Does Feature Reduction Help Improve the Classification Accuracy Rates? A Credit Scoring Case Using a German Data Set

Jozef Zurada, University of Louisville, USA

ABSTRACT

The paper broadly discusses the data reduction and data transformation issues which are important tasks in the knowledge discovery process and data mining. In general, these activities improve the performance of predictive models. In particular, the paper investigates the effect of feature reduction on classification accuracy rates. A preliminary computer simulation performed on a German data set drawn from the credit scoring context shows mixed results. The six models built on the data set with four independent features perform generally worse than the models created on the same data set with all 20 input features.

Keywords: credit scoring, loan-granting decisions, data and feature reduction, Weka, data mining methods

1. INTRODUCTION

Knowledge discovery in databases (KDD) deals with finding patterns, rules, relationships, deviations, and rare events in massive amounts of data stored in relational databases and data warehouses. Though data can be generated and stored in the form of audio, video, images, pictures, text, and numbers, etc., we assume that data preprocessing has already been done and all relevant information for data mining is stored in a large flat file. The file may contain millions of samples, thousands of features, and hundreds/thousands of unique values that each feature may take. The assumption that the vast majority of data mining activities are performed on such large flat files containing mainly numeric features is consistent with the results of the survey published at www.kdnugets.com, one of the prominent web sites devoted to data mining topics. Files containing textual and time series data, as well as item sets (transactions), are also mined often. Other types of files are mined rather infrequently (Table 1).

Table 1: Types of Data Analyzed/Mined in [%] (from September 2009; multiple answers were allowed)

Table data (fixed num of columns)	80.0
Time series	45.3
Text free-form	37.9
Itemsets/transactions	28.4
Anonymized data	18.9
XML data	14.7
Web content	13.7
Social network data	12.6
Images/video	12.6
Spatial data	9.5
Other	9.5
Web clickstream	8.4
E-mail	8.4
Music/audio	7.4

Source: www.kdnuggets.com

This paper broadly discusses the data reduction issues. These involve sampling, feature reduction, and reduction of the number of values that each feature may take. Numerous experiments described in the literature show that these activities are desired as they reduce the dimensionality of data sets. As a result, predictive capability of the created models is generally improved, and models are easier to understand and explain as they contain fewer variables. The paper concentrates on one aspect of data reduction, i.e., feature reduction. The results from computer simulation performed on a German data set show, however, that feature reduction worsens the classification accuracy rates for five out of six models which we created.

Sections 2 and 3 deal with data reduction techniques and feature reduction, respectively. Section 4 describes the data set used in computer simulation. Feature reduction techniques available in Weka are presented in section 5. Results from computer simulation are depicted in section 6. Finally, section 7 provides conclusions and discusses possible future extensions of the paper.

2. DATA REDUCTION TECHNIQUES

The process of KDD involves several phases including data preparation; data reduction; data mining, which is at the heart of the KDD process; and interpreting the discovered knowledge to name a few. In this paper, we concentrate on a narrow aspect of the KDD process, i.e., data reduction issues which are often the most important and time consuming tasks in all KDD activities.

Large data sets often suffer from the "curse of dimensionality" problem which seems to be quite common in many encountered studies. Multiple dimensions could be visualized as a porcupine, with edges representing each dimension. When the data dimension grows, more data points are located on the edges of the porcupine, not at its center as one would desire. Thus, it appears that the majority of data points in a multidimensional space are outliers. To clarify this phenomenon further, one can state that the size of a data set yielding the same density of data points in an n-dim space increases exponentially with dimensions (the number of features). If a 1-dim sample containing n data points has a satisfactory level of density, then to achieve the same density of points in k dimensions, we need n^k points. For example, if n=100 data points in 1-dimension provides satisfactory density (domain coverage), then in k=3 dimensions one would need 100^3=1,000,000 data samples to achieve the same level of density. Due to this "curse of dimensionality", it is impossible to obtain a data set with the satisfactory level of data density where the domains for all variables are represented well (Kantardzic, 2003).

