

Advances in Predictive Models for Data Mining

Se June Hong and Sholom Weiss

**To appear in
Pattern Recognition Letters Journal**

Advances in Predictive Models for Data Mining

Se June Hong and Sholom M. Weiss
IBM T.J. Watson Research Center
P.O. Box 218, Yorktown Heights, NY 10598, USA
sjhong@us.ibm.com and sholom@us.ibm.com

Abstract

Expanding application demand for data mining of massive data warehouses has fueled advances in automated predictive methods. We examine a few successful application areas and their technical challenges. We review the key theoretical developments in PAC and statistical learning theory that have lead to the development of support vector machines and to the use of multiple models for increased predictive accuracy.

Keywords: data mining, text mining, machine learning, boosting

1 Introduction

Predictive modeling, which is perhaps the most-used subfield of data mining, draws from statistics, machine learning, database techniques, pattern recognition, and optimization techniques. The proven utility of industrial applications has led to advances in predictive modeling. We will review a few application areas that have demonstrated the importance of predictive modeling and have also fueled advances. Some important advances in learning theory will be considered in section 3. The idea of using multiple models to enhance the performance of a model has spawned several useful approaches with a theoretical understanding for their success (section 4). Alternative ways of measuring predictive performance will be discussed in section 5.

2 Challenging applications

It has been well-established that a substantial competitive advantage can be obtained by data mining in general and predictive modeling in particular. For some applications, maximizing accuracy or another utility measure is of paramount importance, even at the expense of weaker explanatory capabilities. We will briefly examine three challenging application areas: insurance, fraud detection, and text categorization.

2.1 Insurance

Risk assessment is at the core of the insurance business, where actuarial statistics have been the traditional tools to model various aspects of risk such as accident, health claims, or disaster rates, and the severity of these claims. The claim frequency is rare and probabilistic by nature. For instance, the auto accident rate of an insured driver is never a clear no-accident class vs. accident class problem and instead is modeled as a Poisson distribution. The claim amounts usually follow a log-normal distribution which captures the phenomenon of rare but very high damage amounts. Neither of these distributions are well-modeled by conventional modeling tools such as CHAID, CART, C4.5, SPRINT or classical statistical regression techniques that optimize for traditional normal distributions. In general, different kinds of insurance can use different statistical models depending on the fundamental nature of the claims process, requiring a predictive model that can be optimized for different underlying distributions.

Other factors in insurance applications complicate the modeling process. Often, the desired target to be modeled is the expected claims amount for each individual policy holder, which is produced by a joint distribution of the claim rate and claim amount. Stored records usually have a significant proportion of missing data or back filled data updated only at the time of accidents. The claims data usually has hundreds of fields, and demographic data must also be included. Insurance actuaries demand that the model must be actuarially credible, i.e. the parameters of the model be within 5% of the expected true value with 90% confidence.

The Underwriting Profitability Analysis (UPA) application [1] embodies a new approach for generating predictive models for insurance risks. Groups of equal risk are identified by a top down recursive splitting method similar to tree generation algorithms. A key difference from traditional tree splitting is that the splits are selected by statistical models of insurance risk, e.g. joint Poisson and log-normal distribution for auto accident claim amount, otherwise known as pure premium. The methods tries to optimize the maximum likelihood estimation (in this case, negative log likelihood) of all the examples given the assumed distribution. This application has yielded demonstrably superior results for a major insurance firm, and many of the extracted rules, the leaf nodes of the trees, have replaced existing actuarial rules. They also report that in this application, the model improves as more data are used in the training set, contrary to many applications which reach a plateau of performance after tens of thousands of examples.

2.2 *Fraud detection*

Fraud detection is an important problem because fraudulent insurance claims and credit card transactions alone cost tens of billions of dollars a year. In the case of credit card fraud, artificial neural-networks have been widely-used by many banks. Frauds are relatively rare, i.e. a skewed distribution that baffles many traditional data mining algorithms unless stratified samples are used in the training set. Some large banks add to the transaction data volume by millions of transactions per day. The cost of processing a fraud case, once detected, is a significant factor against false positive errors while undetected fraud adds the transaction cost in the loss column. This not only influences the decision whether to declare a transaction to be processed as a fraud or not, but also calls for a more realistic performance measure than traditional accuracy. The pattern of fraudulent transactions varies with time, requiring relatively frequent and rapid generation of new models.

