Classification Using Mass Spectrometry Proteomic Data with Kernel-Based Algorithms

Zhenqiu Liu and Shili Lin *

Abstract

Motivation: Early detection of cancer is crucial for successful treatment, and protein profiling using mass spectrometry (MS) data has been investigated as a potential tool. However, due to the high correlation and huge dimensionality of MS data, it is crucial to modify existing algorithms and to develop new ones, where necessary, for analyzing such data.

Results: We develop a group of logistic regression - kernel coupling algorithms for classification of normal versus cancer samples. Furthermore, we propose a systematic three-step protocol for analyzing MS data, from removal of baseline noise and normalization, to feature extraction and reduction, and finally to classification. The systematic analysis paradigm and the proposed algorithms were applied to an ovarian and a prostate cancer dataset. The results show that our proposed approaches can be an effective tool for analyzing MS proteomics data.

Keywords: Proteomics; Mass Spectrometry; peptide; Classification; Logistic Regression; Kernel methods

1 Introduction

Proteomics is very important in understanding biological systems and uncovering disease mechanisms. Because of their high level of variability and complexity, it is an extremely challenging endeavor to conduct massive analysis of thousands of proteins. In the last decade, mass spectrometry (MS) has increasingly become the method of choice for analyzing complex protein samples. A mass spectrometer measures two properties of ion mixtures in the gas phase under a vacuum environment: the mass-to-charge ratio (m/z) of ionized proteins in the mixture, and the numbers of ions present at different m/z values (Roboz, 2002).

*Department of Statistics, The Ohio State University, Columbus, OH 43210, USA and University of Maryland, Greenebaum Cancer Center, 22 South Greene Street, Baltimore, MD 21201, USA. Tel: 410-328-7828 Email: zliu@umm.edu

The output is a mass spectrum or chart with a series of spike peaks, each representing the ions of a specific m/z value in the sample. The volumes of peaks and the m/z values of the peaks are a fingerprint of the sample (Samuelsson *et al.*, 2004). Mass spectrometry has not only been used extensively to identify proteins via peptide mass fingerprints, but also found promising applications in cancer classification (Adam *et al.*, 2002; Petricoin *et al.*, 2002; Lilien *et al.*, 2003; Qu *et al.*, 2003, Wu *et al.*, 2003; Zhu *et al.*, 2003).

While MS is increasingly used for protein profiling, significant challenges have arisen with regard to analyzing the data generated. Specifically, for cancer classification based on MS data, the analysis can be divided into three steps: data preprocessing, feature selection, and classification. The critical preprocessing step includes baseline correction, peak identification and alignment, data normalization, and visualization. The feature selection (dimension reduction) step extracts features, and further reduces the number of features by means of statistical testing or heuristic methods. The final step is the classification of disease status based on selected features. Each step in this three-step protocol is critical and can have a non-negligible effect on final performance, although recent publications on cancer classification with MS data have largely focused on feature selection and comparison of various classification methods (Adam *et al.*, 2002, Lilian *et al.*, 2003). For example, the t-test, principal component analysis (PCA), and partial least square methods (PLS) (Liu *et al.*2004,2005) have been proposed for the selection of relevant features. For the final step, classification methods such as linear discrimination analysis, k-nearest neighbor classification, decision trees (Adam, *et al.*, 2002), and support vector machines have been used to distinguish between cancer and normal samples (Adam *et al.*, 2002, Lilien, *et al.*, 2003, Qu *et al.*, 2003).

In this paper we investigate a number of methods for each of the three steps in this analysis paradigm; we focus on how analysis in each of the three steps may

affect the performance as a whole. In particular, novel kernel-based analysis procedures for classification of MS data are proposed. These methods use kernel matrix as an input instead of the original data, which can compress the MS data greatly when the sample size is much smaller than the input dimension. Several methods for data preprocessing, such as baseline removal and data normalization, are discussed. Feature extraction algorithms with first order derivatives and wavelet coefficients are also studied in detail. We assess the performance of different combinations of steps and methods using several publicly available MS datasets.

This paper is organized as follows. In the next three sections, we discuss, in turn, methods for data pre-processing, feature extraction and selection, and classification. These are followed by applications of the methods to real data. Additional discussions and concluding remarks are provided at the end.

