Bankruptcy Prediction Using Feature Projection Based Classification

H. Altay Güvenir, Sengör Altıngövde, İlhan Uysal,
Computer Engineering and Information Science Department, Bilkent University
Ankara 06533, Turkey

And

Erdal Erel
Management Department, Bilkent University
Ankara 06533, Turkey

ABSTRACT

Bankruptcy prediction has been an important decision-making process for financial analysts. One of the most common approaches for the bankruptcy prediction problem is the Discriminant Analysis. Also, the k-Nearest Neighbor classifier is very successful in such domains. This paper proposes a Feature Projection based classification algorithm, and explores its applicability to the problem of predicting bankruptcy of large firms. The algorithm is evaluated on a particular data set, and its performance is compared with the techniques mentioned above. The experiments indicate that the feature projection based classification algorithm introduced here performs better than these techniques.

Keywords: Bankruptcy, prediction, classification, feature projections, feature intervals.

1. INTRODUCTION

Bankruptcy prediction has been an important decision-making process for financial analysts. Many techniques have been proposed for helping financial analysts in this process. Among those techniques Multivariate Discriminant Analysis (MDA) [7] and k-Nearest Neighbor (k-NN) [3] are among the most successful. In these techniques, the status of a firm analyzed is modeled by its financial ratios; such as cash flow / total assets, cash / total sales, net income / total assets, and (total assets - inventories) / total liabilities. In MDA a discriminant function of such financial ratios is constructed. If the value of the discriminant function for a given firm is below the cutting score (a threshold value), then the firm is predicted to go bankrupt, otherwise it is predicted to stay in business. The k-NN classifier, on the other hand, searches for k training cases that are most similar to the firm being tested, in terms of its financial ratios. The firm is predicted to go bankrupt if most of these similar firms went bankrupt. The k-NN algorithm assumes that all features are equally important in the classification, unless the feature weights are provided externally.

Clearly, constructing the true model to predict bankruptcy in firms, from such ratios alone, is impossible or unattainable. However, machine learning techniques can form an approximate model from the past cases (training instances) to make a prediction for novel cases (test instances). In this paper, we propose such a machine learning technique that represents the learned information in the form of projections of the training instances on each feature. Classification algorithms that use this knowledge representation technique are called *Feature Projection based Classifiers*, and many such classifiers have been shown to be successful in very large range of real-world problem domains [1, 4, 5, 6]. The difference between these algorithms is in the way they process those projections.

In this study, we introduce another feature projections based classification algorithm, called *Feature Intervals Learner* (FIL5, for short), for the bankruptcy prediction problem. Given the values of n financial ratios (features), a firm is represented as a point in an n dimensional space. First, all training cases are stored as their projections on n feature dimensions. Then, neighboring projection points representing the same class of cases are grouped to form feature intervals. Given a new case, with the values of each ratio, a prediction, as bankruptcy or non-bankruptcy, is made by each feature. The prediction based on a single feature is made by considering the observed class values (bankrupt or non-bankrupt) of the interval that the projections of that case falls in on that feature. The final decision is based on a voting on the predictions made by all n features. We have compared the FIL5 algorithm with MDA and k-NN algorithms on a real-world bankruptcy data set. In terms of the prediction accuracy, k-NN performed better than the MDA algorithm. However, the time required by k-NN for a classification was much longer than that of the MDA algorithm. The FIL5 classifier achieved an even better classification accuracy than k-NN in this data set. Further, the FIL5 algorithm was much faster than the k-NN algorithm in classification. The reason for the better predictive accuracy of the FIL5 classifier over the k-NN algorithm is that it learns the relative relevance of each interval of feature values.

The next two Sections briefly describe the MDA and the k-NN techniques. Section 4 introduces the FIL5 algorithm. Section 5 describes the bankruptcy data set used for the evaluation. The results of the evaluation are presented in Section 6, along with the comparison with the MDA and k-NN techniques. Section 7 concludes with an overall evaluation of the method proposed, and planned feature work.

