

Using Output of Linear Programming-based models

Venkat Chalasani
SRA International
Fairfax, VA, 22033 U.S.A.

Abstract

Among classification problems multi-modal problems form an important category. They naturally arise in many fields when a single class can occupy disjoint areas in the feature space. They are particularly hard to solve for traditional classification techniques. Decision tree Classifiers(DTCs) in principle are able to solve such problems by the recursive nature of their algorithm. In principle, DTCs should perform well on multi-modal data sets because they have the ability to partition each class into disjoint areas. In practice, however, DTCs can have significant difficulty with such data sets. In the presence of multimodality decision trees can become very bushy leading to high error rates. Clustering of the data before classification can lead to simpler trees and lower error rates. We apply linear programming based decision trees to multimodal problems. We present a technique that receives input from the discrimination to alter the feature space for clustering in such a way as to improve classification accuracy.

Keywords : classification, clustering, linear programming.

1 Introduction

Classification is a supervised learning process. Given a set of n measurement vectors $\mathbf{x} = \{x_1, x_2, \dots, x_f\}$ in f -dimensional euclidean space \mathbf{X} with class membership assigned to each vector from a set of classes $C = \{c_1, c_2, \dots, c_n\}$ a classification algorithm $D(\mathbf{x})$ defines a mapping $\mathbf{X} \rightarrow C$. In other words the classification algorithm partitions \mathbf{X} so that any measurement vector can be assigned a class label depending on its location in the euclidean space. The coordinates x_1, x_2, \dots, x_n of the measurement vector are commonly referred to as features. Consider for example, a medical diagnostic classification problem, in which there are two classes, Class 1 consists of patients at high risk of

having a heart attack, and Class 2 consists of patients at low risk of having a heart attack. In this example x_1 could refer to a patient's blood pressure and x_2 could refer to the patient's cholesterol level, and so on. We will refer to the f -dimensional euclidean space \mathbf{X} as the feature space and to a measurement vector \mathbf{x} as a data point or an instance. Classification algorithms are rarely completely accurate in practice. One typically associates a cost with misclassification of an instance. Our objective in classification is to minimize such costs.

2 Multi-modal Problems

The motivation for the current work arises out of a set of classification problems known as multi-modal problems. Multi-modal problems are among the most difficult classification problems, and present special difficulty for traditional statistical techniques. Multi-modal problems arise when the classes of a classification problem occupy disjoint areas in feature space. Problems of this type are important in a variety of applications. In sensor fusion, changes in spatial or temporal conditions can give rise to disjoint measurements from the same sensor. Similarly, in emitter classification the emitter can change its signature by simply changing a setting. Consider for example, the Radar dataset consisting of operating characteristic data for several types of radar [4]. The associated classification problem is to classify a radar by type given its operating characteristics. Each radar can operate on 5 different settings resulting in as many as five modes in the feature space. In Figure 1 we show data on the Frequency-Pulse Repetition interval for the first two classes. Five modes of each class are apparent.

Multi-modal problems also arise in data sets which have one or more integer valued attributes. Our third example problem, the Digits problem, is a fundamental task in numeral or digit recogni-

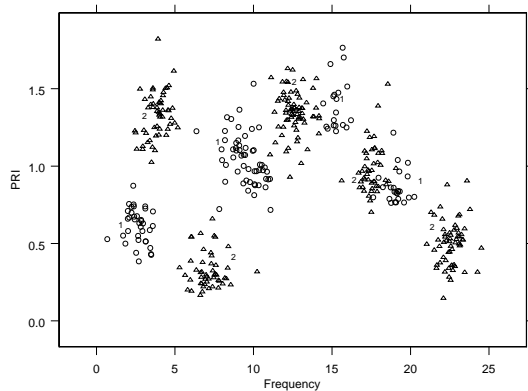


Figure 1: Plot of Radar data

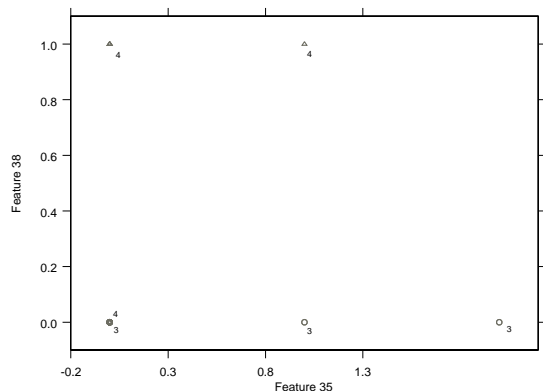


Figure 2: Plot of Digits data

tion. The data consists of 37 integer-valued features (in contrast with the real-valued features of the emitter data sets). Originally obtained from LACIS-ITEPEA laboratory at University of Rouen, France, the digits data consists of attributes of digits obtained from 45 license plates. The attributes extracted simulate attentive and preattentive aspects of human vision. A total of 161 observations of 10 digits were collected. Figure 2 shows classes 3 and 4 of the digits data set on features 35 and 38.