2   Data Pre-processing Methods

A MS dataset with M samples can be represented by a $p \times (M + 1)$ matrix: $(mz, X) = [mz, \mathbf{x}_1, \ldots, \mathbf{x}_M]$ where p is the number of m/z ratios, mz is a column vector denoting the measured m/z ratios, and the vector $\mathbf{x}_j$ contains the corresponding intensities of the jth samples, $j = 1, \cdots, M$. Let $\{y_1, \ldots, y_M\}$ be the set denoting the cancer status of the samples. For the n $(< M)$ training samples, their corresponding y values are known, whereas the labels are unknown for test samples. Our goal is to predict the label y_j for each of the test samples based on the data for the training samples and the intensity profiles for the test samples.

MS data are non-stationary sequential data with very high dimensionality. The consecutive values of MS data are usually highly correlated, and thus there is a lot of redundancy in the dataset. MS data can also be contaminated by noise. Data preprocessing has to be done to remove the noise and the correlations within a sequence. In addition, in order to classify multiple sequences into different classes, we need to define a similarity (distance) measure between different sequences. It is easy to see that Euclidean distance is not a good measurement of sequence similarity with the following simple example. Consider three sequences $A = (3, 8, 3, 8, 4), B = (6, 4, 5, 6, 6)$, and $C = (13, 18, 13, 18, 14)$ shown in Figure (1). Note that shifting A up by 10 units coincides with C. However, under Euclidean distance, A is more similar to
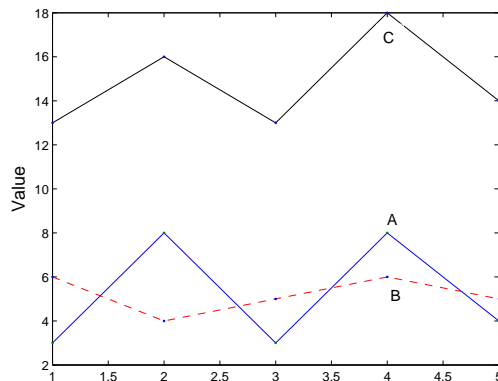

Figure 1: Shortcoming of the Euclidean Distance

B than to C. In order to overcome the shortcoming of Euclidean distance in this type of applications, the minimum distance measure (Lee, 2003) is often used to gauge sequence similarity. The minimum distance can give a better estimation of similarity between two sequences with similar trends running at two different levels. Given two sequences $A = (a_1, a_2, \ldots, a_p)$ and $B = (b_1, b_2, \ldots, b_p)$, the minimum distance is defined as follows:

$$D(A, B) = \Big(\sum_{i=1}^{p} |a_i - b_i - \mu|^k \Big)^{\frac{1}{k}},$$

where $\mu = \sum_{i=1}^{p} (a_i - b_i)/p$. For k=2, the minimum distance is Euclidean distance with normalization. We can first shift both sequences into zero means (normalization) and then calculate the Euclidean distance. Distances with other k values can also be defined in a similar fashion.

For our purpose of analyzing proteomics data, we adopt the following normalization strategy by first mapping all intensities to be within the 0-1 range and then obtaining mean-corrected values. Specifically, let $\mathbf{x}' = (x_1, x_2, \ldots, x_p)$ be a given input MS sample/profile/feature. The mapping function $\mathbf{w} = (w_1, w_2, \cdots, w_p)' = (\mathbf{x} - \min(\mathbf{x}))/(\max(\mathbf{x}) - \min(\mathbf{x}))$ scales all intensities to be within 0 and 1, where $\min(\mathbf{x}) = \min\{x_1, \cdots, x_p\}$, and $\max(\mathbf{x})$ is defined similarly. Then the mean-subtracted values $\mathbf{z} = \mathbf{w} - \sum_i w_i/p$ is treated as a normalized sample.

Removal of baseline

Another task in the preprocessing step is the removal of baseline wandering, which often appears in the ac-
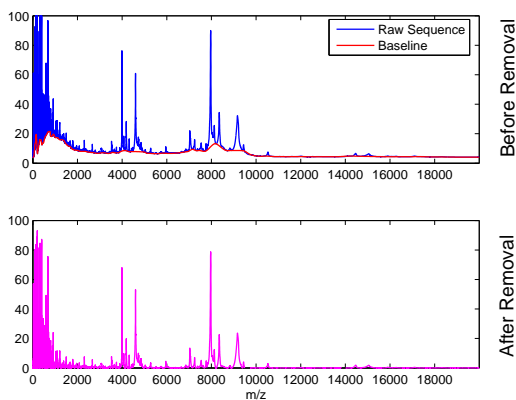
Figure 2: An MS profile before and after removal of baseline

