feature Selection for Multiclass Pattern Recognition,* LARS Publication 082377, The Laboratory for Application of Remote Sensing, Purdue University, W. Lafayette, Indiana, 1977.

[33] C.E. Shannon and W. Weaver, *The Mathematical Theory of Communication,* University of Illinois Press, Chicago, 1963.

[34] R.J. McEliece, *The Theory of Information and Coding,* Encyclopedia of Mathematics and its Applications, vol. 3, Addison-Wesley Publishing Company, Reading, Massachusetts, 1977.

[35] J.A. Benediktsson and P.H. Swain, *Methods for Multisource Data Analysis in Remote Sensing,* School of Electrical Engineering and Laboratory for Application of Remote Sensing, TR-EE 87-26, Purdue University, W. Lafayette, IN 47907, 1987.

[36] R.F. Bordley, "A Multiplicative Formula for Aggregating Probability Assessments," *Management Science,* vol. 28, pp. 1137-1148, 1982.

[37] D.H. Krantz, R.D. Luce, P. Suppes and A. Tversky, *Foundation of Measurement Vol. 1,* Academic Press, New York, New York, 1971.

[38] P.A. Morris, "An Axiomatic Approach to Expert Resolution," *Management Science,* vol. 29, pp. 24-32, 1983.

[39] D.V. Lindley, "Another Look at an Axiomatic Approach to Expert Resolution," *Management Science,* vol. 32, pp. 303-306, 1986.

[40] M.J. Schervish, "Comments on Some Axioms for Combining Expert Judgements," *Management Science,* vol. 32, pp. 306-312, 1986.

[41] M. H. DeGroot, "Reaching a Consensus," *Journal of the American Statistical Society,* vol. 69, no. 345, pp. 118-121, 1974.

[42] P.A. Morris, "Combining Expert Judgements, a Bayesian Approach" *Management Science,* vol. 23, pp. 679-693, 1977.

[43] J.O. Berger, *Statistical Decision Theory and Bayesian Analysis,* 2nd edition, Springer-Verlag, New York, 1985.

[44] S. French, "Updating of Belief in the Light of Someone Else's Opinion," *Journal of the Royal Statistical Society, Series A,* vol. 143, pp. 43-48, 1980.

[45] B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, Monographs on Statistics and Applied Probability, Chapman and Hall, New York, New York, 1986.

[46] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, A Wiley Interscience Publication, John Wiley and Sons, New York, New York, 1973.

[47] D.W. Scott, R.A. Tapia and J.R. Thompson, "Nonparametric Probability Density Estimation by Discrete Maximum Penalized-Likelihood Criteria," *The Annals of Statistics*, vol. 8, no.4, pp. 820-832, 1980.

[48] T. Kohonen, "An Introduction to Neural Computing," *Neural Networks*, vol. 1, no. 1, pp. 3-16, 1988.

[49] B.P. Lathi, *Modern Digital and Analog Communication Systems*, Holt, Rinehart and Winston, New York, 1983.

[50] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the brain," *Psychological Review*, vol. 65, pp. 386-408, 1958.

[51] R.A. Lippman, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, April 1987.

[52] J.J. Hopfield, "Neural Networks and Physical Systems with Emergen Collective Computational Properties Like Those of Two-State Neurons," *Proceeding of the National Academy of Science*, vol. 81, pp. 3088-3092, May 1984.

[53] S. Grossberg, "Nonlinear Neural Networks: Principles, Mechanisms, and Architectures," *Neural Networks*, vol. 1, no. 1, pp. 17-62, 1988.

[54] G.A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," *Neural Networks and Natural Intelligence*, S. Grossberg (ed.), MIT Press, Cambridge, MA, pp. 251-315, 1988.

[55] J.A. Hartigan, *Clustering Algorithms*, John Wiley and Sons, New York, 1975.

[56] T. Kohonen, *Self-Organization and Associative Memory*, 2nd edition, Springer-Verlag, New York, 1988.

[57] M. Minsky and S. Pappert, *Perceptrons* - Expanded Edition, MIT Press, Cambridge, MA, 1988.

[58] G.E. McClellan, R.N. DeWitt, T.H. Hemmer, L.N. Matheson and G.O. Moe, "Multispectral Image Processing with a Three-Layer Backpropagation Network," *Proceedings of IJCNN '89*, vol. 1, pp. 151-153, Washington D.C., 1989.

[59] S.E. Decatur, "Application of Neural Networks to Terrain Classification," *Proceedings of IJCNN '89*, vol 1., pp. 283-288, Washington D.C., 1989.

[60] S.E. Decatur, Application of Neural Networks to Terrain Classification, M.S. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 1989.