The JAM System (Java Agents for Meta-Learning) [2] is a recent approach for credit card fraud detection. The massive set of data with binary labels of fraud or legitimate transactions is divided into smaller subsets, for each participating bank unit and for multiple samples to gain better performance. They produce models by some fast existing methods in a distributed fashion. These multiple base models are then combined to form a meta-learner. (See sections 4 and 6). Using data from Chase and First Union banks, the induced models produced a substantial cost savings over existing methods.

2.3 *Text mining*

Electronic documents or text fields in databases are a large percentage of the data stored in centralized data warehouses. Text mining is the search for valuable patterns in stored text. When stored documents have correct labels, such as the topics of the documents, then that form of text mining is called text categorization. In many text storage and retrieval systems, documents are classified with one or more codes chosen from a classification system. For example, news services like Reuters carefully assign topics to news-stories. Similarly, a bank may route incoming e-mail to one of dozens of potential response sites.

Originally, human-engineered knowledge-based systems were developed to assign topics to newswires. Such an approach to classification may have seemed reasonable, but the cost of the manual analysis needed to build a set of rules is no longer reasonable, given the overwhelming increase in the number of digital documents. Instead, automatic procedures are a realistic alternative, and

researchers have proposed a plethora of techniques to solve this problem.

The use of a standardized collection of documents for analysis and testing, such as the Reuters collection of newswires for the year 1987, has allowed researchers to measure progress in this field. Substantial improvements in automated performance have been made since then.

Many automated prediction methods exist for extracting patterns from sample cases [3]. In text mining, specifically text categorization, the raw cases are individual documents. The documents are encoded in terms of features in some numerical form, requiring a transformation from text to numbers. For each case, a uniform set of measurements on the features are taken by compiling a dictionary from the collection of training documents. Prediction methods look at samples of documents with known topics, and attempt to find patterns for generalized rules that can be applied to new unclassified documents. Once the data is in a standard encoding for classification, any standard data mining method, such as decision trees or nearest neighbors, can be applied.

One of the interesting challenges text mining poses is the problem of minimal labeling. Most text collections are not tagged with category labels. Human tagging is usually costly. Starting from some tagged examples, one wishes to develop a text categorization model by asking certain selected examples to be tagged, and one naturally wishes to minimize the number of such requests. Many approaches to this problem are being pursued by theorists as well as practical algorithm developers. Another interesting application of text mining technology is web-mining where a variety of features beyond the original text present a special challenge.

3 Theoretical advances

The theory of predictive modeling sheds light on what kind of functions, i.e. mapping of feature vectors to the target values, can be learned efficiently with a given set of models. These results give an understanding of model complexity and how it can be used to assess the future performance of models on unseen data. These new concepts are beginning to guide the model search process as well as the model evaluation process for practical cases, complementing traditional techniques from statistics. For further details on these theoretical advances, see [4,5].

3.1 Computational and statistical learning theory

A model generation process can be viewed as selecting a “best” possible model, from a given family of models, i.e. functions that map input feature space to the target variable. A model is “best” if it optimizes the error rate or more generally a loss function defined over the example space and the predicted output. Computational learning theory is concerned with the complexity of such a process, but more focused on finding when the process can be efficient. One of the key areas in computational learning theory is the PAC, Probably Approximately Correct, learning model. Informally speaking, a concept from a given concept class is PAC learnable if a model can be found from a given model class such that the “error” of the model on the examples of the concept is bound by some given ϵ within a given confidence bound of δ . The learning algorithm is said to be efficient if the complexity is polynomial in the number of examples needed to learn the concept, $1/\epsilon$ and $1/\delta$.