Figure 3: A segment of MS profile (after normalization) and the corresponding features of first order derivatives.

quisition stage due to various reasons. Other tasks such as peptide detection and classification, can be affected by it. Baseline removal can be done either before or after normalization. It is better, however, to remove the baseline first, since it is just a low frequency noise and has nothing to do with the usefulness of peptides. There are many techniques for baseline removal in the engineering literature, including lowpass or highpass filters, wavelets, and polynomial interpolation (Froning *et al.*, 1988). Here we borrow the polynomial interpolation technique for this biological application. This method first detects the characteristic points of the sequence (minimum value for each small window), then interpolates either with a linear, quadratic, or cubic spline function. Finally, the estimated value of the spline is subtracted from the original signal for each m/z ratio to obtain the baseline-corrected intensity. Figure 2 shows one MS sample before and after the baseline removal. In this example, the baseline is interpolated with a cubic spline with a window size of 100.

Feature Extraction and Reduction

We have discussed data preprocessing for removal of baseline noise and data normalization. In this section, we discuss how to extract and reduce features either in the time, or the frequency domain. Since consecutive data points of a sequence are highly correlated, we propose the use of first order derivatives for feature extraction, and we will compare its performa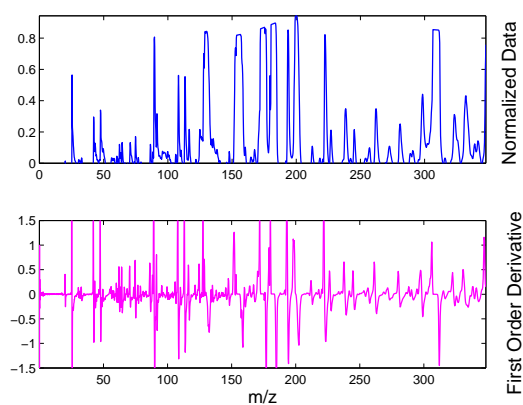nce with a popular approach in the literature, wavelet decomposition. For feature reduction, we propose to use a simple statistical testing procedure.

Feature extraction

Our proposed feature extraction method is based on the first order derivative of the intensity profile. The idea of using first order derivatives is to reduce the redundancy in the MS sequence and correlation, the assumption being that features exhibiting low correlation should lead to a better performance. The first order derivatives (estimates) of input vector $\mathbf{x}_j$ is defined as follows:

$$\mathbf{z}_j(i) = \frac{1}{2}\left[\frac{\mathbf{x}_j(i+2) - \mathbf{x}_j(i-2)}{mz(i+2) - mz(i-2)}\right] + \\ + \frac{1}{2}\left[\frac{\mathbf{x}_j(i+1) - \mathbf{x}_j(i-1)}{mz(i+1) - mz(i-1)}\right], \ i = 1 \dots p.$$

As an example to show that the peaks in the first order derivative series are much clearer than in the original data, we plotted a segment of a normalized MS sample and its first order derivative counterpart in Figure 3.

There are of course other ways to define estimates of first order derivatives, but they tend to give similar results and will not be pursued here. We also note that there are many other potential methods for extracting relevant features, which have been discussed extensively in the signal and image processing literature (Farid *et al.*, 2004, Silapachote *et al.*, 2004). For example, features can also be defined by using second order derivatives, or some combinations of derivatives

in the first and second orders. Alternatively, multiplication of backward differences can also be an effective means of extracting features by considering consecutive first order derivative. More advanced methods also exist, including linear segmentation with dynamic programming. These more elaborative algorithms can usually extract out more accurate features, but their computational costs are much greater, which diminishes their desirability as efficient tools for analyzing proteomics data given the huge dimensionality. It is precisely this computational cost consideration that propels us to opt for the simple first order derivative approach, which will be shown to perform reasonably well.

For comparison purposes we also investigate the performance of using a discrete wavelet transformation for feature extraction, as this is a popular approach in proteomics analysis (Qu *et al.*, 2003). Wavelets are families of functions that can accurately describe other functions in a parsimonious way with a small number of nonzero transform coefficients. A simple and commonly used one is the Haar wavelet (Qu *et al.*, 2003), which is adopted for our usage here because of its simplicity.

