MULTI-HYPERPLANE FORMULATIONS FOR CLASSIFICATION AND DISCRIMINATION ANALYSIS

Marco Better, Leeds School of Business, University of Colorado at Boulder, Campus Box 419, Boulder, Colorado 80309, marco.better@colorado.edu
Fred Glover, University of Colorado at Boulder, Campus Box 419, Boulder, Colorado 80309, fred.glover@colorado.edu
Michele Samorani, DEIS - Università degli Studi di Bologna, Bologna, Italy, michele.samorani@libero.it

ABSTRACT

We introduce a new mixed integer programming formulation for the two-group classification and discrimination problem that makes use of multiple separating hyperplanes. Our work constitutes an innovation in the area of support vector machines in the context of *successive perfect separation* decision trees, by constructing a discriminant function that approximates complex non-linear decision boundaries without the need for kernel transformations of the data. Unlike piecewise-linear formulations, our model does not require that one of the groups belong to a convex region, making our formulation more effective for complex data sets. We conduct a computational study using well known data sets in banking and in cancer detection that shows the merit of our model.

INTRODUCTION

Discrimination analysis is one of the key tools for classifying data in real world data mining, and finds important uses in settings that range from bioinformatics to healthcare, and from financial analysis to military planning (see [2], [7], [8], and [9]). Recent contributions have been provided by mathematical programming (MP) approaches based on piecewise-linear models that approximate non-linear boundaries, or on kernel transformations that attempt to render the data linearly separable.

On one hand, piecewise-linear models make use of stringent requirements that constrain their ability to handle complex structures, and fail to scale up efficiently to large data sets. For example, these models require that the elements of one of the two groups lie entirely in a convex region, so they must be solved twice: once to constrain all of Group 1 elements to a convex region, and once to do so for Group 2 elements [3]. On the other hand, kernel-based methods like Support Vector Machines (SVM) as in [1] usually rely on a limited number of known kernel transformations to project the original data into very high-dimensional space in the hope of rendering it linearly separable.

In this paper we present a mixed integer formulation that generates multiple hyperplanes simultaneously, thus forming a type of decision tree structure that does away with the convexity requirements in piecewise-linear models with significantly fewer variables and constraints. In addition, instead of relying on kernel transformations, our approach approximates a non-linear discriminant function that seeks to separate the original data directly. This affords access to the best of both worlds, for our approach can readily exploit a kernel transformation in a case where one is known to be relevant to the application at hand.

THE CLASSIFICATION AND DISCRIMINATION PROBLEM

Let a_{ij} denote the value of attributes of the elements in a data set, where each element i $(i=1,...,m)$ is described by attribute j $(j=1,...,n)$. We seek a decision rule to correctly identify whether a given vector $A_i=(a_{i1},...,a_{in})$ should belong among the elements of Group 1 or among those of Group 2 (G_1 and G_2, respectively). For instance, the elements A_i may refer to credit applications we seek to correctly classify according to whether they involve "good" risk ($i \in G_1$) or "bad" risk ($i \in G_2$); the first component a_{i1} of A_i may refer to the applicant's age, the second component a_{i2} may refer to the applicant's annual income, and so on. Given the knowledge of the A_i vectors and their group membership, we seek a decision rule that not only performs well in discriminating whether a particular one of those vectors belongs in Group 1 or Group 2, but also whether a new vector A not among the original known vectors should belong in one group or the other. The decision rules we investigate are based on hyperplane separation approaches. Our design makes special use of a procedure called *successive perfect separation* that compels one of the two separating regions to contain all points of one of the groups at each branch.

INTEGER PROGRAMMING MODELS FOR MULTIPLE SEPARATING HYPERPLANES

As a first step toward introducing more advanced mixed integer models, we begin by examining a simple model to minimize the number of misclassified points by means of a single hyperplane. The following model denoted as *Model 1* is due to Glover [5] and seeks to minimize the sum of the binary z_i variables, and hence to minimize the number of misclassified points:

$$\begin{array}{lll} \text{Minimize} & \sum_{i\in G} z_i & \qquad (1.1) \\ \text{Subject to:} & a_{ij}x_j - Mz_i \le b, \quad i\in G_1 & \qquad (1.2) \\ & a_{ij}x_j + Mz_i \ge b, \quad i\in G_2 & \qquad (1.3) \\ & x_j,\ b \text{ unrestricted} & \qquad (1.4) \\ & z_i \in \{0,1\}, \quad i\in G & \qquad (1.5) \\ & \sum_{j} x_j = C & \qquad (1.6) \end{array}$$

Note that (1.2) and (1.3) express the inequalities $A_ix - Mz_i \le b$ (for elements of G_1) and $A_ix + Mz_i \ge b$ (for elements of G_2). The constant M takes a large value to assure that the inequality will be redundant whenever $z_i = 1$. Equation (1.6) is called the "normalization" constraint, necessary to avoid a trivial solution where all $x_j = 0$ and $b = 0$.

