Journal of Data Science **1**(2003), 391-404

Boosting Applied to Classification of Mass Spectral Data

K. Varmuza[1], Ping He[2,3] and Kai-Tai Fang[2]
[1] *Vienna University of Technology,*
[2] *Hong Kong Baptist University*
and [3] *Sichuan University*

Abstract: Boosting is a machine learning algorithm that is not well known in chemometrics. We apply boosting tree to the classification of mass spectral data. In the experiment, recognition of 15 chemical substructures from mass spectral data have been taken into account. The performance of boosting is very encouraging. Compared with previous result, boosting significantly improves the accuracy of classifiers based on mass spectra.

Key words: Boosting, data mining, decision tree, mass spectra.

1. Introduction

Identification of compounds or automatic recognition of structural properties from mass spectral (MS) data has been attracted by many authors in chemometrics. Commonly, there are two classes of methods for identifying compounds: one is library search methods based on spectra similarities; another is classification methods. A number of mass spectral library search algorithms are offered and have been routinely used in identification. These algorithms perform well when the substance that needs to be identified is included in the library. At present, the number of compounds in the available libraries is limited (about 100,000 spectra in the NIST 98 MS Database and about 130000 spectra in the Wiley/NBS collection), while there are more than twenty million chemical compounds described by Chemical Abstracts

Service. So the spectrum of an unknown substance often deviates considerably from the spectra in the reference library and the unknown can not be identified directly. In this case the library search method may lose some efficiency. On the other hand, identification of an unknown compound that is not in the library can be supported by classification methods indicating the probabilities of presence or absence of certain chemical substructures or compounds classes. So numerous classifiers based on MS data and multivariate statistics have been developed for automatic recognition of substructures and other structural properties. Part of them are efficient enough to be used together with automatic isomer generation for a systematic and exhaustive structure elucidation (Varmuza and Werther 1996). However, there are still many substructures which can not be recognized efficiently by existing classifiers because the relationship between MS data and chemical structures is too complex to be detected. So seeking a better technique for mass spectral pattern recognition has being a mission in chemometrics.

A number of classification methods have been applied in the analysis of mass spectra. Linear discriminant analysis (LDA) is one of the methods firstly applied to mass spectral data, because its decision rule is simple to implement and describe. However, this method has severe limitations, because of arithmetic problems caused by collinearity in the high dimension mass spectral data. Principal component analysis (PCA) and partial least squares (PLS) are used to deal with this problem via abstracting most principal components of predictors (Werther *et al.* 2002). However, classifiers based on PCA or PLS are weak on detecting the local character of data and modeling irregular decision boundaries. When the relationship between MS data and chemical structures is complex, such classifiers may be not efficient. Recently a tree-based method, classification and regression tree (CART), has attracted attention. The method, which generates a tree structure though recursively splitting the predictor space until the space is completely partitioned into a set of non-overlapping subspaces, is effective in capturing the local character and the complex interaction. Another advantage is its interpretability. In spite of these advantages, its instability may make interpretation somewhat precarious because small changes in the data may lead to a complete different decision rule. Neural networks (NN) based on artificial intelligence principles is another method which has al-

ready been widely used for the classification of mass spectral data (Klawun and Wilkins 1996). It has the powerful capability of a non-linear separation of classes. However, overfitting is dangerous and interpretation of the model is difficult.

In order to identify compounds correctly, high predictive abilities of the classifiers are very important. All the methods mentioned above have been applied to mass spectral data for constructing classifiers. However, for many substructures, the prediction accuracy of these classifiers are far from satisfied, and an improvement is highly desirable. Boosting, the effective ensemble method proposed by Freund and Schapire (1996, 1997), is regarded as the "best off-the-shelf classifier in the world" (Breiman 1998). It is a procedure that combines the outputs of many classifiers to produce a powerful "committee" of the classifiers obtained by training certain versions of the training sample. In our knowledge the boosting technique has never been applied in classification of MS data. In this paper, we apply boosting method to mass spectral data for the automatic recognition of 15 substructures. Compared with the result obtained by a given single classifier or previous classifiers, the classifiers obtained by boosting are indeed more accurate in prediction.