Feature reduction

The size of the feature matrix for the training samples extracted with either the first order derivatives or wavelet decomposition methods are still $p \times n$, the same as that of the original feature (MS profile) matrix, but with many zeros or near zeros in them. To further reduce the size of input data for the classification step, we propose the use of a simple t-test procedure to select only a small subset of the features that are deemed to contribute the most to the discrimination between the two classes. Let μ_j^+ and σ_j^+ be, respectively, the mean and standard deviation of the ith feature over $n+$ class one (cancer) samples in the training set. Define μ_j^-, σ_j^-, and $n-$ similarly for class two (normal) samples. Then the t-test statistic for the ith feature is calculated as follows:

$$t_j = \frac{|\mu_j^+ - \mu_j^-|}{\sqrt{\frac{(\sigma_j^+)^2}{n^+} + \frac{(\sigma_j^-)^2}{n^-}}}.$$

These t statistics and/or their associated p-values can then be ordered and an appropriate number of them be selected for the classification analysis. One advantage of a statistical testing procedure like the one based on the above t-test (or any other appropriate statistical tests) is that it is usually very fast. The disadvantage compared with wrapping methods is that

relevant features may be left out or unimportant ones may sneak in, depending on what tests are being used. Also, features selected may still include highly correlated and redundant ones.

3 Kernel Based Algorithms

Kernel methods such as support vector machines (SVM) have been used extensively and successfully for various classification tasks (Burges 1998, Zhu and Hastie 2005). It is well known that SVM use kernel matrix instead of the original sequences as input. More generally, for a problem in which the training sample size n is much smaller than the input dimension p, such as that of proteomics data, using a kernel matrix has an apparent advantage as the dimension of the kernel matrix is $n \times n$. Logistic regression is a popular classification tools in statistics, as it can provide associated probabilities in addition to assignments of class labels, but it is not applicable to proteomics data directly since $n \ll p$. It is the desire to use logistic regression for MS data analysis that motivated us to develop new algorithms that combine logistic regression with kernel matrix to circumvent the high-dimensionality problem.

A kernel function is defined as an inner product of two vectors in feature space. Therefore, one needs to first transform each input vector $\mathbf{x}$ from the original input space F_0 into a higher dimensional feature space F_1 with a transformation $\Phi : \mathbf{x} \to \Phi(\mathbf{x})$ before performing the inner product operation (Burges 1998, Rosipal and Trejo 2001). Polynomial kernels defined as follows constitute a class of kernel functions among the popular ones:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i'\mathbf{x}_j + p_2)^{p_1},$$

where the two input vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ are the original (not transformed) data for subjects i and j, although they may have been preprocessed. The parameters p_1 and p_2 are both nonnegative integers, specifying the form and degree of polynomials in the transformation from original input to feature vectors. In particular, $(p_1 = 1, p_2 = 0)$ and $(p_1 = 2, p_2 = 1)$ lead to a linear and a second order kernel, respectively, which are the two kernel functions that will be used in this paper.

We define in the following a kernel matrix

$$K = \begin{pmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \cdots & K(\mathbf{x}_1, \mathbf{x}_n) \\ \cdots & \cdots & \cdots \\ K(\mathbf{x}_n, \mathbf{x}_1) & \cdots & K(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix},$$

which are of dimensions $n \times n$. It is easily seen that K is the Gram matrix under linear polynomial kernel, that is, $K = X'X$. Discussion of the properties of the Gram matrix, in particular its relationship with the covariance matrix $C = XX'$ and principal component analysis, can be found in Rosipal and Trejo (2001).

We propose to use three algorithms for proteomics data analysis that combines logistic regression with kernel matrix, namely, logistic kernel regression (LKR), logistic kernel principal component regression (LKPCR), and logistic kernel partial least square regression (LKPLSR). Input data to the algorithms include a training dataset $\mathbf{X} = \{\mathbf{x}_j\}_{j=1}^n$ with known class labels $\mathbf{y} = \{y_j\}_{j=1}^n$ and a test dataset $\{\mathbf{x}_t\}_{t=1}^{n_t}$ whose class labels are to be predicted. All three of these algorithms (with detailed step-by-step operations deferred to the Appendix and Supplementary Information) carry out logistic regression in the feature space. More specifically, LKR makes use of all transformed features, while LKPCR tries to first find principle components before performing logistic regression based on them. Although the components included explain much of the variance, the approach does not take into account the labeling of the training samples. In the worst case scenario it is possible to leave out exactly the components that are needed to distinguish between the different groups of samples in the classification step. In the applications explored in this paper, though, LKPCR was able to preserve much of the information contained in the initial data.

