

A Novel Approach to Classification in Financial Applications

Marco Better

Fred Glover

*Leeds School of Business, University of Colorado at Boulder
Boulder, Colorado, USA*

Gary Kochenberger

*School of Business, University of Colorado at Denver
Denver, Colorado*

Haibo Wang

*College of Business Administration, Texas A&M International University
Laredo, Texas*

Abstract:

Modern methods for classification analysis involve processes for “learning” to correctly assign elements of a data set to certain classes. In many settings, the learning processes are supervised; i.e. the classes that the training data belong to are known in advance. In many other settings, however, the classes are not known a priori, and a process utilizing unsupervised learning is necessary.

We present a novel, two-stage unsupervised learning methodology for the classification problem. Stage one consists of a special clustering method based on a quadratic, unconstrained optimization model that finds optimal classes for the data. Stage two makes use of enhanced mathematical programming models for classifying the data into the optimal classes found during stage one.

A significant advantage of our approach, as demonstrated by computational testing, is the ability to yield more meaningful classifications than previously achieved in a variety of settings. We report the outcome of training and testing our method on various data sets from the data mining literature, with specific applications in finance. The comparative results disclose the effectiveness and versatility of the approach, and its merit as a tool for modeling and solving practical problems.

Introduction

Many classification and discrimination analysis applications involve *supervised learning*, in which the training data is labeled with the appropriate class definition. In some settings, however, the class definition itself may have been subjective or ambiguous. For example, bond rating agencies such as Moody’s and Standard and Poor each have a proprietary algorithm for rating bonds, which may result in different rating scales, and thus, different assessments of the risk of the same underlying bond.

In such instances, it is unclear whether one class definition is better than another. Furthermore, there is a certain amount of subjectivity in the class definition inasmuch as the “experts” evaluating the elements of the different classes may disagree on the relative importance of each of the attributes used as criteria for classification.

In order to overcome this problem, we propose a two-stage approach to the classification problem. The first stage clusters the data into “optimal” classes, and the second stage seeks to classify the data correctly into the optimal classes found in stage one. For the purpose of clustering the data, we use the method described in Kochenberger et al (2005), which makes use of a quadratic unconstrained binary quadratic program (UBQP) for clique partitioning. A tabu search (TS) procedure from Glover et al (1999) is used to efficiently solve the UBQP. The classification stage is carried out by a multi-hyperplane mixed integer programming formulation for discrimination analysis, similar to those described in Better et al (2006).

The paper is organized as follows: section 1 provides a brief description of our clustering algorithm; section 2 describes a basic multi-hyperplane model for classification of data in two groups; section 3 introduces two examples that use real data in order to illustrate our approach; and section 4 summarizes our results and our conclusions.

1. Clustering via UBQP Clique Partitioning

The clique partitioning (CP) problem consists of partitioning a graph $G = (V, E)$ ¹ into *cliques* such that the sum of the edge weights over all cliques formed is maximized. A clique can be defined as a fully connected subgraph (Feder and Motwani 1991). Classical formulations for clique partitioning involve linear integer models that use binary decision variables related to the *edges* of the graph. In practice, the classical model works well for relatively small graphs, but explodes in size even for graphs that are moderate in size.

By using binary variables associated with the *nodes* (vertices) of the G , and by using a parameter to limit the maximum number of cliques, Kochenberger et al (2005) show that the problem can be reformulated to solve much larger instances of CP very efficiently. Although the node-based model is a quadratic program, it uses much fewer variables and constraints. After some algebraic manipulation (the details of which are in the referenced paper), the node-based model can be formulated as:

Let K = the maximum number of cliques allowed;
 Let $x_{ik} = 1$ if node i is assigned to clique k ; 0 otherwise;
 Let w_{ij} denote a weight of unrestricted sign on edge (i, j) .

Then, the model (which we will call model 1) is:

$$\text{Maximize} \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^n w_{ij} \sum_{k=1}^K x_{ik} x_{jk} \quad (1.1)$$

Subject to:

$$\sum_{k=1}^K x_{ik} = 1 \quad \text{for } i = 1, \dots, n \quad (1.2)$$

$$x_{ik} \in \{0, 1\} \quad \text{for } i = 1, \dots, n; k = 1, \dots, K \quad (1.3)$$

The quadratic objective function (1) ensures that w_{ij} is not part of the partition weight unless nodes i and j are both assigned to clique k . Equation (2) ensures that each node is assigned to a clique. Parameter K should be estimated from domain knowledge. If K is initially set too low, the solution will result in the optimal number of cliques for this choice of K . In such an event, we can increase the value of K as needed, until the optimal number of cliques is less than K , thus obtaining the globally optimal number.