One of the simpler multi-modal problems is the parity data. We generated a 2-dimensional parity data set by using normal distribution at each of the vertices. The data consists of 10,000 data points equally distributed at the four vertices $(0,0,0,1,1,0,1,1)$. The data was generated using a standard deviation of 0.175 and the two variables were treated independently. The four vertices are marked separately in the figure (1 and 2 represent even parity and 3 and 4 represent odd parity).

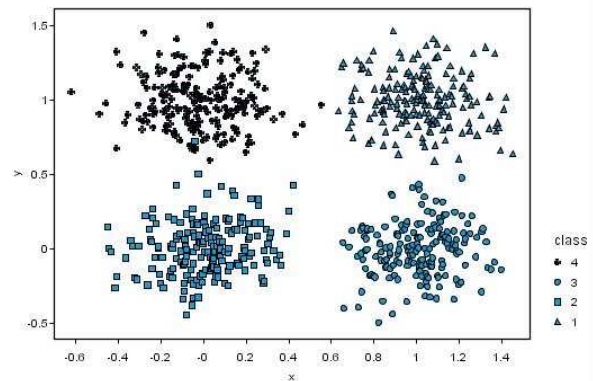


Figure 3: Two dimensional Parity data

3 The Splitting Problem

DTCs represent a divide and conquer strategy for the classification problem. The feature space is recursively partitioned into two or more mutually exclusive and exhaustive sub-partitions till each of those regions can be assigned a unique class label. The partitioning of the feature space is carried out by using split functions. As such choice of the split function is the single most important aspect of the decision tree creation.

Depending on the number of sub-partitions, the decision tree can be binary with two partitions, tertiary with three partitions etc. Since any ordered decision tree can be represented using a binary tree [22], we will limit ourselves to binary decision trees. In binary decision trees, each partition is divided into two sub-partitions using a boolean split function such as “Is $x_1 > 0$?”, which we write simply as “ $x_1 > 0$ ”. One partition represents the part of the feature space with “ $x_1 > 0$ ” and the other “ $x_1 \leq 0$ ”. Each of these partitions can again be divided using another split function and so on.

The splits can be univariate or multivariate. In addition they can be linear or non-linear. Most of the commonly used decision trees use univariate splits. The creation of univariate splits is achieved by deciding on an optimization criterion, the *goodness of a split*. Goodness of split (gs) is a criterion which judges the merit of a split and is used to choose from a set of available splits.

4 Multi-variate splits

DTCs that use univariate splits, evaluate a large number of alternative splits and choose one of them based on a gs criterion. CART for example, uses the Gini criterion (it can also use the Twoing criterion) to judge the quality of a split. For each

feature, CART considers a split placed midway between every two data points. The split which maximizes the Gini criterion is then chosen as the current split. This criterion works well with univariate splits because all possible univariate splits with a specific purity-gs can be exhaustively enumerated and evaluated. For such algorithms the splitting problem consists of:

1. Deciding on the criterion that will be used to judge the quality of a split. This is the goodness of split criterion.
2. Choosing one split, among all possible splits, based on the goodness of split criterion.

Univariate splits have the advantage of being simple and easy to understand, and they can work well on many problems. On problems with linear or higher order structure, however, univariate-split DTCs may be forced to use many univariate splits to accomplish the separation that could be achieved with a single multivariate split [3].