Data preparation and data reduction are the most critical steps in data mining. Their overall purpose is the reduction of dimensionality of the data set which involves three activities: feature reduction, sampling, and value transformation and reduction. Table 2 summarizes the methods used in these three activities. Data preprocessing and transformation and data reduction require a great deal of time and effort, but typically lead to better performance of the models in terms of their predictive or descriptive accuracy, diminishing of computing time needed to build models as data mining algorithms learn faster, and better understanding of the models. Data preprocessing methods depend on the types of values (numeric, categorical, nominal) that the variables store. It also depends on a specific data mining task, amount of data, and whether the data is temporal/dynamic (changes in time) or static (Pyle, 1999; Han and Kamber, 2002; Kantardzic, 2003).

Table 2: Data Reduction Techniques

Feature Reduction Methods	Sampling	Value Transformation
Data analyst judgment. Domain expert judgment. Principal component analysis. Feature selection based on comparison of means and variances. Entropy measures for ranking features. Correlation analysis. R^2 method. Chi Square. Decision trees. Regression with stepwise, forward or backward selection.	Systematic sampling. Random sampling with or without replacement. Stratified sampling. Average sampling Incremental sampling. Inverse sampling.	Handling of outliers using statistics. Distance-based methods. Deviation-based techniques. Normalization. Variable transformation. Binning. Smoothing. Feature discretization. Removing or replacing missing values.

3. FEATURE REDUCTION

In this paper we deal with one aspect of data reduction, i.e., feature reduction. The purpose of feature reduction is to eliminate irrelevant, correlated, and redundant features. Having a model operating with fewer features has important implications on future data gathering. Simply, in the next round of data collection, irrelevant features do not have to be collected.

Elimination of some features is obvious and one does not have to be the domain expert to do this. For example, social security number, employee id, zip code, last name, street address, columns containing constant values, etc. could be safely eliminated in most data mining applications as they do not have effect on the data mining results. SAS Enterprise Miner (EM) detects such features automatically and eliminates them by default. Other features could be chosen (or eliminated) with the help of the domain expert. For example, in the application of the real estate price assessment, a real estate agent or property tax assessor who are the domain experts, could advice the data analyst to select the following features for the model: location of the property, year built, size (in square feet), number of rooms, number of bathrooms, garage size, type of basement, distance to schools, distance to nearby stores, etc. Feature reduction could also be achieved by feature transformation. For example, one could easily convert two features representing the weight and height of the person to a single feature - the body mass index. In financial applications, dealing with firms' financial distress or loan granting decisions, the income/debt ratio is commonly used. Still, after the mentioned steps, the number of features can still be pretty large.

There are several algorithms for feature ranking and selection (Kantardzic, 2003). For example, features could be ranked according to some measure of statistical dependence or distance between samples. These measures do not tell one what the minimum set of features is used for analysis. They specify, however, the relevance of a feature compared to other features. Decision trees, for example, could rank features according to their relative importance, with the feature that has the most predictive power being placed at the top of the tree.

Feature selection is a space search problem. For a small number of features, one could build models for each combination of the features and check which combination produces the best results. For 3 features: A, B, and C, one would have $8=2^3$ combinations (subsets) of features only. These are: {- - -}, {- - C}, {- B -}, {- B C}, {A - -}, {A - C}; {A B -}; and finally {A B C}. Small number of features yields a small search space that can be searched exhaustively. For 20 features, however, there are 2^{20} of all possible subsets, yielding more than a million of possible combinations. In such situations, one needs to use heuristic search to obtain near optimal subset with the models' performances comparable to the full set of features. For example, the method of independent examination of features based on the means and variances compares two features only at a time without regard to other features. Another method based on a collective examination of features based on feature means and covariances is impractical and computationally prohibitive. It yields huge search space. Therefore, alternative heuristic methods for feature reductions are used. The most popular and very well-established method is the principal component analysis (PCA). It is, however, complex in terms of calculations. PCA can potentially reduce *m* features to *n* features, where $n \leq m$ with totally new values and with little loss of information. In other words, features which contribute the least to the variation in the data set are eliminated and features with the largest variation (those which have the most predictive power) are retained. Most statistical packages such as SAS, SPSS, Minitab, MatLab and data mining packages (SAS EM, Weka) are equipped with PCA. One interesting and effective feature reduction technique measures entropy, the concept borrowed from information theory. This technique is based on the approach that removing an irrelevant feature (or features) from a data set may not change the basic characteristics (the information content) of the data set. The algorithm is based on a similarity measure that is in inverse proportion to the normalized distance [0, 1] between two *n*-dim samples. If distance between samples is small, samples are considered similar; otherwise, if distance is large, samples are regarded dissimilar.