Nevertheless, to circumvent this drawback, we also consider LKPLSR, a method discussed in Rosipal and Trejo (2001). Unlike principal component analysis, partial least square finds components from the input data $\mathbf{X}$ that are also relevant for the output (labels) $\mathbf{y}$. More specifically, partial least square searches for a set of components that perform a simultaneous decomposition of both $\mathbf{X}$ and $\mathbf{y}$ with the constraint that these components explain as much as possible of the covariance between $\mathbf{X}$ and $\mathbf{y}$. The number of components k used in the model can be selected based on Akaike's information criteria (AIC) as detailed in the Appendix.

Finally, in addition to the three kernel-based algorithms outlined above, we also include the support vector machine in our evaluation and comparison of methods.

4   Results

Ovarian cancer

We evaluate the proposed analysis steps and associated algorithms first with an ovarian cancer dataset. The data were downloaded from http://home.ccr. cancer.gov/. The dataset includes 91 controls and 162 ovarian cancer samples, each measured at 15154 data points (m/z ratios and associated intensities). We randomly selected 45 and 81 of the control and cancer samples, respectively, into a training set and used the rest as a test set to assess the performance of the proposed methods. We replicated this experiment 100 times. For each of the replicates, we performed several analyses, depending on the combinations of the first two analysis steps and associated methods used. More specifically, for each of the four kernel-based methods discussed in the previous section, we used the original raw data (None), data preprocessed with only baseline removal (BR), as well as with both BR and normalization (DP). These three sets of analyses were designed to evaluate the performances of these four classification algorithms with at most minimally preprocessed data, focusing on the effect of normalization using the proposed scaling and mean correction method. We then considered the effects of the two data extraction methods, first order derivatives in addition to DP (DP+FOD), and discrete wavelet transformation (also in addition to the same DP; DP+DWT). The performances of these two analyses were then compared and contrasted with only using the t-tests for feature selection (reduction) after subjecting to the same DP (DP+FR). Finally, we considered two more sets of analyses that used all proposed steps for data processing before inputing to the classification step, namely, DP+FOD+FR, and DP+DWT+FR. For all the analyses whose results are given in table 1, the kernels used are second order polynomials with parameters $p_1 = 2$ and $p_2 = 1$. Furthermore, the number of principal components in LKPCR was chosen to be 30, while the number of partial least square components in LKPLSR was chosen to be 11 determined by the AIC procedure described in the Appendix. For both LKR and SVM, the regularization constant was set to be $\lambda = 1$ (see Supplementary Information).

The average prediction accuracy (percentage) and the standard deviation among the 100 test sets are given in Table 1.

The most significant feature emerged from the table is the importance of normalization in addition to

Table 1: Performances of four algorithms under various preprocessing and/or feature selection/reduction schemes. The mean and SD are the average and standard deviation over 100 replications.

Algorithm	Stat	None	BR	DP	DP +FOD	DP +DWT	DP +FR	DP +FOD +FR	DP +DWT +FR
LKPCR	Mean	87.9	86.3	95.8	98.3	97.3	98.3	99.9	99.7
	SD	6.7	3.6	1.8	1.3	1.8	1.2	0.2	0.2
LKPLSR	Mean	92.4	89.7	98.6	99.1	99.3	98.9	100	100
	SD	4.3	4.7	1.6	0.7	0.5	1.5	0.0	0.0
LKR	Mean	72.6	88.5	96.7	98.2	98.4	97.6	99.9	100
	SD	2.4	2.3	1.9	1.4	1.5	1.3	0.1	0.0
SVM	Mean	74.5	87.6	95.3	98.5	98.7	97.9	100	99.9
	SD	5.7	4.6	2.1	1.1	1.2	2.2	0.0	0.2

baseline removal in data preprocessing. It is somewhat surprising to see that, for classification algorithms LKPCR and LKPLSR, without any data preprocessing actually outperformed their counterparts with only baseline removal. On the other hand, for LKR and SVM, even just preprocessed with BR had considerable effects on the performances. Also, the performance of the algorithm without any data preprocessing can be much more variable. Both of the feature extraction methods led to further improvements, and they are deemed to perform similarly. Applying feature reduction without feature extraction (the DP+FR column) seemed to do almost as well compared to feature extraction with feature reduction (the DP+FOD and DP+DWT columns), although with slightly smaller accuracy rates for three of the four classification algorithms. Using all proposed data processing steps yielded the best results, as expected. Although the two feature extraction method performed similarly, FOD is perhaps a better choice from a computational efficiency standpoint.