2. DISCRIMINANT ANALYSIS

MDA is the statistical technique used most frequently by financial analysts for bankruptcy of financial distress analysis. The primary objectives of multiple discriminant analysis are to understand group differences and to predict the likelihood that an entity will belong to a particular class based on several continuous (numeric) independent variables [7]. In a domain with n features (independent variables), MDA results in a discriminant function for each class Y_i in the following general form:

$$Y_i = w_{i0} + w_{i1}f_1 + w_{i2}f_2 + \cdots + w_{in}f_n \quad (1)$$

Here, w_{ij} represents the weight of feature f_j for the class Y_i. The w_{i0} is called the intercept. When observed on a normalized scale, the more important features usually have weights greater magnitude that contribute more to the final score. For a domain with n features and c classes, the MDA solution is a set of c equations, each of which is a weighted combination of the n features as shown below.

$$\begin{aligned} Y_1 &= w_{10} + w_{11}f_1 + w_{12}f_2 + \cdots + w_{1n}f_n \\ Y_2 &= w_{20} + w_{21}f_1 + w_{22}f_2 + \cdots + w_{2n}f_n \\ &\vdots \\ Y_c &= w_{c0} + w_{c1}f_1 + w_{c2}f_2 + \cdots + w_{cn}f_n \quad (2) \end{aligned}$$

A new case is classified by replacing the features with measured values in the equations and computing the scores, Y_i's. The class with the largest score is selected as the prediction.

Because the solution obtained by MDA is always linear, the dimensions of the solution are fixed by the number of features. Some reduction in complexity is achieved by reducing the number of features through feature selection [10]. Feature selection reduces complexity, but rarely produces better results than the full feature set. Also, MDA has an obvious difficulty with missing feature values.

3. NEAREST NEIGHBOR

Instance-based machine learning methods such as nearest neighbor are conceptually straightforward approaches to approximating real-valued or nominal target functions. These methods construct only a local approximation to the target function that applies only in the neighborhood of the test instance. They do not try to construct an approximation designed to perform well over the entire instance space.

The most basic instance-based method is the nearest neighbor algorithm. The nearest neighbor classification algorithm is based on the assumption that examples which are closer in the instance space are of the same class. Namely, unclassified ones should belong to the same class as their nearest neighbor in the training data set. After all the cases in the training set is stored in memory, a new example is classified with the class of the nearest neighbor among all stored training instances. Although several distance metrics have been proposed for nearest neighbor algorithms [9], the most common metric is the Euclidean distance metric. The Euclidean distance between two instances, represented in the form of feature vectors, $x =< x_1, x_2, ..., x_n >$ and $y =< y_1, y_2, ...y_n >$ on an n dimensional space is computed as:

$$dist(x,y) = \sqrt{\sum_{f=1}^{n} diff(f,x,y)^2} \quad (3)$$

$$diff(f,x,y) = \begin{cases} |x_f - y_f| & \text{if } f \text{ is linear} \\ 0 & \text{if} f \text{ is nominal} \\ & \quad \text{and } x_f = y_f \\ 1 & \text{if} f \text{ is nominal} \\ & \quad \text{and } x_f \neq y_f \end{cases} \quad (4)$$

Here $diff(f,x,y)$ denotes the difference between the values of instances x, and y on feature f. Note that this metric requires the normalization of all feature values into a same range.

Although several techniques have been developed for handling unknown (missing) feature values [8], the most common approach is to set them to the mean value of the values on corresponding feature.

A generalization of the nearest neighbor algorithm, k-NN, classifies a new instance by a majority voting among its k ($1 \leq k$) nearest neighbors using some distance metrics. This algorithm can be quite effective when the attributes of the data are equally important. However, it can be less effective when many of the attributes are misleading or irrelevant to classification. Another obvious difficulty of the k-NN classifier is its classification time complexity, which is $O(nm)$, where m is the number of training instances. Since the classification of an instance requires to measure its distance to every training instance, the time required for the classification becomes quite long, especially when large number of training examples are available. Finally, the choice of the k value is arbitrary. In order to determine the best value for k, one has to experiment with several values, and then choose the one that gives the best accuracy.

4. FEATURE PROJECTIONS

Feature Projections based learning techniques store the training examples in the form of their projections on each feature dimension separately. The neighboring projection points representing the same class of cases can be grouped into feature value intervals. In these techniques the the class of a new instances is predicted by some form of voting among the individual predictions of features, which is based on the number of training instances that have the similar value for that feature.