An important PAC learning result shows that a weak PAC learner with ϵ less than $1/2$ can be turned into a strong learner with “error” close to 0, by multiple models, giving rise to a theoretical understanding of boosting (see section 4). Another interesting direction of the computational learning theory is to allow learning algorithm to make certain queries to an oracle about the data vectors. By asking for the probability of the example event within some given “noise”-tolerance, this approach makes it possible to analyze the learning problems in the presence of noise. This line of research has also been applied to the minimal labeling problem of text mining with some promising results.

Statistical learning theory has its origin in the work of Vapnik and Chervonenkis in the late 60s, who developed a mathematical basis for comparing models of different forms. The theory focuses on finite sample statistics (classical statistics usually rely on asymptotic statistics) in the process of determining what is the best among the given set of models in fitting the data. In classical statistics approach, it is assumed that the correct model is known and the focus is on the parameter estimation. Statistical learning theory focuses on estimating relative performance of competing models so that the best can be selected.

For predictive models, consider an example vector \mathbf{z} given as the input feature vector \mathbf{x} and the target value y . A model α operates on \mathbf{x} and predicts $f(\mathbf{x}; \alpha)$. If we are modeling the conditional probability distribution of y as a function of \mathbf{x} , an appropriate loss function $Q(\mathbf{z}, \alpha)$ of the model on \mathbf{z} is the same negative log-likelihood often used in classical statistical modeling: $Q(\mathbf{z}, \alpha) = -\log p(y|\mathbf{x}; \alpha)$. For classification error, the loss function $Q(\mathbf{z}, \alpha) = 0$ if $y = f(\mathbf{x}; \alpha)$ and 1 otherwise. If it is only known that the data vector \mathbf{z} is generated according to some given probability measure $F(\mathbf{z})$, the best model

would be the α that minimizes the expected loss

$$R(\alpha) = \int Q(\mathbf{z}, \alpha) dF(\mathbf{z}).$$

In practice, the probability measure $F(\mathbf{z})$ is not known, and one can only compute an empirical expected loss for the given example set $\mathbf{z}_i, i = 1, \dots, \ell$, assuming that they are *iid* generated:

$$R_{emp}(\alpha, \ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} Q(\mathbf{z}_i, \alpha).$$

Under what conditions does minimizing the empirical loss $R_{emp}(\alpha, \ell)$ also minimize the expected loss $R(\alpha)$ without knowing the distribution $F(\mathbf{z})$? Statistical learning theory answers this key question by offering the confidence regions for $R(\alpha)$ given $R_{emp}(\alpha, \ell)$ for a given lower bound of probability $1 - \eta$. These bounds are based on a measure known as VC-dimension of the set of models being considered. It suffices to say here that the VC-dimension might be considered to be a more reliable measure of the complexity of the model family than the degree of freedom concept used in the classical statistics. It directly implies that the number of examples should be much more than the VC-dimension to obtain a reliable model.

While the VC-dimension of a set of models is often very difficult to compute, the theory does offer a practical method for selecting the “best” model, when there is “enough” data, by use of randomly split data set into training and validation sets: the search for the best fitting model proceeds using the training set and the loss is estimated from the validation set using the confidence bound. The confidence bound can be expressed without explicit dependence on the VC-dimension for the loss in the validation set. This is similar to the cross validation method in classical statistics where the technique is often used in searching for the parameter values of a fixed model family.

3.2 Support vector machine

The support vector machine has the baseline form of a linear discriminator. Here we give a brief sketch of the support vector machine model. For a detailed introduction to the subject, see [6,7]. Let D be the smallest radius of the sphere that contains the data (example vectors). The points on either side of the separating hyperplane have distances to the hyperplane. The smallest such distance is called the *margin* of separation. The hyper plane is called optimal if the margin is maximized. Let ρ be the margin of the optimal hyperplane. The points that are distance ρ away from the optimal hyperplane are called the *support vectors*. It has been shown that the VC-dimension depends only on the number of support vectors. This implies that one can generate arbitrarily

many derived features, e.g. all pairwise products of the original features, as long as the number of support vectors for the optimal hyperplane (in the expanded dimensions) does not increase much. One can see that, although the final form of the model is a linear function, the decision boundary can be of almost arbitrary shape in the original feature space because of the nonlinearity introduced by derived variables.