The paper is organized as the follows. Section 2 introduces the method of boosting tree. The used mass spectral data sets are described in Section 3. The result of experiments are shown and discussed in Section 4. The last section gives a conclusion.

2. Methodology

2.1 Algorithm of boosting

The boosting used in this paper is the most popular boosting algorithm called "AdaBoost.M1." (Adaboost for short) (Freund and Schapire 1997). Suppose there is a training sample with N observations $(x_{(i)}, y_{(i)})$, each $x_{(i)} = (x_{i1}, \cdots, x_{ip})$ having p predictor variables, and $y = 1$ for category I and $y = -1$ for category II. A classifier $\mathrm{G}(\mathbf{x})$ can be obtained by applying a classification algorithm, for example LDA, to the training sample. And given a vector of the predictor variables $\mathbf{x} = (x_1, \cdots, x_p)$, it will produce a

prediction by taking one of the two values -1, 1. Then the fitting error rate of the classifier on the training sample is defined as

$$\overline{err} = \frac{1}{N}\sum_{i=1}^{N} I(y_i \neq G(x_{(i)}) \tag{1}$$

where I(A) is the indicator function, I(A)=1 if A is true, otherwise I(A)=0. In order to reduce the error rate, boosting sequentially applies the classification algorithm to repeatedly modified versions of the training data, thereby producing a sequence of classifiers $G_m(\mathbf{x}), m = 1, \cdots, M$. Then the final prediction can be obtained from the combination of these classifiers by a weighted major vote $G(\mathbf{x}) = \text{sign}(\sum_{m=1}^{M} \alpha_m G_m(\mathbf{x}))$, where $\alpha_1, \cdots, \alpha_m$ are the weights of classifiers and sign$(\cdot)$ is the sign function, sign$(A) = 1$ if A>0 , otherwise sign$(A) = -1$. Figure 1 shows a schematic of the AdaBoost procedure. The process of averaging weighted classifiers not only reduces the fitting error rate but also protects against overfitting (Freund *et al.* 2001) The algorithm of Adaboost can be implemented as follows (Hastie *et al.* 2001).

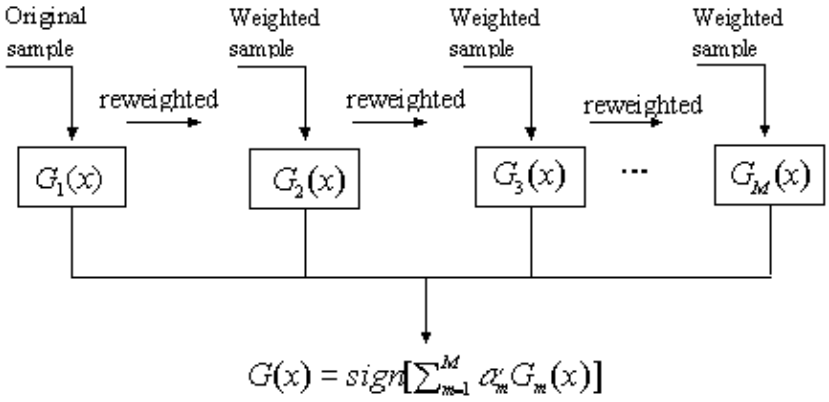

Figure 1. The schematic of the AdaBoost procedure

1. Initialize the observation weights, $w_i = 1/N, i = 1, \cdots, N$ and choose a positive integer M.

2. For m=1 to M
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute $err_m = \sum_{i=1}^{N} w_i I(y_{(i)} \neq G_m(x_{(i)}))/\sum_{i=1}^{N} w_i$.
   (c) Compute $\alpha_m = \log((1 - err_m)/err_m)$.
   (d) Let $w_i = w_i \exp[\alpha_m * I(y_{(i)} \neq G_m(x_{(i)}))], i = 1, ..., N$.