Although the definition of multivariate splits also includes non-linear combinations, we will restrict the following discussion to linear splits or hyperplanes. The number of possible univariate splits for a problem domain with f features and n data points is nf in the worst case. Whereas there exist a maximum of $2^{\binom{f}{n}}$ distinct multivariate splits that can divide the set into two non-overlapping subsets [20]. Utgoff and Brodley carried out a comprehensive study of four algorithms used for creating multivariate splits [3]: Recursive Least Squares (RLS) [29], Pocket Algorithm (PA) [13], the Thermal Training Procedure (TTP) [11] and CART. All algorithms in this study, excluding CART, use error-gs. RLS minimizes the mean squared error over the training data, PA maximizes the number of correct classifications of the training data and TTP uses a simulated annealing like training procedure for learning the coefficients of a linear machine. RLS and CART produce binary splits, whereas PA and TTP produce multi-way splits. One approach to creating multivariate splits is to employ linear programming. The earliest linear programming literature includes a convex hull approach by Kendall [16], Magnasarian's LP approach [19] and Smith's formulation [24]. The early approaches were restricted to regression-type procedures and did not find wide acceptance. Freed and Glover [9] formalized the convex hull procedure to solve a discrimination problem.

5 Linear Programming for Splits

The split functions in a decision tree can be created using an LP formulation that creates a linear

discriminant in feature space between two groups of data, which we will label 1 and 2. Let n_k denote the number of data points associated with group k in the training set, let X_{ij}^k represent the value of j th attribute of the i th instance of group k , and let the number of features be F . We formulate the following linear program:

$$\min \quad 1/n_1 \sum_{i=1}^{n_1} y_i + 1/n_2 \sum_{i=1}^{n_2} z_i$$

s. t.

$$\begin{aligned} \sum_{j=1}^F X_{ij} w_j + y_i &\geq \gamma + 1 && \text{for } i = 1, 2, \dots, n_1 \\ \sum_{j=1}^F X_{ij} w_j - z_i &\leq \gamma - 1 && \text{for } i = 1, 2, \dots, n_2 \\ y_i &\geq 0 && \text{for } i = 1, 2, \dots, n_1 \\ z_i &\geq 0 && \text{for } i = 1, 2, \dots, n_2 \end{aligned}$$

The variables in this formulation are w_i , y_i , z_i , and γ . The variables w_i and γ define a linear hypersurface that separates the observations of group 1 from those of group 2. If the two groups cannot be cleanly separated by a linear hypersurface, some of the variables y_i , z_i may be forced to take on positive values. The objective is to minimize a weighted average of the errors. This formulation is based on Bennet's linear programming formulation [2].

6 Clustering

Clustering is a technique for identifying structure in data. In the artificial intelligence literature, the term 'unsupervised classification' is generally used, whereas in psychology, it is referred to as Q-analysis. Clustering is sometimes also referred to as clumping or grouping.

Definition 1 (Clustering) : *Clustering is the process of a useful division of a collection of n objects having f attributes into groups, where both the number and properties of the groups are to be determined [7].*

It is not always clear how clusters should be organized but inherent in the process is the idea that the objects placed into a single group are similar to each other in some respects and different from objects in other groups. Cormack and Gordon use 'internal cohesion' and 'external isolation' as properties which would define clusters [5],[14], whereas other authors suggest that usefulness and the investigator's value judgment are all that is required to define clusters. If the data is two dimensional, clusters may be visually identified by plotting the data points in what is known as a 'scattergram'.

Multi-dimensional data can be visualized as a series of two dimensional images by using two variables at one time . Non Linear Mapping (NLM) is another technique that can be used to visualize multi dimensional data [23]. The technique is based on mapping from a higher dimensional space to a lower dimensional space such that the pairwise distances between the datapoints are preserved. We used NLM to map the iris data set containing the four dimensions(widths and lengths of the petals and sepals) of three varieties of Iris flowers (Setosa, Virginica and Versicolour) using NLM.

Everitt broadly divides clustering techniques into hierarchical techniques, optimization methods, mixture models and other techniques. Here we consider only hierarchical clustering techniques.

Hierarchical techniques form an important class of clustering techniques. In hierarchical clustering, the data are not partitioned into a particular number of clusters in a single step. Instead, the process consists of creation of a series of partitions which may run from a single cluster containing all the data points to n clusters containing individual data points [7]. Hierarchical clustering techniques can be divided into agglomerative techniques and divisive techniques. Agglomerative techniques start with individual data points as groups and fuse them successively to end up with a single group. Divisive techniques on the other hand, start by treating all the data points as a single group and then break this group down into individual subgroups. In this work we use agglomerative techniques, which are more common than divisive techniques.