In terms of comparisons of the four classification algorithms, they all performed comparably well, with the partial least square algorithm LKPLSR being the best (or tied for the best) in all of the seven sets of analyses carried out. It is also worth noting that, for LKPCR to obtain comparable results with LKPLSR as shown in Table 1, more components (30) were need. Finally, it seems that the SVM benefits more from the feature extraction step than most of the other algorithms.

As mentioned earlier, our algorithms can not only predict class labels for test samples, but also the probabilities associated with the calls to measure the degree of confidence. To illustrate this feature and to com-
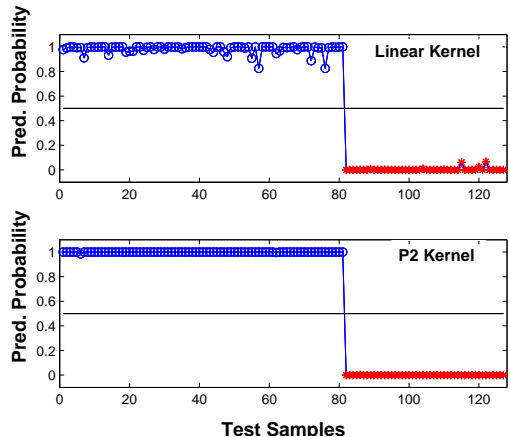

Figure 4: Average predictive probabilities for samples in test sets with linear or second order (P2) polynomial kernels.

pare the second-order polynomial kernel, used in all the analyses shown in table 1, with a linear polynomial kernel ($p_1 = 0, p_2 = 1$), we plot the average (out of 100 sets of test samples) predictive probabilities of being labelled cancer for each of the test samples under LKPLSR. Note that the first 81 samples in each test set were cancerous while the remaining were normal. As can be seen from the figure, the labels of all test samples were predicted correctly under both kernels. However, although the confidence for such predictions was uniformly high for all test samples under the second-order kernel, the confidence was somewhat lower for some of the test samples under the linear kernel.

Prostate cancer

The prostate cancer data were also downloaded from the same web site where we obtained the ovarian data. The data consist of three classes: normal, benign, and prostate cancer with 63, 190, and 69 samples, respectively. As with the ovarian data, we randomly chose half of the samples in each of the three classes to form a training set and used the rest as a test set. The experiments were also carried out 100 times for evaluation and comparison. All proposed data processing steps (DP+FOD+FR, with feature extraction performed using first order derivatives) were performed before classification analysis. A "one-against-all-others" scheme was applied to predict the class label of each test sample. Specifically, for each test sample, the probability of each class label (using the corresponding training set) was calculated. Then the class label with the largest probability among the three competing ones was the predictive class of the test sample. The kernel for this analysis was the same second-order polynomial used in the analysis of the ovarian data. For LKPCR, the number of principal components used was 40 while the number of partial list square components was 23, again chosen based on AIC. The regularization constant was set to be $\lambda = 1$ as in the analysis of the ovarian data.

Table 2 shows the average prediction accuracy (percentage) and the associated standard deviation among the 100 test sets. Four sets of prediction accuracy were assessed: among cancer, normal, and benign samples, as well as among all samples combined. These results painted a similar picture as the one that emerged from the ovarian analysis for comparison of the four algorithms considered, although the results are not quite as good across the board. In other words, the four algorithms performed comparably, with LKPLSR being the best overall and best for predicting the healthy and benign samples. For predicting the cancer samples, the performance of LKPLSR came in second, but only less than half of a percentage point behind LKR.

5   Discussion

In this paper, we systematically examined the effect of each step in a three-step protocol in proteomic profiling for cancer prediction. It has been recognized that data preprocessing and feature selection are two important steps before the final step of classification, but there has not been systematic evaluation of their effects on the performances of classification methods. Part of the purpose of our study is to fill this void.

Furthermore, we have proposed the use of several kernel-based algorithms as alternatives to the popular support vector machine approach.