3. Output $G(x) = \text{sign}[\sum_{m=1}^{M} \alpha_m G_m(x)]$.

At each step, boosting modifies the model by applying weights $w_1, \cdots w_N$ to each of training observations $(x_{(i)}, y_{(i)})$, that is, the higher the weight, the more the observation influences the classifier learned. Initially all samples are assigned the same weight $w_i^1 = 1/N$, so that the first step simply trains the classifier on the data in the usual manner. At each round $m = 1, \cdots, M$, the classifier $G_m(\mathbf{x})$ is constructed by applying the classification algorithm to the samples with weight w_i^m and the error of this classifier (err_m) is also measured with respect to the weights. The determination of w_i^m is according to the rule that the weights of those observations that were misclassified by the classifier $G_{m-1}(\mathbf{x})$ are increased, whereas the weights are decreased for those that were classified correctly. The re-weighting architecture reflects the idea that the modified version of the data makes the next learner focusing on the samples hard to classify. At last, the final classifiers $G(\mathbf{x})$ are obtained by a weighted majority vote and these weights $\alpha_1, \cdots, \alpha_M$ are the monotonic decreasing function of $err_m, m = 1, \cdots, M$. They reflect that the more accurate classifiers in the sequence, the higher influence they will have. As far as the choice of M, the number of iteration in boosting, although there are theoretical analysis about it (Freund and Schapire 1997), these tend not to give practical answer. So in practice, the number of rounds M is determined by cross validation. The reasons for success of boosting in classification can be described as follows. Firstly, one can see boosting is an ensemble method that averages the models, which are produced by applying certain classification algorithm to different version of training samples. The way of averaging models can reduce the variance of the finally model and enhance the prediction ability (Hastie *et al.* 2001). Secondly, the re-weighted

process makes the successive classifier focus on the observations which are misclassified by the previous one, so that the final model is also effective for the sample which is hard to be classified. The Adaboost approach also has a good theoretical fundament. It has been proved by Friedman *et al.* (2001) that the final classifiers obtained by Adaboost.M1 is the optimum solution to the forward stepwise additive modeling with the loss function $L(y, G(\mathbf{x})) = \exp(-yG(\mathbf{x}))$.

2.2 Boosting tree

From subsection 2.1., we know the boosting strategy can be applied to different basic classification techniques. In this subsection we briefly introduce boosting combined with a decision tree. A decision tree as proposed by Breiman *et al.* (1984) is a powerful classification tool in data mining. It generates a tree structure though recursively splitting the predictor space until the space is completely partitioned into a set of non-overlapping subspaces. In the process of growing a tree, the splitting rule which includes splitting variables x_j and the corresponding split point λ can be automatically selected according to a certain criterion, for example, minimizing cross-entropy/deviance (Hastie *et al.* 2001). When the split rule is obtained, the data can be partitioned into two regions $x_j > \lambda$ and $x_j < \lambda$, and then the splitting process in the sub-regions is repeated until all the stopping criteria are satisfied. The details of the algorithm have been published by Breiman *et al.* (1984).

One of the disadvantages of decision tree is its instability because of the hierarchical nature of the process. The effect of an error in the top split will be propagated down to all other splits below. The instability can be alleviated by combining sequential trees. In the process of boosting decision tree, the successive trees can be achieved by directly fitting the weighted data. It is fulfilled via altering the criterion of finding the split pairs (variable, point) and the stopping criteria to the weighted criteria. For example, suppose a region R_t including N_t observations $(x_{(i)}, y_{(i)})$, the deviance of this region is defined as

$$deviance = -\sum_{k=-1,1} \hat{p}_{tk} \log \hat{p}_{tk}, \tag{2}$$

here

$$\hat{p}_{tk} = \frac{1}{N_t} \sum_{x_{(i)} \in R_t} I(y_{(i)} = k). \tag{3}$$

Then the weighted criterion can be described

$$deviance_w = - \sum_{k=-1,1} \hat{p}_{tkw} \log \hat{p}_{tkw}, \tag{4}$$

and

$$\hat{p}_{tkw} = \frac{1}{\sum_{x_{(i)} \in R_t} w_i} \sum_{x_{(i)} \in R_t} w_i \cdot I(y_{(i)} = k). \tag{5}$$

Other steps of boosting can also be directly implemented according with Adaboost.