The most important aspect of the clustering process is the assessment of relative distances between data points. We will use the term *similarity* as a measure of closeness of two data points, and *dissimilarity* to mean the opposite of similarity. If a dissimilarity measure fulfills the metric property $d_{ij} + d_{ik} \geq d_{jk}$ for all i, j, k , it is generally known as a distance measure.

The basic operation in all agglomerative techniques is similar. At each stage, select two individuals or groups to join which are closest based on a defined distance measure. They differ only in the way the distance is defined. The distance between a group k and a group (ij) , formed by the fusion of the groups i and j , is given by

$$d_{k_{(ij)}} = \alpha_i d_{ki} + \alpha_j d_{kj} + \beta d_{ij} + \gamma |d_{ki} - d_{kj}|,$$

where d_{ij} is the distance between groups i and j . The most common agglomerative techniques are defined through the choice of the parameters α_i , α_j , β , and γ , and through the definition of d . Single linkage clustering, for example, corresponds to the parameter values $\alpha_i = \alpha_j = 1/2$, $\beta = 0$, and

$\gamma = -1/2$. For single linkage, complete linkage, and group average, d_{ij} can represent either distance or similarity. Centroid clustering can be incorporated into the scheme with d_{ij} as the Euclidean distance. Wishart showed that this scheme can also accommodate Ward's method if d_{ij} represents squared Euclidean distance [28].

7 Distance Function

Clustering is an effective pre-processing step for identifying groups in multi-modal classification problems. In our previous work we have shown that clustering can improve the performance of CART [27]. It has also been shown that clustering can significantly improve classification accuracy of LP-based decision tree classifiers [27].

The distance measure most commonly used in clustering is the euclidean distance. Euclidean distance weighs all dimensions equally. In many mutivariate problems some of the variables can be noisy and provide very less information for discrimination. There can also be scaling issues for the variables. Variables that have higher means can dominate the distance function even if they have less discriminative power as compared to variables that have smaller means. The most common method employed in such situations is to use scaling. The most common method is 0–1 linear scaling with the minimum of the range mapped to 0 and the maximum of the range to 1. In some situations a Z score calculated by subtracting the mean and dividing by standard deviation is an effective transformation.

8 Feature Selection and Detection and Removal of Outliers

The sum of deviation models have been judged to be the best LP models in terms of performance [1], [10]. Minimum Sum of Deviations(MSD) LP models however, tend to use all the variables in the model and are sensitive to outliers. Detecting and removing outliers can improve the accuracy of such models. One way of approaching this problem is to look at the total distance between the groups instead of just the positive deviations. An empirical measure, the contribution of each variable to the total interclass distance can be used as the measure of effectiveness of that feature. The interclass distance has some similarity with the Mahalanobis distance as a measure of separation between two groups. We illustrate this idea further with an example. We use bankruptcy data [15] used by Nath and Jones in their jackknife based approach for feature selection for LP discriminant analysis [21]. This data, which

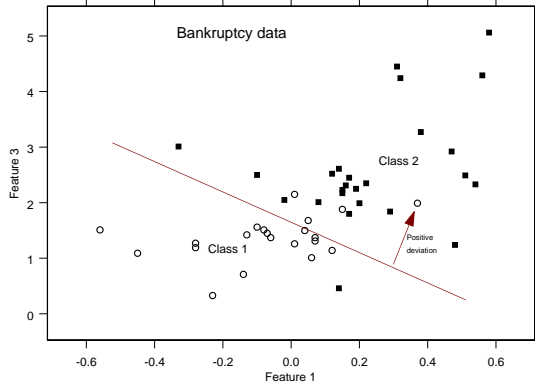


Figure 4: Plot of Bankruptcy data on f1-f3 plane

consists of financial ratios collected approximately two years prior to bankruptcy, has 4 ratio features:

1. cash-flow/total-debt;
2. net income/total assets;
3. current assets/current liabilities; and
4. current assets/net sales.

Nath and Jones used the following MSD model proposed by Bajgier and Hill [1] and Freed and Glover [10]:

$$\min \sum_{i=1}^{n_1+n_2} y_i$$

subject to

$$\begin{aligned} \sum_{j=1}^F X_{ij}w_j + y_i &\geq \gamma \quad i = 1, 2, \dots, n_1 \\ \sum_{j=1}^F X_{ij}w_j - y_i &\leq \gamma \quad i = n_1 + 1, n_1 + 2, \dots, n_1 + n_2 \\ y_i &\geq 0 \quad i = 1, 2, \dots, n_1 + n_2 \end{aligned} \quad (1)$$

Nath and Jones obtained the following classification rule: A firm belongs to class 1 bankrupt if $12.91X_1 - 26.89X_2 + 5.41X_3 - 2.49X_4 < 10$ and to class 2 or financially sound otherwise.