We can represent the quadratic model in matrix notation as:

$$\begin{aligned} \text{Maximize} \quad & xQx \\ \text{Subject to:} \quad & Ax = b \\ & x \in \{0, 1\} \end{aligned}$$

A further transformation of the model results in the desired UBQP. We incorporate the constraint into the objective function by imposing a scalar penalty P on it. The resulting UBQP model can be written in matrix form as:

$$\text{Maximize} \quad x'Qx - P(x-I)'(x-I), x \in \{0, 1\}$$

which is equivalent to the model in its final form:

$$\text{Maximize} \quad xQx, x \in \{0, 1\}.$$

In order to use this formulation for clustering, we need to perform some data preprocessing. Namely, we consider graph G to be a representation of the *affinities* between pairs of elements in the data

¹ Here, $G = (V, E)$ refers to a graph G containing a set V of vertices and a set E of edges.

set. Thus, loosely speaking, w_{ij} may be viewed as a distance measure between nodes i and j . We then solve the UBQP model by *minimizing* the total weight among the cliques.

2. The Classification Problem

Let a_{ij} denote the value of specific characteristics or attributes of a population of elements in a data set, where each element i ($i=1, \dots, m$) is described by attribute j ($j=1, \dots, n$). We seek a decision rule to classify these elements in a manner to correctly identify whether a given vector $A_i=(a_{i1}, \dots, a_{in})$ should belong among the elements of Group 1 or instead among those of Group 2 (denoted 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 ($i \in G_2$), and 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 forth.

Given the A_i vectors and their group membership, we devise 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 be classified as belonging in one group or the other. The decision rules we investigate here are based on hyperplane separation approaches set forth in Better et al (2006), by simultaneously generating multiple hyperplanes utilizing mixed integer programming formulations. Our design, based on a proposal of Glover (1990), makes special use of a variant called *successive perfect separation* (SPS) that compels one of the two separating regions to contain all points of one of the groups at each branch. Our multi-hyperplane model (which we call model 2) for the 2-group classification problem can be written as follows:

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

Subject to:

$$A_i x[d] - M \left(\sum_{h=1}^{d-1} v_i[h] + z_i[d] \right) \leq b[d] - \varepsilon \quad i \in G_1, d=1, \dots, D \quad (2.2)$$

$$A_i x[d] + M \left(\sum_{h=1}^{d-1} v_i[h] + z_i[d] \right) \geq b[d] + \varepsilon \quad i \in G_2, d=1, \dots, D \quad (2.3)$$

$$y[d] \geq z_i[d] \quad i \in G_1, d=1, \dots, D-1 \quad (2.4)$$

$$1 - y[d] \geq z_i[d] \quad i \in G_2, d=1, \dots, D-1 \quad (2.5)$$

$$v_i[d] \geq y[d] \quad i \in G_1, d=1, \dots, D-1 \quad (2.6)$$

$$v_i[d] \geq 1 - y[d] \quad i \in G_2, d=1, \dots, D-1 \quad (2.7)$$

$$v_i[d] \geq 1 - z_i[d] \quad i \in G, d=1, \dots, D-1 \quad (2.8)$$

$$\sum_{d=1}^D \sum_{j=1}^F x_j[d] = 1 \quad \text{“normalization”} \quad (2.9)$$

$$x[d], b[d] \text{ unrestricted} \quad d=1, \dots, D \quad (2.10)$$

$$z_i[d] \in \{0,1\} \quad d=1, \dots, D \quad (2.11)$$

$$y_i[d] \in \{0,1\} \quad d=1, \dots, D-1 \quad (2.12)$$

$$0 \leq v_i[d] \leq 1 \quad d=1, \dots, D-1 \quad (2.13)$$

where D is a parameter for the maximum allowable depth of the decision tree. We use parameter ε as a measure of the width of a “separation zone” to either side of the separating hyperplane, in order to avoid solutions where points from both groups lie directly on the hyperplane. As we will show, changing the value of ε can often affect the quality of the solution.

Our model constructs a decision tree of a specific structure which we call a SPS Tree, so named because it uses a *successive perfect separation* strategy at each step, to produce a separation that causes all

elements of one group to lie entirely on one side of the hyperplane. Figure 1 shows one possible SPS tree for the case where $D = 3$.