This understanding leads to a new strategy to search for the support vectors and the coefficients for the optimal hyperplane simultaneously as an optimization problem in the expanded dimensional space. Actually, one need not explicitly compute the values of the derived features if they are chosen in a judicious way. The feature expansion makes use of several popular family of kernel functions, such as polynomial, radial basis functions or sigmoid functions as in the two layer neural network. Traditionally, such models with a linear outer function were constructed to minimize the error. To bound the error, support vector machines are optimized for the margin. Efficient search techniques for the optimal hyperplane and selecting the right basis function are active areas of research. The support vector machine is a significant new addition for predictive modeling.

4 Use of multiple models

Recent research results in learning demonstrate the effectiveness of combining multiple models of the same or different types for improving modeling accuracy. These methods, such as bagging [8] and boosting [9], have taken different approaches to achieve maximized modeling performance.

Let's look at an example where diverse predictive methods can be applied to obtain a solution. For example, a classification problem that can be solved by either a neural net method or a decision tree or a linear method. Until recently, the typical approach would be to try both methods on the same data, and then select the method with the strongest predictive performance. Researchers have observed that predictive performance often can be improved by inducing multiple solutions of the same type, for example multiple decision trees. These models are generated by sampling from the same data [11]. The final answer on a new case is determined by giving a vote to each of the multiple decision trees, and picking the answer with the most votes.

Although the techniques for generating multiple models from the same data are independent of the type of model, the decision tree is the most commonly used. How are the trees generated? No change is made to a standard tree induction method. The difference is in the sampling method. In the simplest approach, called bagging [8], a sample of size n is taken with replacement from

the original set of n examples. (For very large data, a smaller sample can be taken.) Some examples will be repeated in the sample, others may not occur. The expected proportion of unique examples in any given sample is 63.2%. Thus, it is possible to generate many samples, induce a decision tree from each sample, and then vote the results. An alternate sampling technique randomly selects a subset of features for each base model [10].

Adaptive resampling, usually performs better than bagging. Instead of sampling all cases randomly, so that each case has a $1/n$ chance of being drawn from the sample, an incremental approach is used in random selection. The objective is to increase the odds of sampling cases that have been erroneously classified by the trees that have previously been induced. Some algorithms use weighted voting, where some trees may be given more weight than others. The “boosting” algorithm, such as AdaBoost [9] uses weighted voting and an explicit formula for updating the likelihood of sampling or weighting each case in the training sample. While an ensemble of models does not necessarily increase the complexity of the solution in the sense of statistical learning theory, such a combined model diminishes the understandability that might have existed in a single model.

5 Practical performance measures

Although a measure of accuracy is generally useful for evaluating predictive models, the utility of the model’s output is a more direct goal. Models that optimize utility are widely available, for example users can enter a cost factors to CART. When the utility of a prediction can be computed in terms of the prediction statistics, it can be used in the model generation process in many different ways, e.g. in determining a split in a tree, in pruning process, etc. When the utility is not easily expressed in terms of computable measures with the model generation process, the negative log loss function is generally more useful than the accuracy measure.

Two alternative ways of evaluating model’s performance, *ROC curves* and *lift curves* are of interest. These are not new, but can offer insight into how different models will perform for many application situations. Many classification models can be modified so that the output is a probability of the given class, and hence depending on the threshold or some other decision making parameter value, one can get a family of models from one, for example a tree or a rule set. The Receiver Operating Characteristic (ROC) curve originated from signal detection theory. It plots the true positive rate (y-axis) against the false positive rate (x-axis). If there are two models in this space one can obtain any performance on the connecting line of the two just by randomly using the models with some probability in proportion to the desired position