Referring to the MSD model used by Nath and Jones [21], the contribution of feature j can be defined as:

$$\sum_{i=1}^{n_1} X_{ij}w_j - \sum_{i=n_1+1}^{n_2} X_{ij}w_j \quad (2)$$

Note that in this expression, the first part is contributed by class 1 and the second part by class 2 [26].

Contribution of variables to interclass separation				
Contribution	X1	X2	X3	X4
class 1	18.46	-45.98	-155.27	22.91
class 2	-75.91	37.38	-350.78	26.59
Total	94.37	-83.36	195.52	-3.69

Table 1: Contribution of Variables to Interclass Separation

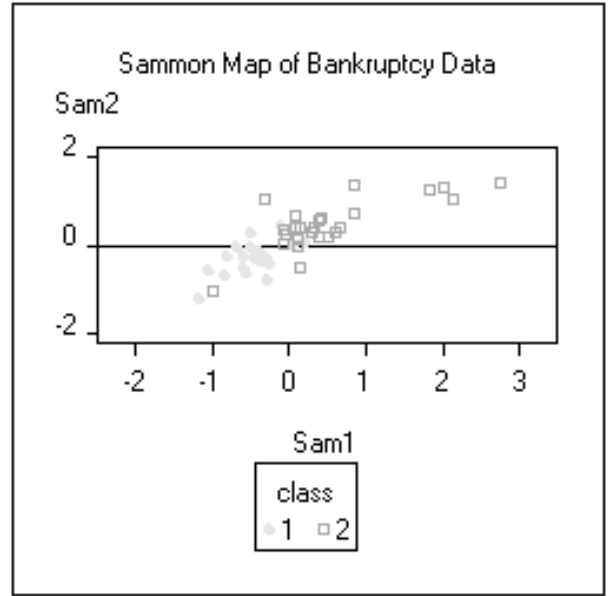


Figure 5: Sammon Map of Bankruptcy Data

8.1 Revising Linear Program

The Sammon Map of the bankruptcy data shows that the overlap between the classes is small. However, the model misclassifies about six points. The reason is the presence of an extreme outlier as seen in the picture. Comparing this picture to the one with only features 1 and 3 we can see that there must be some information value in features 2 and 4. However, a strong negative contribution of X_2 indicates that it was more influenced by outliers and caused loss of accuracy for the model. We ran our Linear Program after removing the outlier as well as variable X_2 and obtained a solution in which only two points were misclassified.

8.2 Revising Clustering

As remarked earlier clustering algorithms typically use euclidean distance giving equal importance to all features noisy or otherwise. We can use the output from a linear program to transform the