3. Mass spectral data

3.1. Mass spectrometry

Mass spectrometry is a commonly used instrumental technique for the characterization and identification of chemical organic compounds. In a mass spectrometer molecules of the investigated sample are ionized and the produced ions are separated according to their mass-to-charge ratio (m/z, mostly z=1), and their abundances are measured. A mass spectrum can be graphically represented as a bar plot with m/z (mass of ions) versus abundance of ions (peak height). For example, Figure 2 shows a mass spectrum of C_2H_4O, called acetaldehyde. The distribution of peaks in a mass spectrum is very characteristic for a compound, although not unique. Main information obtained from a mass spectrum are molecular mass, and hints about substance class and parts (substructures) of the molecular structure.

In our work, mass spectra and chemical structures are taken from the NIST mass spectral database (NIST 1992) that contains more than 62000 compounds. Substances considered for this work are restricted to the molecular mass range 50-400, with elements C, H, N, O, S, Si, and halogens allowed, resulting in 50286 compounds. A set of 15 substructures as described in Table 1 are used for defining the development of spectra classifiers

(Yoshida *et al.* 2001). In this paper, for each substructure two random samples were generated: a class 1 set with 300 different substances containing the substructure, and a class 2 set with 300 different substance not containing the substructure. For the substructure searches and the elimination of structure duplicates software ToSiM and SubMat (Varmuza and Scsibrany 2000) have been used.

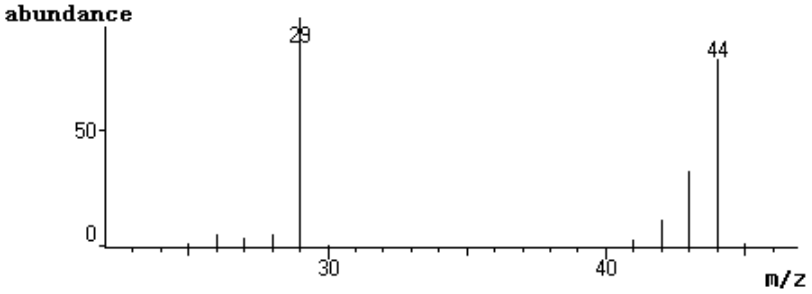

Figure 2. Mass spectrum of C_2H_4O

3.2 Spectral feature

Previous work shows that the generation of suitable spectral features is a crucial step in the development of classifiers (Werther *et al.* 1994). Spectral features are variables obtained by (mostly nonlinear) transformations of mass and abundance data. Appropriate spectral features are simpler related to molecular structures than the original peak heights of mass spectra (Varmuza 2000 and Werther *et al.* 2002). In this work, a set of 400 spectral features have been used as summarized in Table 2 (Yoshida *et al.* 2001). All used features are in the range from 0 to 100 and can be automatically calculated from a low resolution mass spectrum (mass and abundance values).

4. Results and Discussion

In the experiment, Boosting tree is applied to the 15 mass spectral data sets. As described in section 3, every data set contains 600 substances

Table 1: Substructures used for spectral classifiers

No.	Substructure
1	$(CH_3)_3C$ (tertiary butyl)
2	(C,C,C)C-OH (tertiary alcohol)
3	CH_3-O-CH_2 (methyl ether)
4	CH_3-C=O (acetyl)
5	CH_3-COO (acetoxy)
6	O=C-OCH_3 (methyl ester)
7	$(CH_3)_2N$ (dimethyl amine)
8	C_6H_5 (phenyl)
9	Benzene ring with -CH_2 (benzyl)
10	Benzene ring with -O
11	C_6H_5-CH_2-O
12	Benzene ring with -N
13	Benzene ring with -Cl
14	Cl in molecule
15	$(CH_3)_3Si$ (trimetyl silyl)

Table 2: Spectral feature

Group	Feature description	feature numbers
1	Modulo-14 summation for mass ranges m/z 31-800, 31-120, and 121-800	1-42
2	Spectra type features describing the distribution of peaks	43-45
3	Logarithmic intensity ratios of m/z 39-109 with mass differences of 1 and 2	46-187
4	Autocorrelation for mass differences of 1,2, and 14-51 in mass ranges m/z 31-800, 31-120, and 100-800, respectively	188-307
5	Peak intensities (% base peak) at masses m/z 31 and 3-120	301-396
6	Averaged intensities in mass ranges m/z 33-50, 51-70, 71-100, and 101-130	397-400