4. DATA SET USED IN COMPUTER SIMULATION

In computer simulation, we used open-source software Weka (www.cs.waikato.ac.nz/ml/weka/), an excellent and free tool for modeling and data reduction. We performed computer simulation on a German data set which contains 1,000 samples and 20 input attributes as well as 1 output attribute. The attributes describe financial, personal, and social attributes of loan applicants.

The data set contains the following attributes (expressed on nominal, ordinal, or interval scales): (1) *age* (of applicant in years), (2) *amount* (of credit requested), (3) *checking* (balance in existing checking account), (4) *coapp* (other debtors or guarantors), (5) *depends* (number of dependents), (6) *duration* (length of loan in months), (7) *employed* (length of present employment at present), (8) *existcr* (number of existing accounts at this bank), (9) *foreign* (foreign worker or not), (10) *history* (credit history), (11) *housing* (rent, own, free), (12) *installp* (debt as a percent of disposable income), (13) *job* (employment status), (14) *marital* (marital status and gender), (15) *other* (other installment loans), (16) *property* (collateral property for loan), (17) *purpose* (reason for loan request), (18) *resident* (years at current address), (19) *savings* (savings account balance), (20) *telephon* (telephone: none or registered under the customer's name), (21) *good_bad* (credit rating status: *bad* [loan denied] or *good* [loan granted]) – output variable. Out of 1,000 samples, 700 and 300 represented *good loans* and *bad loans*, respectively.

5. FEATURE REDUCTION TECHNIQUES AVAILABLE IN WEKA

Table 3 represents some of the feature reduction methods available in Weka. We performed feature reduction using all methods presented in Table 3. Depending on the techniques used, the methods returned between 4 and 16 significant attributes. All methods were very consistent, however, in retaining the following four attributes (out of 19 independent variables) as the most significant: *checking*, *duration*, *history*, and *employed*.

Table 3: Attribute reduction methods available in Weka. (For more, see Witten and Frank, 2005)

Method Name (in Weka)	Brief description of the algorithm
CfsSubsetEval	Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred (Hall, 1998). Identifies locally predictive attributes. Iteratively adds attributes with the highest correlation with the class as long as there is not already an attribute in the subset that has a higher correlation with the attribute in question.
ChiSquaredAttributeEval	Evaluates the worth of an attribute by computing the value of the chi-squared statistic with respect to the class.
Classifier subset evaluator	Evaluates attribute subsets on training data or a separate hold out testing set. Uses a classifier to estimate the 'merit' of a set of attributes.
ConsistencySubsetEval	Evaluates the worth of a subset of attributes by the level of consistency in the class values when the training instances are projected onto the subset of attributes. Consistency of any subset can never be lower than that of the full set of attributes; hence the usual practice is to use this subset evaluator in conjunction with a random or exhaustive search which looks for the smallest subset with consistency equal to that of the full set of attributes (Liu and Setiono, 1996).
GainRatioAttributeEval	Evaluates the worth of an attribute by measuring the gain ratio with respect to the class. GainR(Class, Attribute) = (H(Class) - H(Class | Attribute)) / H(Attribute).
InfoGainAttributeEval	Evaluates the worth of an attribute by measuring the information gain with respect to the class. InfoGain(Class,Attribute) = H(Class) - H(Class | Attribute).
PrincipalComponents	Performs a principal components analysis and transformation of the data. Use in conjunction with a ranker search. Dimensionality reduction is accomplished by choosing enough eigenvectors to account for some percentage of the variance in the original data---default 0.95 (95%). Attribute noise can be filtered by transforming to the PC space, eliminating some of the worst eigenvectors, and then transforming back to the original space.
SignificanceAttributeEval	Evaluates the worth of an attribute by computing the probabilistic significance as a two-way function (attribute-classes and classes-attribute association). For more information see Ahmad and Dey (2005). A feature selection technique used mainly for classificatory analysis.