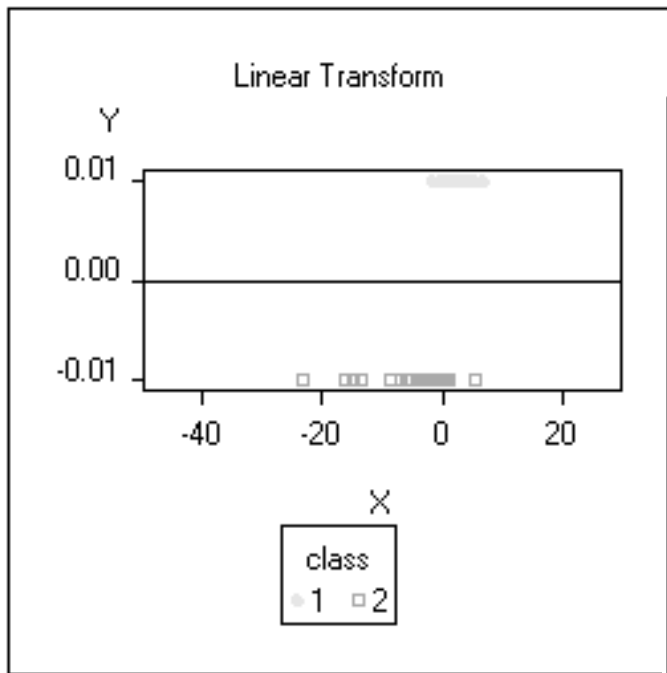


Figure 6: Transform of Bankruptcy Data to a Line

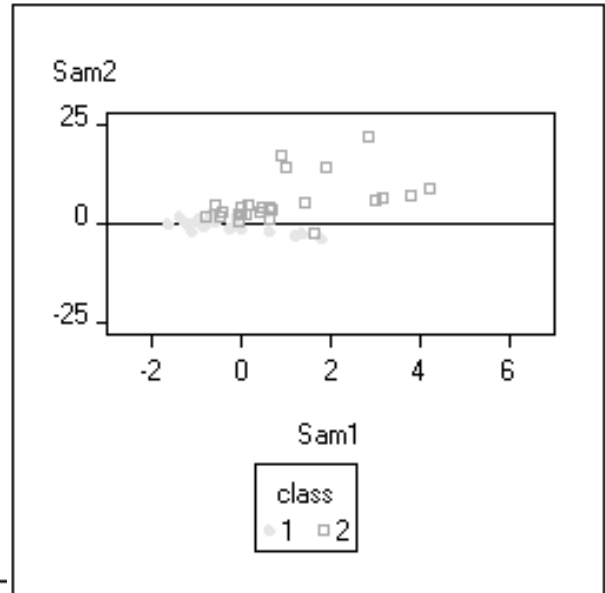


Figure 7: Sammon Map with Discriminant Information

feature space. Any linear discriminant can also be used as a linear transformation [18]. The criterion from which the linear discriminant has been derived decides the conditions that the linear transformation will satisfy. The linear transformation line is perpendicular to the discriminant plane from which it is derived. Therefore the process of finding a separating plane that minimizes the sum of deviations is equivalent to finding a linear transformation and a point on that line that minimizes the distance of misclassified points (points falling on the wrong side of the plane). The linear transformation of the bankruptcy data is shown in Figure 6. To show the picture clearly points of class 1 have been given a slight positive y value and those of class 2 a slight negative y value.

Clustering can possibly be improved by adding information from this transform to the original features after feature selection. We first add the discriminant value as a feature Figure 7 and recreate the picture after scaling the discriminant value to a 0 – 1 range. It is clear that unscaled discriminant value distorts the feature space to a large extent. The scaled discriminant value might be useful in changing the feature space for other algorithms Figure 8.

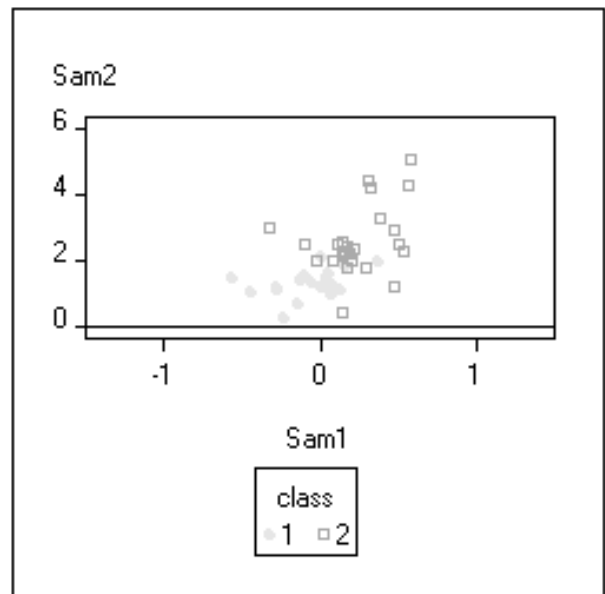


Figure 8: Sammon Map with Discriminant Information Scaled

9 Conclusions and directions for further research

We used the output of an LP model to alter the feature space for clustering. We are in the process of showing that this can improve the performance of LP-based decision trees. We are at present further experimenting with our approach for feature selection in Decision Tree Classifiers based on Linear Programming. To make our feature selection technique approach practical, we will also have to address combining features and feature weights obtained from multiple LP discriminants into a single function. Most practical problems are multicategory problems and more than one LP discriminant is needed for separation of the classes.