Table 3: The result of boosting tree and PLS for test sample

sub-structure No.	PLS (test sample)			Desicion tree (test sample)			Boosting tree test sample		
	P_1(%)	P_2(%)	P(%)	P_1(%)	P_2(%)	P(%)	P_1(%)	P_2(%)	P(%)
1	79	88	84	74	76	75	81	87	84
2	64	73	69	69	69	69	76	74	75
3	86	92	89	84	82	83	88	91	89
4	69	85	77	78	79	79	84	85	84
5	80	84	82	80	79	79	87	83	85
6	72	83	78	76	72	74	82	81	82
7	79	83	81	79	75	77	85	83	84
8	72	81	77	78	77	77	87	82	84
9	83	70	77	70	71	71	81	74	78
10	82	70	76	72	70	71	82	76	79
11	94	93	94	90	91	90	95	94	94
12	75	76	76	69	68	69	80	75	78
13	91	89	90	87	86	87	94	93	94
14	78	89	84	84	87	85	88	92	90
15	97	97	97	92	93	93	95	97	96

belonging to two categories: one with the substructures present denoted by class 1, the other with the substructures absent denoted by class 2. Each substance has 400 spectral features/predictors. Three indices are used to evaluate the classifiers: P_1 is the correctly classified rate from class 1, P_2 is the correctly classified rate from class 2 and $P = (P_1 + P_2)/2$. In the procedure of this experiment, randomly select 200 observations from class 1 and 200 from class 2 have been used as training sample and the remains as test sample. For boosting and decision tree applications, We repeated the random partitioning procedure 100 times and got the averaged values of indices as the final results. For PLS only one experiment was performed for each classifiers. Table 3 shows the results of test samples obtained by boosting tree, decision tree and PLS which has been applied to the same data sets before. (Note: all the results have been rounded to integer). From this table, we can see the boosting tree significantly improves the predictive ability of single decision tree and yields the better results than PLS.

A graphical comparison of the predictive abilities for the boosting and PLS is presented in Figure 3. The horizontal coordinates of the three plots are in turn the values of three indices measuring PLS's predictive ability and the vertical coordinates are those of Boosting tree. From this plot, we can see for class 1, the boosting tree classification gives much higher predictive abilities than PLS; for class 2 both methods are equal. In summary boosting tree is better than PLS.

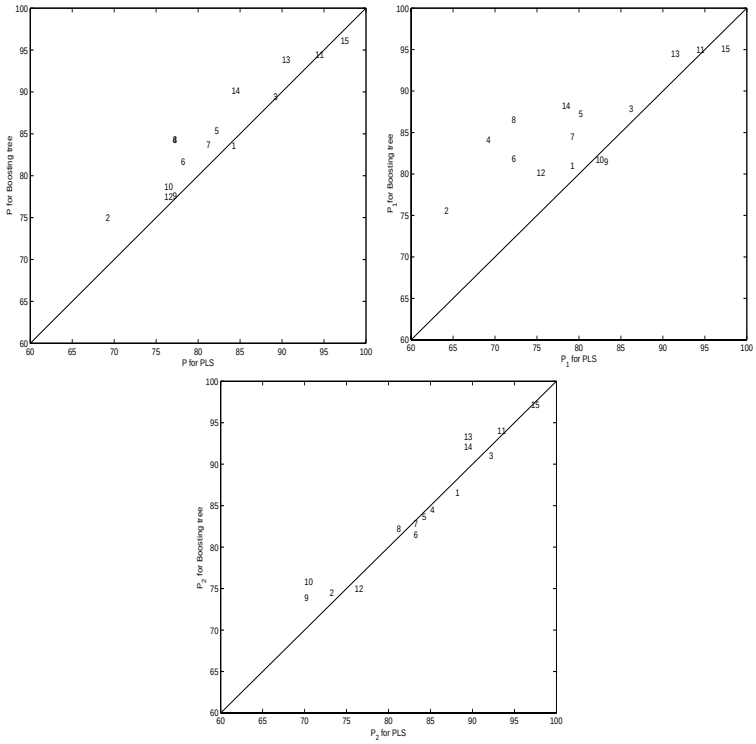

Figure 3: Comparison of the predictive abilities for PLS and boosting tree

5. Conclusion

Improving the accuracy of prediction of classifiers based on mass spectra is important for chemical structure elucidation. In this paper, the boost-