Our experiments with two cancer datasets show that baseline removal and data normalization are indeed indispensable for data preprocessing, and the proposed methods for these two tasks seem to work adequately. In addition to polynomial interpolation, there are other methods for baseline removal in the signal processing literature that may be worth exploring (Froning *et al.*, 1998). Similarly, there are alternatives for data normalization (Colantuoni *et al.*, 2002), which could potentially outperform our scaling and mean-correcting procedure. There are also appreciable improvements with the feature selection step even after data preprocessing, although these additional improvements are relatively much smaller compared to those seen with the preprocessing step. Of the two feature extraction procedures, the first-order derivatives approach appears to be preferable. Although the performances of first order derivatives and wavelet transformation are comparable, the former is computationally much more efficient. It also has the advantage of extracting features in the time rather then the frequency domain, enabling visual examination of the m/z values associated with the peaks explicitly (figure 2).

The three algorithms that combine logistic regression with kernel matrix are promising. Their performances in terms of prediction accuracy rates are comparable to those using the popular SVM, with LKPLSR outperforming all the others (including SVM) in almost all of the situations considered. One advantage of these algorithms over SVM is the additional probabilistic information regarding class assignments, which can be used to assess prediction strength. In a way, one can view the three logistic regression - kernel coupling algorithms as nonlinear versions of logistic regression that can handle the large-*p*-small-*n* problem, a setting not amenable with the standard logistic regression approach. A plausible explanation for LKPLSR seemingly outperforming LKPCR is that the principle components selected under LKPCR try to explain the variance in the features only while the partial least square approach under LKPLSR attempts to strike a balance between explaining the variance in the features and finding correlation with the corresponding labels. In other words, the latter makes use of information provided by the known class labels of the training samples in addition to the data from the explanatory variables.

Table 2: Performances of four algorithms for overall and individual group classifications. The mean and SD are also based on 100 replicates.

Algorithm	Stat	Overall	Normal	Benign	Cancer
LKPCR	Mean	89.8	93.5	85.3	88.3
	SD	2.56	2.76	2.48	3.27
LKPLSR	Mean	91.9	97.6	85.9	90.7
	SD	1.92	3.1	2.12	2.86
LKR	Mean	90.4	95.8	84.1	91.1
	SD	2.38	3.54	2.51	2.51
SVM	Mean	90.2	95.8	84.6	90.6
	SD	2.76	2.16	2.95	3.76

Of the two datasets to which the analysis procedures and algorithms were applied, the prediction accuracy rates are higher for the ovarian cancer, although the results from both datasets are very good. There are a number of possible explanations, including data quality and sample size in each class. From a statistics standpoint, the multiclass prostate cancer dataset poses more challenges than the two-class ovarian data. It would be interesting to see whether a different approach than the "one-again-all-others" scheme may give better results. Also, the prostate cancer class is in fact composed of two subtypes, thus analyzing them as a single cancerous class as done here might not be the best way as there may be non-negligible amount of heterogeneity between the subtypes. These are some of the issues that will be studied in a future investigation.

Our exploration thus far indicates that mass spectrometry measures, together with data preprocessing, feature selection (dimension reduction), and classification algorithms could be an effective tool for the reliable prediction of cancer samples. However, our enthusiasm is dampened somewhat by recent debates in the literature regarding the reproducibility of mass spectrometry data with the current technology (see Baggerly et al. 2004, 2005, and references therein). To understand the issue further, we also analyzed the same two ovarian datasets (including the one analyzed in detail above which gave excellent results) considered in Baggerly et al. (2005). We used one of the datasets to train the classifiers that were in turn used to predict the labels of the samples in the other dataset. Our results indicate that, regardless of which data were used as a training set, the prediction accuracy suffered due to the low correlation between the two datasets of the samples of the same class. In

fact, the highest correlation (albeit still quite low) was achieved between the cancer samples in one dataset and the normal samples in the other, beating out the correlations between samples of the same class. The same pattern was observed by using instead the minimum distance measure discussed in an early section. This example re-enforces the notion that statistical analysis cannot be fruitful unless quality and consistent data across different experiments are available. Despite these reservations regarding data quality at the moment, we believe that the proposed algorithms have their values as they are easy to implement and can have high prediction accuracy for data with high quality.

Appendix

Outlines of the steps for each of the three algorithms (LKR, LKPCR, and LKPLSR) are given below. Detailed algorithms and programs implementing them can be found in the Supplementary Information. Step 1&2 in the following is common to all three algorithms:

Step 1: Given a training dataset $Z = \{\mathbf{z}_i\}_{i=1}^n$ with known class labels $\mathbf{y} = \{y_i\}_{i=1}^n$ and a test data $\{\mathbf{z}_t\}_{t=1}^{n_t}$, compute the kernel matrix $K = [K_{ij}]_{n \times n}$ for the training data, where $K_{ij} = K(\mathbf{z}_i, \mathbf{z}_j)$. Compute the kernel matrix $K_{te} = [K_{ti}]_{n_t \times n}$ for the test data similarly. K_{ti} projects the test data $\mathbf{z}_t$ onto training data $\mathbf{z}_i$ in the higher dimensional feature space in terms of the inner product.