References

- [1] S. M. Bajgier and A. V. Hill, "An experimental comparison of statistical and linear programming approaches to the discriminant problem", *Decision Sciences*, 13, 1982, pp.604-618.
- [2] K. P. Bennett, "Machine Learning via Mathematical Programming", Doctoral dissertation, Computer Sciences Department, University of Wisconsin-Madison, 1993.
- [3] C. E. Brodley and P. E. Utgoff, "Multivariate Decision Trees", *Machine Learning*, No. 19, pp. 45-77, 1995.
- [4] D. Brown, V. Corruble and C. Pittard, "A Comparison of Decision Tree Classifiers with Backpropagation Neural Network for Multimodal Classification Problems", *Pattern Recognition*, Vol. 26, pp. 953-961, 1993.
- [5] R. M. Cormack, "A review of classification", *J. Roy. Statist. Soc. A* 134, pp. 321-367, 1971.
- [6] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York 1973.
- [7] B. Everitt, *Cluster Analysis*, Edward Arnold, 1993.
- [8] D. H. Foley and J. W. Sammon, "An Optimal Set of Discriminant Vectors", *IEEE Transactions on Computers*, March 1975.
- [9] N. Freed and F. Glover, "A Linear Programming Approach to the Discriminant Problem", *Decision Sciences*, Vol. 12, 1981.
- [10] N. Freed and F. Glover, "Evaluating alternative linear programming models to solve the two-group discriminant problem", *Decision Sciences*, Vol. 17, 1986, pp. 151-162.
- [11] M. Frean, "Small nets and short paths: optimising neural computation", Doctoral dissertation, Center for Cognitive Science, University of Edinburgh, 1990.
- [12] K. Fukunaga and L. G. Koontz, "Application of the Karhunen-Loeve Expansion to Feature Selection and Ordering", *IEEE Transactions on computers*, April 1970.
- [13] S. I. Gallant, "Optimal Linear Discriminants", *Proceedings of the International Conference on Pattern Recognition*, pp. 849-852, IEEE Computer Society Press.
- [14] A. D. Gordon, *Classification*, Chapman and Hall, London 1980.
- [15] R. A. Johnson, and D. W. Wichern, *Applied multivariate statistical analysis*, Prentice Hall, Englewood Cliffs, NJ, 1982.
- [16] M. G. Kendall, "Discrimination and Classification", in P. R. Krishnaiah(Ed.) *Multivariate Analysis*. New York: Academic Press, 1966.
- [17] J. Kittler, "Feature Selection and Extraction", in Young, T. Y. and Fu, King-Sun (Ed.) *Handbook of Pattern Recognition and Image processing*, New York: Academic Press, 1986.
- [18] C. Lee, and D. L. Landgrebe, "Feature Extraction Based on Decision Boundaries", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 4, 1993.
- [19] O. L. Magnasarian, "Multisurface Method of Pattern Separation", *IEEE Transactions on Information Theory*, Vol IT-14, No. 6, pp.801-807, 1968.
- [20] S. K. Murthy, S. Kasif and S. Salzberg, "A System for Induction of Oblique Decision Trees", *Journal of Artificial Intelligence Research*, pp. 1-32, 1994.
- [21] R. Nath, and T. W. Jones, "A Variable Selection Criterion in the Linear Programming Approaches to Discriminant Analysis", *Decision Sciences*, Vol. 19, 1988.
- [22] E. Rounds, "A combined non-parametric approach to feature selection and binary decision tree design", *Pattern Recognition*, Vol. 12, pp. 313-317, 1980.
- [23] J. W. Sammon, "A Nonlinear Mapping for Data Structure Analysis", *IEEE Transactions on Computers*, Vol. c-18, pp. 401-407, 1969.
- [24] F. W. Smith, "Pattern Classifier Design by Linear Programming", *IEEE Transactions on Computers*, Vol. C-17, pp. 367-372, 1968.
- [25] V. Chalasani and P. Beling "Contribution-based approach for feature selection in Linear Programming-based models", *Proceedings of IEEE International Conference on Systems Man and Cybernetics*, Nashville, TN, 2000.
- [26] V. Chalasani and P. Beling, "Optimization based Decision Trees for Multi-modal Problems", *Proceedings of IEEE International Conference on Systems Man and Cybernetics*, Nashville, TN, 2000.
- [27] V. Chalasani, "Improving Decision Trees by Clustering", *Proceedings of MLMTA*, Las Vegas, NV, 2004.
- [28] D. Wishart, "An Algorithm for Hierarchical Classifications", *Biometrics*, Vol. 25, pp. 97-98, 1969.