ing tree is applied to mass spectral data for classifying presence/absence of substructures in compounds with unknown chemical structure. Experiments show that the boosting tree indeed can find classifiers with higher predictive ability than obtained with PLS classifiers. This approach is new for mass spectra classification. There are many work worth for further developing, for example, how to interpret the relationship between substructure and mass spectrum using the model established by boosting.

References

Breiman, L. (1998). Arcing classifiers. *Annals of statistics*, **26**, 801-849.

Breiman, L., Friedman, J.H., Olsen, R.A. and Stone, C.J. (1984). *Classification and Regression Tree.* Chapman & Hall, New York.

Freund, Y. and Schapire, R.E. (1996). Experiments with a new boosting algorithm. In *machine learning: Proceeding of the Thirteenth International Conference.*

Freund, Y. and Schapire, R.E. (1997). A decision theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Science*, **55**, 119-139.

Freund, Y., Mansour, Y. and Schapire, R.E. (2001). Why averaging classifiers can protect against overfitting. *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics.*

Friedman, J.H., Hastie, T. and Tibshirani, R. (2001). Additive Logistic Regression: a statistical view of boosting, *Annals of Statistics*, **28**, 337-307.

Hastie, T., Tibshirani, T.R. and Friedman, J.H. (2001). *The Elements of statistical Learning- Data Mining, Inference, and prediction.* Springer, New York.

Klawun, C. and Wilkins, C.L. (1996). Joint neural network interpretation of infrared and mass spectra. *J. Chem. Inf. Comput. Sci.*, **36**, 249-257.

NIST (1998). *NIST'98 Mass Spectral Database.* National Institute of Standards and Technology, Gaithersburg MD 20899, USA.

NIST (1992). *NIST Mass Spectral Database*. National Institute of Standards and Technology, Gaithersburg MD 20899, USA.

Varmuza, K. and Scsibrany, H. (2000). Substructure isomorphism matrix. *J. Chem. Inf. Comput. Sci.*, **40**, 308-313

Varmuza, K., in: Lindon, J.C., Tranter, G.E. and Holmes, J.L. (Eds.), (2000). Chemical structure information from mass spectrometry. *Encyclopedia of Spectroscopy and Spectrometry*, Academic Press, London.

Varmuza, K. and Werther, W. (1996). Mass spectral classifiers for supporting systematic structure elucidation. *J. Chem. Inf. Comput. Sci.*, **36**, 323-333.

Werther, W., Lohninger, H., Stancl, F. and Varmuza, K. (1994). Classification of mass spectra. A comparison of yes/no classification methods for recognition of simple structural properties. *Chemometrics and Intelligent Laboratory Systems*, **22**, 63-76.

Werther, W., Demuth, W., Krueger, F.R., Kissel, J., Schmid, E.R. and Varmuza, K. (2002). Evaluation of mass spectra from organic compounds assumed to be present in cometary grains. Exploratory data analysis. *J. Chemometrics*, **16**, 99-110.

Wiley (1998) Wiley/NBS Mass Spectral Database 4th edition: Electronic Publishing Division, John Wiley & Sons, Inc. New York

Yoshida, H., Leardi, R., Funatsu, K. and Varmuza,K. (2001). Feature selection by genetic algorithms for mass spectral classifiers. *Analytica Chimica Acta*, **446**, 485-494

Received May 20, 2003; accepted August 8, 2003

Varmuza, K.
Laboratory for Chemometrics, Institute of Chemical Engineering
Vienna University of Technology, Vienna, Austria
Kvarmuza@email.tuwien.ac.at

Ping He
Department of Mathematics

Hong Kong Baptist University
Kowloon Tong, Hong Kong, P. R. China
01400894@hkbu.edu.hk

Kai-Tai Fang
Department of Mathematics
Hong Kong Baptist University
Hong Kong, P. R. China
ktfang@hkbu.edu.hk