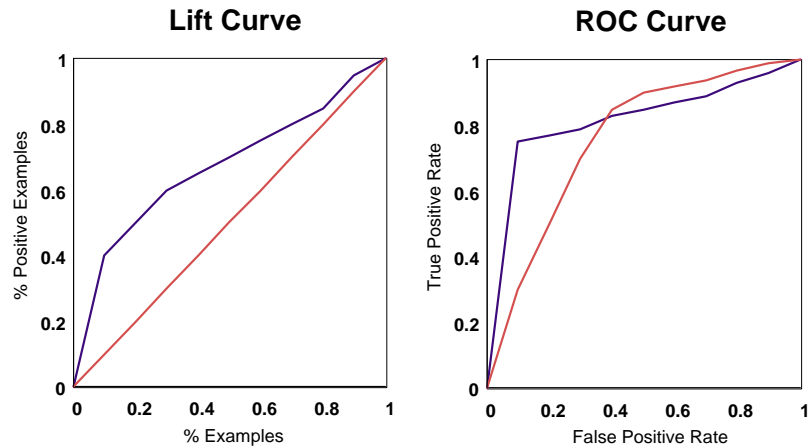


Fig. 1. Lift and ROC Curves

on the line. This curve allows one to select the optimal model depending on the assumed class distribution at the time of prediction. Any model that falls below the convex hull of other models can be ignored [12]. Figure 1 shows an example of a ROC curve and a lift curve.

For many applications, the aim of prediction is to identify some desired class members (e.g. customers) to whom some action (e.g. mailing advertisement circulars) is to be performed. Rather than classification, it is more flexible if the prediction is in the form of ranking based on the predicted class probability. The Lift curve then plots cumulative true positive coverage (y-axis) against the rank-ordered examples (x-axis). A random ranking will result in a straight diagonal line on this plot. A lift curve of a model is usually above this line, the higher the better for any given coverage of the examples in the preferential order.

6 Conclusion

We have presented an overview of some notable advances in predictive modeling. Clearly, this is a major area of interest to many research communities. Readers may think of other advances, not cited here, that are better-suited to other types of pattern recognition. Our emphasis has been on techniques for data mining of data warehouses, massive collections of data, such as the historical record of sales transactions. These data may require “clean up” but

the need for pre-processing is far less than many other pattern recognition tasks like image processing. It is no accident that gains in predictive performance have arrived in parallel with major enhancements of computing, storage, and networking capabilities. These rapid developments in computing and networking, along with e-commerce, can only increase interest in the theory and application of predictive modeling.

References

- [1] Apte C., Grossman E., Pednault E., Rosen B., Tipu F., White B, “Probabilistic Estimation Based Data Mining for Discovering Insurance Risks”, *IEEE Intelligent Systems*, Volume 14, Number 6, pp. 49-58, November/December 1999.
- [2] Stolfo S.J., Prodromidis A., Tselepis S., Lee W., Fan W. & Chan P., “JAM: Java Agents for Meta-Learning over Distributed Databases”, *Proc. of KDDM97*, pp. 74-81, 1997.
- [3] Weiss S. & Indurkha N., *Predictive Data Mining: A Practical Guide*, Morgan Kaufmann, 1998.
- [4] Hosking J.R.M., Pednault E.P.D. & Sudan M., “A Statistical Perspective on Data Mining”, *Future Generation Computer Systems: Special issue on Data Mining*, Vol. 3, Nos. 2-3, pp. 117-134, 1997.
- [5] Kearns M.J. & Vazirani U.V., *An Introduction to Computational Learning Theory*, MIT Press, 1994.
- [6] Vapnik V.N., *Statistical Learning Theory*, Wiley, 1998.
- [7] Cristianini, N. and Shawe-Taylor, J., *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.
- [8] Breiman L., “Bagging Predictors”, *Machine Learning*, Vol. 24, pp. 123-140, 1996.
- [9] Freund Y. & Schapire R., “Experiments with a New Boosting Algorithm”, *Proc. of the International Machine Learning Conference*, Morgan Kaufmann, pp. 148-156, 1996.
- [10] Ho, J., *The Random Subspace method for Constructing Decision Forests*, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 8, pp. 832-842, 1998.
- [11] John G., Kohavi R. & Pflieger K., “Irrelevant features and the subset selection problem”, *Proc. 11th International Conf. on Machine Learning, ICML-94*, pp. 121-129, 1994.
- [12] Provost F., Fawcett T., & Kohavi R., “The Case Against Accuracy Estimation for Comparing Induction Algorithms”, *KDDM98*, 1998.