The goal of classification and discrimination analysis approaches that involve constructing multiple hyperplanes is to approximate a non-linear boundary that accommodates more complex structures in the underlying data. In figure 1, we seek to separate the **x**-elements from the **o**-elements. Figure 1.a shows a data set that is completely separable by a single linear boundary, while figure 1.b shows one where a set of two hyperplanes is necessary.

Our approach, developed in the context of *successive perfect separation* (SPS) decision trees, achieves the same goals as piecewise-linear models (i.e. approximating non-linear boundaries to correctly discriminate complex data), but without the convexity requirement and using about half the number of attribute weight variables. The multi-hyperplane model may be interpreted as identifying hyperplanes that are applied successively to generate a tree of conditional rules for separating the points, thus creating a formulation for the conditional hyperplane approach sketched in [4]. In order to explain our model, we start by defining a SPS tree.

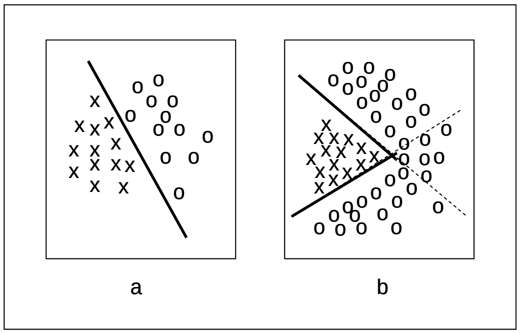

Figure 1. Single and multiple separating hyperplanes

Definition 1: ***Successive Perfect Separation.*** We call Successive Perfect Separation (SPS) a procedure by which at each depth $0 < d < D$ of a binary decision tree we compel all elements of either group 1 (G_1) or group 2 (G_2) to lie entirely on one side of the hyperplane. (At the final depth $d = D$ we just perform a separation of the residual elements into their corresponding groups.)

Definition 2: ***SPS Decision Tree.*** A SPS decision tree is a decision tree that results from applying the SPS procedure to the classification problem. Thus, at each depth d (for $0 < d < D$) the tree has one leaf node that terminates the branch that correctly classifies elements in a given group. At $d = 0$ the tree has a root node containing all the elements in the data set, and at $d = D$ the tree has two leaf nodes corresponding to the final separation step. Figure 2 illustrates the structure of a SPS tree.

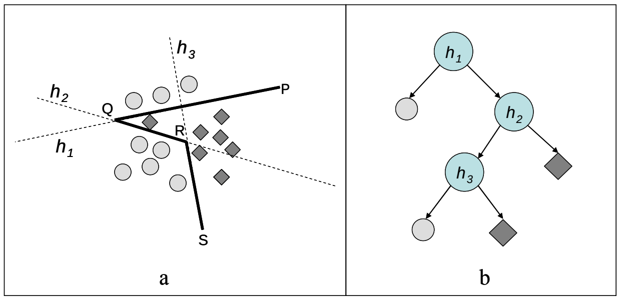

Figure 2. One particular type of SPS tree for $D = 3$

A SPS Decision Tree Model: For a selected maximum depth $d = D$ of the decision tree, we provide a model that implicitly considers each possible SPS tree type of depth d. The type of tree produced is determined by the position of its root nodes and leaf nodes. In the context of our model, a root node is considered a "problem" node where points from both groups need to be separated, while a leaf node is considered a "decision" node where points are classified into their particular group. Figure 2 shows one possible type of SPS tree for $D = 3$, and its corresponding classification rules. In figure 2.a, we seek to separate points represented as *squares* from points represented as *circles*. The points are separated by three hyperplanes, denoted by h_1, h_2 and h_3. The boundary PQRS (shown as heavier solid lines) is formed by segments *PQ, QR* and *RS* of the three original hyperplanes. As demonstrated by this example, it is not necessary for the boundary to form a convex region, which enhances the flexibility of our approach for accommodating complex data structures. For the type of tree depicted in figure 2.b, a *circle* will be correctly classified by the tree if it is either correctly classified by (i.e. lies on the correct side of) h_1, **or** by both h_2 **and** h_3 (corresponding to the circular leaf nodes); conversely, a *square* will be correctly classified by the tree if it is correctly classified by both h_1 ***and*** h_2 **or** by both h_1 ***and*** h_3 (corresponding to the square leaf nodes). Our model, which we will call the *Generalized Structure* model, captures all possible SPS tree structures without having to explicitly identify and enumerate each one within the model framework. Instead, the Generalized Structure model automatically embraces all of the 2^D structures in an implicit fashion.