FIL5 is a feature interval learning algorithm. During its training phase, the FIL5 algorithm constructs feature intervals for each feature, using the training instances provided (Fig.1.a). First, all training instances are projected on each feature dimension (Fig.1.b). Then, those instances that have the same value on a feature dimension are grouped into point intervals (Fig.1.c). An interval has two components: range and the vote vector. The range of an interval is represented by the lower and upper values. For a point interval, the lower and upper values are equal to the feature value of the corresponding projections. The vote vector has one element for each class. It is in the form $< v_1, v_2, \cdots, v_c >$, where c is the number of classes in the domain. The elements of the vote vector represent the votes that the interval will cast for each class. The vote for class c of an interval at value x on a feature f is determined as

$$v_i = \frac{|\{e \mid e_{class} = c \text{ and } e_f = x\}|}{|\{e | e_{class} = c \ \}|}. \quad (5)$$

The class counts on an interval is normalized by the total number of training instances of a given class to eliminate effects of possible uneven distribution of classes in the training set. In the next step, neighboring intervals whose highest voting classes are the same are combined into single non-point intervals. Finally, the vote values of each interval is normalized so that the total votes of an internal is 1 (Fig.1.d).

When classifying a query instance, it is first projected onto each feature dimension. The intervals on each feature dimension is sought. This search can be done using a binary search, with time complexity of $O(\log m)$, since there are at most m intervals on each feature dimension. The total time complexity of the search is $O(n \log m)$. The vote vectors of the intervals found in each feature dimension is summed. If

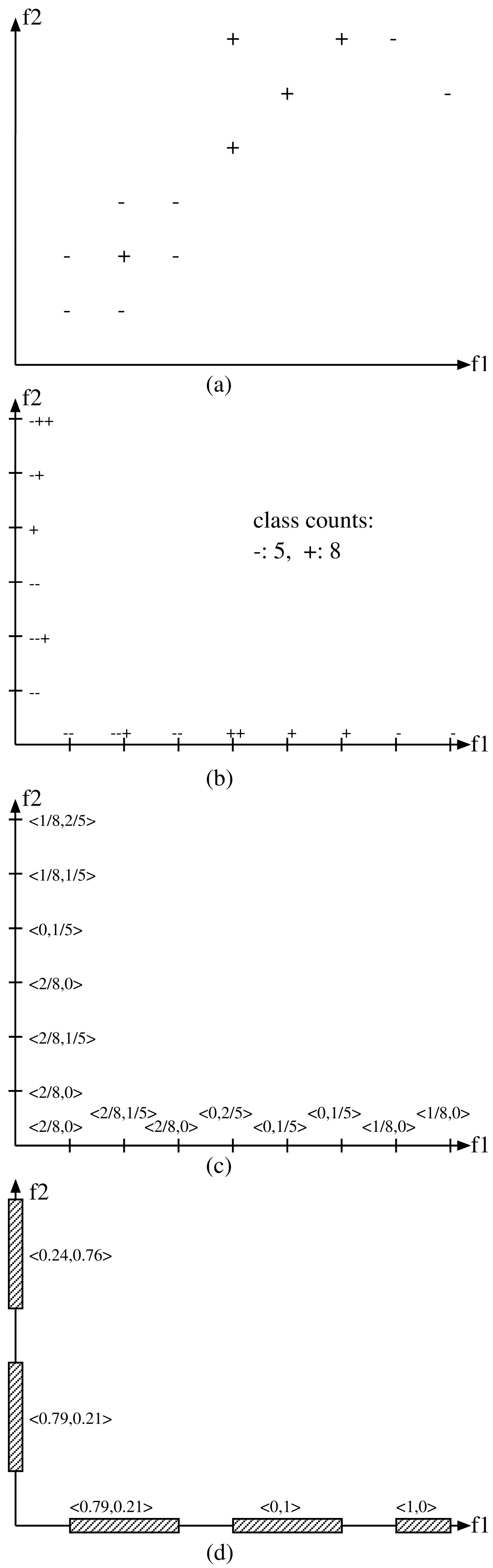

Figure 1: Construction of feature intervals in FIL5 for a two dimensional domain; (a) Training examples, (b) Projections, (c) Point intervals, (d) Final intervals.