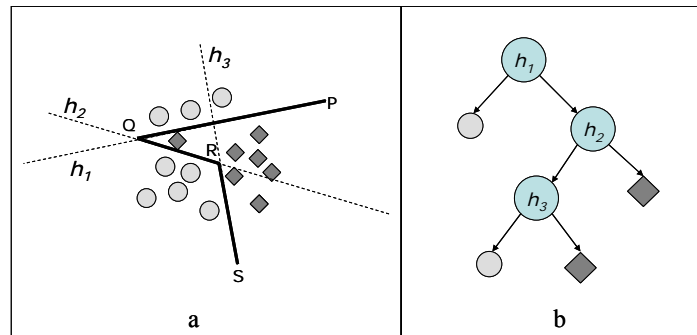


Figure 1: One Type of SPS Tree for $D=3$.

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. Figure 2.b. shows the structure of the corresponding tree.

3. Two Well-Known Examples

In order to illustrate our approach, we use two well-known data sets from the data mining and machine learning literature. The first, from Sueyoshi (2001), shows a ranking of 100 Japanese financial institutions by a panel of experts. We will call this the Japanese Banks data set. The second, which we will call Bond Rating, consists of Moody’s ratings for 95 US corporate bonds, taken from an online repository of the Sloan School of Business at MIT. In both cases, the data set is classified into two (or more) groups by a panel of experts. (In the case of bond ratings, the panel of experts is the rating agency.)

The Japanese Banks dataset consists of 7 financial ratios for each of the 100 Japanese banks, given by: (1) Return on Assets; (2) Equity to Total Assets; (3) Operating Cost to Profits; (4) Return on Domestic Assets; (5) Bad Loan Ratio; (6) Loss Ratio on Bad Loans; and (7) Return on Equity. In Sueyoshi (2001) the data are given to a panel of financial executives, and the banks are ranked from best to worst. Then, the banks are divided into a “good” group (banks 1 through 50) and a “bad” group (banks 51 through 100). However, it is possible that some banks were classified into the wrong group. For instance, there is no apparent foundation to conclude that bank 51 should not belong in the same group as bank 50. Through the use of our clustering-based approach we seek a more objective classification.

For the Bond Rating dataset, we are given 10 financial ratios: (1) Operating Margin; (2) Pre-Tax Fixed Charge Coverage; (3) Long-Term Debt to Capitalization; (4) Long-Term Debt to Total Equity; (5) Leverage; (6) Net Tangible Assets to Long-Term Debt; (7) Cash Flow to Long-Term Debt; (8) Acid Test Ratio; (9) Current Assets to Current Liabilities; and (10) Accounts Receivable Turnover. Moody’s Rating Agency (see www.moodys.com) classifies bonds into 9 different categories of risk, from **Aaa** to **C**. However, anything above a **Baa** rating is considered an “investment grade” bond (i.e. good), and anything at, or below **Baa** is considered a “junk” bond (i.e. bad). As with the Japanese banks, our goal was to see whether there could be a better classification, especially in view of the disagreement among the various bond rating agencies about the true risk level of a bond (Better, 2006).

For both examples, we tested our clustering-based approach against the original classification as follows:

1. We take the original classification by experts and run a *leave one out* (LOO) procedure, so that, given $N = 100$ observations in the data set, at each iteration one element of the data set becomes the holdout element. Each iteration consists of applying our multi-hyperplane model to the training set comprised of $N - 1 = 99$ elements, and once the optimal hyperplanes are obtained we determine in which group the holdout element belongs. We then compare this classification to the correct one. If both are equal, then we have a “hit”; otherwise, we don’t. This is done N times, until each element has been the holdout once.

2. We then record the “hit rate”² for the procedure on the original data.
3. Next we perform our clustering method on the data set to obtain a different classification of the elements using $K = 2$ (we are only interested in 2-group classification in this study).
4. Given the new classification obtained in step 3, we run a new LOO procedure – as in step 1 – on the clustered data for N iterations, as before.
5. We record the “hit rate” for the procedure on the clustered data and compare to the “hit rate” corresponding to the original data.
6. Steps 1 through 5 are repeated for various values of parameter ϵ .