Let d denote the depth index 1, 2,…, D which identifies the total number of hyperplanes that will be generated. For each depth d we introduce variables $x[d]$, $z_i[d]$, and $b[d]$ corresponding to the variables x, z_i and b in model 1 (where the maximum value of d was 1). We also have a continuous variable $v_i[d]$, whose value is forced to be 0 or 1 according to the value received by $z_i[d]$ and by another binary variable $y[d]$. This latter variable constitutes our successive perfect separation variable, and is defined as $y[d] = 0$ if all $i \in G_1$ are compelled to lie on one side of hyperplane d, $y[d] = 1$ if all $i \in G_2$ are compelled to lie on one side of hyperplane d. The variable $y[d]$ is not included for the final hyperplane (the one associated with the maximum depth D), since we do not compel the points of either Group 1 or Group 2 to lie all on one side of the hyperplane. Similarly, the variables $v_i[d]$ will not be included for this last value D. Then, at any given depth d except the final one, we write $y[d] \geq (z_i[d]\text{: } i \in \mathrm{G}_1)$ *and* $(1 - y[d]) \geq (z_i[d]\text{: } i \in G_2)$ to compel the appropriate $z_i[d]$ variables to be 0. The complete formulation of the *Generalized Structure* SPS model, which we denote as *model 2* is as follows:

$$\text{Minimize} \quad \sum_{i \in G} z_i[D] \tag{2.1}$$

$$\text{Subject to:} \quad A_i x[d] - M\Big(\sum_{h=1}^{d-1} v_i[h] + z_i[d]\Big) + s_i = b[d] - \varepsilon,\ i \in G_1,\ d=1,\ldots,D \tag{2.2}$$

$$A_i x[d] + M\Big(\sum_{h=1}^{d-1} v_i[h] + z_i[d]\Big) + s_i = b[d] + \varepsilon,\ i \in G_2,\ d=1,\ldots,D \tag{2.3}$$

$$y[d] \geq z_i[d],\ i \in G_1,\ d=1,\ldots,D\text{-}1 \tag{2.4}$$

$$1 - y[d] \geq z_i[d],\ i \in G_2,\ d=1,\ldots,D\text{-}1 \tag{2.5}$$

$$v_i[d] \leq y[d],\ i \in G_1,\ d=1,\ldots,D\text{-}1 \tag{2.6}$$

$$v_i[d] \leq 1 - y[d],\ i \in G_2,\ d=1,\ldots,D\text{-}1 \tag{2.7}$$

$$v_i[d] \leq 1 - z_i[d],\ i \in G,\ d=1,\ldots,D\text{-}1 \tag{2.8}$$

$$\sum_{d=1}^{D} \sum_{j=1}^{F} x_j[d] = C,\ \text{“Normalization”} \tag{2.9}$$

$$x[d],\ b[d] \text{ unrestricted},\ d=1,\ldots,D \tag{2.10}$$

$$z_i[d] \in \{0,1\},\ d=1,\ldots,D \tag{2.11}$$

$$y_i[d] \in \{0,1\},\ d=1,\ldots,D\text{-}1 \tag{2.12}$$

$$0 \leq v_i[d] \leq 1,\ d=1,\ldots,D\text{-}1 \tag{2.13}$$

Note that in this model we use ε as a parameter in the hyperplane constraints (2.2) and (2.3).

EXPERIMENTATION AND RESULTS

We tested our models on two widely studied benchmark sets: the Japanese Banks data from [8] and the Wisconsin Breast Cancer data from the UCI Machine Learning Repository, both consisting of data from real world applications. We specifically tested our approach against the piecewise-linear model of [3], which appears to provide the best competing results for these data sets, by using the Leave-One-Out (LOO) procedure.

In order to eliminate large discrepancies in scale among the attribute values for the Japanese Banks, we standardized the data. Tables 1 and 2 summarize our results for various values of parameter ε. All tests were performed using CPLEX 10.0, on a Dell Dimension 8400 workstation equipped with a Pentium 4 processor with 3.60 GHz speed and 1.0 GB RAM. We used the parameter setting $M = 100$.

Table 1. LOO Hit Rates and solution times (in seconds) for Japanese Banks

Separation Zone (ε)	**Gen. Structure Model 2**	**Piecewise-Linear Model**	**Time Model 2**	**Time P-L Model**
0.0005	84	86	7.5	126.9
0.0010	88	86	8.5	126.6
0.0100	88	85	165.2	113.6
0.0200	90	85	120.5	142.2
0.0300	88	81	117.4	168.1
0.0400	89	85	186.3	414.4
0.0450	90	90	101.0	506.2

As table 1 shows, our model yields better testing accuracy in the majority of cases. Furthermore, each LOO test takes on average 100.9 seconds by the *Generalized Structure* model. Our implementation of Glen's piecewise-linear model takes an average of 228.3 seconds for each LOO test. We only ran the case where elements from G_2 are required to be in the convex region (In [3], Glen shows that this performs better than the case where G_1 elements lie in the convex region) – otherwise, the time for the piecewise-linear LOO test would have been – at best – approximately double.