6. THE METHODS USED AND RESULTS FROM COMPUTER SIMULATION

We used the following 6 models: logistic regression (LR), neural networks (NNs), radial basis function networks (RBFNNs), support vector machines (SVMs), *k*-nearest neighbor (*k*NN with 10 neighbors), and decision trees (DTs). We ran computer simulation for two scenarios. In each of the two scenarios, we performed 10-fold cross-validation to obtain independent and reliable error estimates on the test set. In scenario 1, we ran computer simulation on the reduced data set having the four mentioned independent attributes, and in scenario 2, we did it on

the entire data set having the 20 original independent variables. The classification accuracy rates for both scenarios are presented in Table 4. One can see that for five out of six models, attribute reduction worsened the overall classification accuracy rates, as well as the classification accuracy rates for *bad* loans. However, feature reduction improved the classification accuracy rates for *good* loans for five out of six models. Thus, if detecting *good* loans is more significant than detecting *bad* loans, variables reduction seems to be quite beneficial. In addition, features reduction makes the models simpler because rules that can be extracted from NN, RBFNN, and DT have fewer variables and are easier to interpret.

Table 4: Classification Accuracy Rates on the Test Set for Two Scenarios

Rates in [%]	**Methods**					
	LR	NN	RBFNN	SVM	*k*NN	DT
Scenario 1						
Overall	74.0	73.7	72.4	71.6	73.9	70.4
Good Loans	89.7	87.6	89.0	93.6	89.7	89.7
Bad Loans	37.3	41.3	33.7	20.3	37.0	29.7
Scenario 2						
Overall	75.6	75.2	73.0	75.8	73.6	70.9
Good Loans	86.7	85.4	86.6	87.7	93.1	84.1
Bad Loans	49.7	51.3	41.3	48.0	28.0	40.0

7. CONCLUSIONS AND FUTURE RESEARCH

Data reduction and, in particular, feature reduction are important steps in the KDD process. In general, it improves the predictive capability of the models and makes the models created simpler to interpret as they use fewer variables. We tested the effect of feature reduction on a single German data set drawn from a loan-decision context. The results from preliminary computer simulation are mixed and indicate that data feature reduction is not very beneficial in this case. More experiments with models containing different configurations of variables are needed for possible improvements of the results.

AUTHOR INFORMATION

Jozef Zurada earned his M.S. degree in Electrical Engineering at the Gdansk University of Technology, Gdansk, Poland, in 1972; and Ph.D. degree in Computer Science and Engineering from the University of Louisville, Louisville, Kentucky, USA, in 1995. Currently, he is a professor in the Computer Information Systems Department in the College of Business at the University of Louisville. He co-edited two books and published over sixty articles in refereed journals, book chapters, and conference proceedings, and delivered more than forty presentations to international and national academic and professional audiences. He teaches data mining and knowledge discovery, infrastructure technologies, and computer information systems. His main research interests include applications of data mining methods to solving challenging business and manufacturing problems.

REFERENCES

1. Ahmad, A., and Dey. L. (2005). A feature selection technique for classificatory analysis. *Pattern Recognition Letters*, 26(1), 43-56.
2. Hall, M.A. (1998). Correlation-based feature subset selection for machine learning. Ph.D. thesis, Department of Computer Science, University of Waikato, Hamilton, New Zealand.
3. Han, J. & Kamber, M. (2001). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers.
4. Kantardzic M. (2003). *Data Mining: Concepts, Models, Methods, and Algorithms*. IEEE Press/Wiley.
5. Liu, H. and Setiono, R. (1996). A probabilistic approach to feature selection - A filter solution. In: 13th International Conference on Machine Learning, 319-327.
6. Pyle, D. (1999). *Data Preparation for Data Mining*. Morgan Kaufmann.
7. Witten, I.H. & Frank, E. (2005). *Data Mining: Practical Learning Tools and Techniques*. Morgan Kaufmann Publishers.

NOTES