4. Results and Conclusions

The results for LOO testing on the Japanese Banks data set are summarized in Table 1. It is important to point out that the clustering method only moved a total of 10 of the 100 banks from their originally assigned group to the other, with a net result of 42 banks in the new “good” group and 58 banks in the new “bad” group. We tested both the raw data and a standardized version of the data, as shown. In order to obtain results in a reasonable amount of time, we limited the time for a complete LOO test to 300 seconds. In the case of the original, raw data, we could not obtain a solution in that amount of time (hence, the entries with N/A in table 1.)

Table 1: Summary LOO hit rates and solution times for Japanese Banks.

ϵ	Original Classification		New Classification	
	Raw	Standardized	Raw	Standardized
0.01	N/A	82	96	90
0.02	N/A	88	93	91
0.03	N/A	87	94	91
0.04	N/A	92	95	91
0.05	N/A	89	92	91
Avg. Time (sec.)	> 300	120.1	6.8	7.1

In addition to yielding a significant improvement in terms of LOO hit rate, our clustering-based approach performed vastly better in terms of computational time, as seen by comparing the two New Classification entries to the two Original Classification entries in the bottom row of Table 1.

Tables 2 and 3 summarize the results for the Bond Rating data set. Here, an original composition of 40 investment-grade bonds and 55 junk bonds resulted in a re-classification into two groups consisting of 32 and 63 elements respectively after clustering. In this case, more than 75% of the bonds conserved their original classification.

Table 2: LOO hit rates for the Bond Ratings

ϵ	Original	Clustering-Based
.00005	69	92
.0005	71	91
.005	74	90
.01	62	88

Table 3: Solution times for the Bond Ratings

ϵ	Original	Clustering-Based
.00005	7.0	7.0
.0005	883	7.0
.005	8237	7.1
.01	8135	6.8

² The “hit rate” is a percentage, defined as the number of correct classifications over the total number of iterations.

As the tables show, our model again yields a significant improvement in the LOO hit rates as well as in the solution times. In addition, the breakout of solution times in Table 3 shows that our approach is very scaleable for this data set, in that the solution time remains constant as a function of ϵ – an outcome dramatically different from that resulting from the original grouping for this data set.

This opens up the possibility for more objective ratings methods in the future. Due to the flexibility of our approach, if some application affords a basis to believe that original rankings by experts should be given greater credibility, we can readily incorporate such rankings as attributes in our model to influence the resulting classification. On the other hand, experiments in behavioral and social psychology have shown that experts often change their evaluations when provided evidence that they may not be consistent with objective considerations (see, e.g., Hammond et al, 1980). In this respect our approach can be used as a tool to help teams of experts to recalibrate and refine their evaluations. It is noteworthy in the present study that our approach in fact appears to key on many of the same features used in the experts' subjective evaluations, in view of the significant overlap between our classifications and the original classifications. Should our model somehow be teasing out properties found relevant by experts, while at the same time disclosing the merit of alternative groupings based on such properties, that would be an unexpected bonus.

References

- Better, M. (2006) "A Study on the Validity of a Clustering Approach to the Classification Problem." working paper, University of Colorado, Boulder, Colorado.
- Better, M., F. Glover and M. Samorani (2006) "Multi-Hyperplane Formulations for Classification and Discrimination Analysis." submitted for the Student Paper Award of the Decision Analysis Society, working paper, University of Colorado, Boulder, Colorado.
- Feder, T. and R. Motwani (1991) "Clique Partitions, Graph Compression and Speeding Up Algorithms." in *Proceedings of the twenty third annual ACM Symposium on Theory of Computing*, New Orleans, Louisiana, pp. 123-133.
- Glover, F. (1990) "Improved Linear Programming Models for Discriminant Analysis." *Decision Sciences*, Vol, 21, No. 4, pp. 771-785.
- Glover, F., G. Kochenberger, B. Alidaee, and M. Amini (1999) "Tabu Search with Critical Event Memory: An enhanced application for binary quadratic programs." in *Meta-Heuristics, Advances and Trends in Local Search Paradigms for Optimization*, S.M.S. Voss, I. Osman, and C. Roucairol (Eds.), Kluwer Publisher, 1999, pp. 93–109.
- Hammond, K. R., G.H. McClelland and J. Mumpower (1980) "Human Judgment and Decision Making." New York: Praeger.
- Kochenberger, G., F. Glover, B. Alidaee and H. Wang (2005) "Clustering of Microarray Data via Clique Partitioning." *Journal of Combinatorial Optimization*, Vol. 10, pp. 77-92.
- Sueyoshi, T. (2001). "Extended DEA-Discriminant Analysis." *European Journal of Operational Research*, 131, pp. 324-351.