The Wisconsin Breast Cancer data consists of 683 patients screened for breast cancer (we eliminated all cases with missing values), and 10 attributes per case. Preliminary testing of our model showed that this data set is completely separable by three hyperplanes, so again we used $D = 3$. However, due to its convexity requirement, the piecewise-linear model failed to separate the data completely with three piecewise-linear segments. As with the Japanese Banks, we ran various tests for different values of ε. It should be noted that for these tests, given the size of the data set, we set a time limit in CPLEX of 120 seconds per iteration. We found that the trade-off between classification accuracy and the time to obtain an optimal solution by CPLEX strongly justifies the use of such a time limit. Table 3 summarizes the results for experiments where $\varepsilon = 0.00004$, 0.00005 and 0.00006. We ran experiments with values at different orders of magnitude for ε, but the results did not vary significantly compared to the ones reported here.

Table 3. LOO Hit Rates for Breast Cancer Data

ε	Gen. Structure	Piecewise-Linear
0.00004	92.8	84.6
0.00005	94.2	84.6
0.00006	91.5	84.6

As table 3 shows, the Generalized Structure model outperforms the piecewise-linear model. We ran the piecewise-linear model twice, alternating the convexity requirement between G_1 and G_2, and recorded the one that resulted in the best LOO classification performance. In terms of solution time, the Generalized Structure model is almost twice as fast as the piecewise-linear model, with an average time for each LOO test around 500 to 600 seconds. The average size of the **.lp** file generated by each model is intended as a proxy for the size and complexity of the MIP model. Our model generated .lp files of 158 kb on average, while for the piecewise-linear model the average .lp file size was 357 kb.

A general characteristic of the models presented here is that the MIP can have many optimal solutions. In other words, in the case of data that is separable by trees of depth *D*, there can be an infinite number of sets of *D* hyperplanes that separate the data. Therefore, the quality of the classifier may depend on the algorithm of the particular solver being used, such as the rules governing its branch and bound procedure. Furthermore, the order in which the variables (attributes) are presented to the solver may also influence the quality of the solution for classification purposes. Glover [6] addresses some of these issues by suggesting retrospective enhancement and robust separation strategies that make use of *maximizing the minimum distance* from critical points to the hyperplanes in order to find an optimal solution that will perform best when classifying holdout observations.

CONCLUSIONS

Our approach to the classification and discrimination analysis problem constitutes an innovation in the area of support vector machines that has broad application to problems of pressing importance in real world data mining. The use of multiple linear separating hyperplanes makes it possible to separate the original data without relying on efforts to discover kernel transformations that seek to project the data into a higher dimensional space. In addition, our models make use of *successive perfect separation* conditions that create a special decision tree structure for classification purposes.

For the data sets we tested, an analysis using the LOO approach demonstrates that our models compare quite favorably in accuracy to the previous best mixed integer multi-hyperplane (piecewise-linear) model in the literature, and not only operates under less restrictive assumptions but is significantly more efficient in terms of solution time.

REFERENCES

[1] Bennett, K. P. and **J.** Blue (1998) “A Support Vector Machine Approach to Decision Trees,” *Neural Networks Proceedings of the IEEE World Congress on Computational Intelligence,* Vol. 3, pp. 2396-2401.

[2] Dudoit,S., Fridlyand,J. and Speed,T.P. (2002) “Comparison of discrimination methods for the classification of tumors using gene expression data,” *Journal of the American Statistical Association*, Vol. 97, pp. 77–87.

[3] Glen, J.J. (2005) “Mathematical programming models for piecewise-linear discriminant analysis,” *Journal of the Operational Research Society*, Vol. 56, pp. 331-341.

[4] Glover, F. (1990) "Improved Linear Programming Models for Discriminant Analysis," *Decision Sciences*, Vol, 21, No. 4, pp. 771-785.

[5] Glover, F. (1993) "Improved Linear and Integer Programming Models for Discriminant Analysis," *Creative and Innovative Approaches to the Science of Management*, RGK Foundation Press, pp. 187-215.

[6] Glover, F. (2006a) “Improved classification and discrimination by successive hyperplane and multi-hyperplane separation,” (Working Paper), University of Colorado at Boulder.

[7] Stam, A. and Ragsdale, C.T. (1992) “On the classification gap in mathematical programming-based approaches to the discriminant problem.” *Naval Research Logistics*, 39, pp. 545-559.

[8] Sueyoshi, T. (2001) “Extended DEA-discriminant analysis.” *European Journal of Operational Research,* 131**,** pp**.** 324-351.

[9] Sun, M. and M. Xiong (2003) “A mathematical programming approach for gene selection and tissue classification.” *Bioinformatics,* 19, 10, pp. 1243-1251.