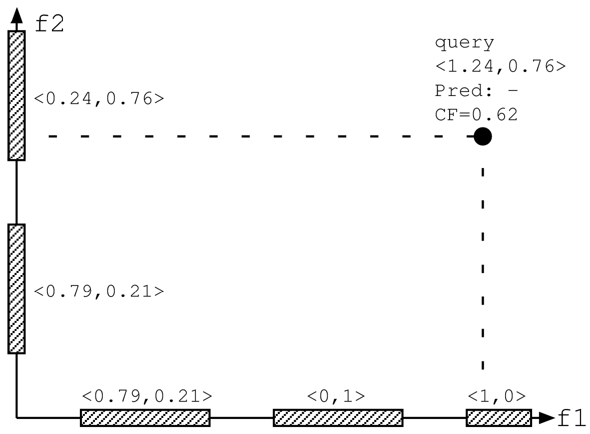

Figure 2: Classification using intervals. The vote vector for the query point is $< 1, 24, 0 >$, the class with highest votes is -, certainty factor with this classification is 0.62.

the projected value of the query instance does not fall in an interval for a feature dimension, that feature does not participate in the voting. Therefor, missing feature values are handled in a natural way. The class that receives the highest amount of votes is predicted as the class of the query instance Fig.2. A certainty factor of

$$CF = \frac{\text{Total votes of the class}}{\text{n}} \tag{6}$$

can be associated with the prediction.

5. DATA SET

In order to evaluate the performance of the feature projections based classification technique proposed above we have implemented the FIL5 algorithm and compared it with the other two classical algorithms. For this comparison we used the data compiled from the Compact Disclosures^{TM} by Dorsey et al. [2]. The data set consists of financial ratios from firms for three successive years, 1989–1991. Each case (instance) in the data set contains the values of the 19 ratios about a firm for a year and an indicator of whether or not the firm failed in the following year. Here the failure is defined as financial distress. A firm is in financial distress if it has entered bankruptcy under chapters 7 or 11 of the U.S. Bankruptcy code. The ratios representing the financial status of a firm are given in Table 1. The data set contains 1444 cases, 414 of which represent bankrupt firms. There are no missing values.

Table 1: Financial ratios used.

Ratio	Definition
CASH/TA	Cash / Total Assets
CASH/TS	Cash / Net Sales
CF/TD	C/F Oper. Income / Total Liabilities
CA/CL	Total Current Assets / Total Current Liabilities
CA/TA	Total Current Assets / Total Assets
CA/TS	Total Current Assets / Net Sales
EBIT/TA	(Interest Expense + Income Before Tax) / Total Assets
LOG(INT+15)	LOG((Interest Expense + Income Before Tax) / Total Assets +15)
LOG(TA)	LOG(Total Assets)
MVE/TK	Shrholder Equity / (Total Assets - Total Current Liabilities)
NI/TA	Net Income / Total Assets
QA/CL	(Total Current Assets - Inventories) / Total Current Liabilities
QA/TA	(Total Current Assets - Inventories) / Total Assets
QA/TS	(Total Current Assets - Inventories) / Net Sales
RE/TA	Retained Earnings / Total Assets
TD/TA	Total Liabilities / Total Assets
TS/TA	Net Sales / Total Assets
WK/TA	(Total Assets / Net Sales) / (Working Capital / Total Assets)
WK/TS	Working Capital / Net Sales

Table 2: Comparison of MDA, k-NN and FIL5 algorithms in terms of their performance on the data set.

Classifier	Accuracy %	Training (msec)	Classification (msec)
MAD	64.06	31	< 1
1-NN	69.4	6	944
10-NN	71.68	6	944
35-NN	73.14	6	944
50-NN	72.93	6	944
FIL5	77.57	4270	4

6. EXPERIMENTAL RESULTS

In order to evaluate the performance of the FIL5 classifier and compare it with MDA and k-NN techniques, we have implemented them in the C language, and ran on a personal computer with Linux operating system. Results are given in Table 2. To measure the performance, 10-fold cross-validation technique is used in the experiments. That is, the whole data set is partitioned into 10 subsets. The 9 of the subsets is used as the training set, and the tenth is used as the test set. This process is repeated 10 times once for each subset being the test set. Each value is the average of these 10 runs. This technique ensures that the training and test sets are disjoint, and every instances is classified exactly once.