[61] O.K. Ersoy and D. Hong, "A Hierarchical Neural Network Involving Nonlinear Spectral Processing," presented at IJCNN '89, Washington D.C., 1989.

[62] P.H. Heermann and N. Khazenie, "Application of Neural Networks for Classification of Multi-Source Multi-Spectral Remote Sensing Data," *Proceedings of IGARSS '90*, vol. 2, pp. 1273-1276, Washington D.C., 1990.

[63] J. Maslanik, J. Key and A. Schweiger, "Neural Network Identification of Sea-Ice Seasons in Passive Microwave Data," *Proceedings of IGARSS '90*, vol. 2, pp. 1281-1284, Washington D.C., 1990.

[64] B. Widrow and M.E. Hoff, "Adaptive Switching Circuits," *1960 IRE WESCON Convention Record*, IRE, pp. 96-104, New York, 1960.

[65] J.A. Anderson and E. Rosenfeld (eds.), *Neurocomputing*, MIT Press, Cambridge, MA, 1988.

[66] P.J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*," Ph.D. thesis, Harvard University, Cambridge, MA, 1974.

[67] D. Parker, *Learning Logic*, Technical report TR-87, Center for Computational Research in Economics and Management Science, MIT, Cambridge, MA, 1985.

[68] Y. Le Cun, "Learning Processes in an Asymmetric Threshold Network," *Disordered Systems and Biological Organization*, E. Bienenstock, F. Fogelman Souli and G. Weisbruch (eds.), Springer, Berlin, 1986.

[69] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning Internal Representation by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Vol. 1, D.E. Rumelhart and J.L. McClelland (eds.), pp. 318-362, MIT Press, Cambridge, MA, 1986.

[70] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning Representations by Back-propagating Errors," *Nature*, 323, pp. 533-536, 1986.

[71] R.A. Jacobs, "Increased Rates of Convergence Through Learning Rate Adaption," *Neural Networks*, vol. 1, no. 4, pp. 295-307, 1988.

[72] J.A. Benediktsson, P.H. Swain and O.K. Ersoy, "Neural Network Approaches Versus Statistical Methods in Classification of Multisource Remote Sensing Data," *Proceedings IGARSS '89 and the 12th Canadian Symposium on Remote Sensing*, vol. 2, pp. 489-492, Vancouver, Canada, 1989.

[73] R.P. Gorman and T.J. Sejnowski, "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Signals", *Neural Networks*, vol. 1, no. 1, pp. 75-90, 1988.

[74] E. Barnard and R.A. Cole, *A Neural-Net Training Program Based on Conjugate-Gradient Optimization*, Technical Report No. CSE 89-014, Oregon Graduate Center, July 1989.

[75] R.L. Watrous, *Learning Algorithms for Connectionis Networks: Applied Gradient Methods of Nonlinear Optimization*, Technical Report MS-CIS-88-62, LINC LAB 124, University of Pennsylvania, 1988.

[76] D.G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed., Addison-Wesley, Reading, MA, 1984.

[77] H. White, "Learning in Artificial Neural Networks: A Statistical Perspective," *Neural Computation*, vol.1, pp. 425-464, 1989.

[78] W. Kan and I. Aleksander, "A Probabilistic Logic Neuron Network for Associative Learning," in *Neural Computing Architectures - The Design of Brain-Like Machines*, edited by I. Aleksander, M.I.T. Press, Cambridge, MA, 1989.

[79] D.F. Specht, "Probabilistic Neural Network (PNN)", *Proceedings of ICNN*, San Diego, 1988.

[80] D.F. Specht, "Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 111 - 121, March 1990.

[81] O.K. Ersoy, Lecture Notes, School of Electrical Engineering, Purdue University, 1990.

[82] J.A. Benediktsson, P.H. Swain and O.K. Ersoy, "Neural Network Approaches Versus Statistical Methods in Classification of Multisource Remote Sensing Data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. GE-28, no. 4, pp. 540-552, July 1990.

[83] D.G. Goodenough, M. Goldberg, G. Plunkett and J. Zelek, "The CCRS SAR/MSS Anderson River Data Set," *IEEE Transactions on Geoscience and Remote Sensing*, vol. GE-25, no. 3, pp. 360-367, May 1987.

[84] J.P. Kerekes and D.A. Landgrebe, *RSSIM: A Simulation Program for Optical Remote Sensing Systems*, TR-EE 89-48, School of Electrical Engineering, Purdue University, West Lafayette, IN, August 1989.

[85] C. Lee, *Classification Algorithms for High Dimensional Data*, Ph.D. thesis proposal, School of Electrical Engineering, Purdue University, 1989.