These experiments indicate that k-NN algorithms attain better classification accuracy then MDA classifier. The accuracy of the k-NN classifier increases as the value of k increases up the $k = 35$, then decreases. The FIL5 classifier, on the other hand, achieves the best accuracy among the three class of algorithms we tested. The k-NN classifier has the lowest training time requirements, while the FIL5 classifier has the highest. However, the classification time requirements of the k-NN classifier is the highest, as expected. The classification time of the FIL5 algorithm is very low, close to that of MDA.

7. CONCLUSIONS

In this paper, a new form of feature projection based classification algorithm, called FIL5, has been presented. This algorithm has been compared with the classical approaches MDA and

k-NN, in terms of classification accuracy and time complexity on a real-world bankruptcy data set. Experiments indicate that the FIL5 classifier achieves the best classification accuracy and has very low classification time requirements. Therefor, not only for bankruptcy prediction as experimented here, but also for any domain where high classification accuracy and low classification time is required, the FIL5 algorithm can be applied successfully. It does not have any extra parameter, as the k of k-NN classifier, and it handles possible missing feature values very naturally. Also, due to the way the feature intervals are constructed, the FIL5 classifier is robust to noisy instances.

The algorithms discussed here aim at achieving high classification accuracy, that is low error rate in the prediction of unseen examples. However, these algorithms do not distinguish the types of errors. That is, for these algorithms, classifying a bankrupt case as a non-bankrupt has the same error as classifying a non-bankrupt case as a bankrupt. However, in real-life, these costs may not be the same for the decision maker. For example, the cost of predicting a case as non-bankrupt that is actually bankrupt may be higher than vice versa. Therefor, as a future work we plan to extend the FIL5 algorithm, so that it can take, as input, the cost of misclassification for all pairs of classes. The FIL5 algorithm can use this extra information in the formation of the feature intervals. Such an extension to the FIL5 algorithms will be very useful in domains such as bankruptcy prediction.

References

[1] A. Akkus, and H. A. Güvenir, "K Nearest Neighbor Classification on Feature Projections", Proceedings of ICML'96, Lorenza Saitta (Ed.), Morgan Kaufmann, Bari, Italy, 1996, pp. 12-19.

[2] R. E. Dorsey, R. O. Edmister and J. D. Johnson, "Bankruptcy Prediction Using Artificial Neural Systems", The University of Mississippi, School of Business.

[3] R. D. Duda, and P. E. Hart, "Pattern Classification and Scene Analysis", Wiley, 1973.

[4] H. A. Güvenir, G. Demiröz and N. İlter, "Learning Differential Diagnosis of Erythemato-Squamous Diseases using Voting Feature Intervals", Artificial Intelligence in Medicine, Vol. 13, No. 3, 1998, pp. 147–165.

[5] H. A. Güvenir and I. Sirin, "Classification by Feature Partitioning", Machine Learning, Vol. 23, No. 1, 1996, pp. 47–67.

[6] H. A. Güvenir and H. G. Koç, "Concept Representation with Overlapping Feature Intervals", Cybernetics and Systems: An International Journal, Vol. 29, No. 3, 1998, pp. 263–282.

[7] J. F. Hair, R. E. Anderson, R. L. Tatham, and W. C. Black, "Multivariate Data Analysis", Prentice-Hall Int. Inc., (Fifth Edition), 1998.

[8] J. R. Quinlan, "Unknown Attribute Values in Induction", Proceedings of the 16th International Workshop on Machine Learning, 1989, pp. 164-168.

[9] S. Salzberg, "Distance Metrics for Instance-Based Learning", Proceedings of ISMIS'91 6th International Symposium, Methodologies for Intelligent Systems, 1991, pp. 399-408.

[10] S. M. Weiss and N. Indurkhya, Predictive Data Mining: A Practical Guide, San Francisco: Morgan Kaufmann Publishers, Inc. 1998.