Step 2: Centralize K and K_{te} using

$$K^c = (\Phi - \bar{\Phi})'(\Phi - \bar{\Phi}) = \left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n'\right)K\left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n'\right),$$

and

$$
\begin{aligned}
K_{te}^c &= (\Phi_{te} - \bar{\Phi}_{te})'(\Phi - \bar{\Phi}) \\
&= \left(K_{te} - \frac{1}{n}\mathbf{1}_{n_t}\mathbf{1}_n'K\right)\left(\mathbf{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n'\right).
\end{aligned}
$$

where $\mathbf{1}_n$ is the vector of n 1's, I_n is an identity matrix of $n \times n$, Φ and Φ_{te} are the transformed feature matrices of the training and test data, respectively, and $\bar{\Phi}$ and $\bar{\Phi}_{te}$ are the respective averages across the columns of Φ and Φ_{te}. In general, we do not need to define the transformation function explicitly to compute the inner product K and K_{te} (Cristianini et al. 2000). This is an advantage of kernel methods.

The specific algorithm for each of LKR, LKPCR, and LKPLSR are given below.

Logistic Kernel Regression Algorithm

Step 1: (see above)

Step 2: (see above)

Step 3: Set $\mathbf{p} = (p_1, \ldots, p_n)'$ with logistic function $g(s) = (1 + exp(-s))^{-1}$ and $p_i = g(\sum_{j=1}^n \beta_j K(\mathbf{z}_i, \mathbf{z}_j)) = f(\mathbf{z}_i, \boldsymbol{\beta})$, where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)'$ are the logistic regression coefficients to be estimated iteratively. Let V denote the diagonal matrix $\mathrm{diag}[p_1(1 - p_1), \ldots, p_n(1 - p_n)]$. Set the regularization parameter λ and the stopping criterion ϵ. Initialize $\boldsymbol{\beta}^{new} = \mathbf{0}$ and
Do

- $\boldsymbol{\beta}^{old} = \boldsymbol{\beta}^{new}$

- $p_i = f(\mathbf{z}_i, \boldsymbol{\beta}^{old})$ for $i = 1, 2, \ldots, n$

- $D_{\boldsymbol{\beta}^{old}} = ((\mathbf{y} - \mathbf{p})'K^c)' - \lambda\boldsymbol{\beta}^{old}$

- $H_{\boldsymbol{\beta}^{old}} = -((K^c)'VK^c + \lambda I)$

- $\boldsymbol{\beta}^{new} = \boldsymbol{\beta}^{old} - H_{\boldsymbol{\beta}^{old}}^{-1}D_{\boldsymbol{\beta}^{old}}$

While $|\boldsymbol{\beta}^{new} - \boldsymbol{\beta}^{old}| > \varepsilon$ At the end of the loop, we obtain the coefficient vector estimate $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^{new}$. In all the applications in this paper, we set the regularization parameter $\lambda = 1$ as $\lambda = 1$ given the best test error for $\lambda \in [0, 10]$. For the stopping rule, we set $\epsilon = 10^{-6}$ to compromised between accuracy and efficiency.

Step 4: Predict results for test data using K_{te}^c and $\boldsymbol{\beta}$. $p_{te} = g(\sum_{j=1}^n \beta_j K_{te}^c(\mathbf{z}_{te}, \mathbf{z}_j))$.

Logistic Kernel Principal Component Regression Algorithm

Step 1: (see above)

Step 2: (see above)

Step 3: Form a $n \times k$ matrix $U = [u_1 \; u_2 \; \cdots \; u_k]$, where $u_1, u_2, \ldots, u_k$ are eigenvectors of K^c that correspond to the largest eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k > 0$. Also form a diagonal matrix D with $1/sqrt\lambda_i$ in position (i, i).

Step 4: Find the projections $\mathbf{V} = K^cUD$ and $\mathbf{V}_{te} = K_{te}^cUD$ for the training and test data, respectively, to form the k principal components.

Step 5: Build a usual logistic regression model using $\mathbf{V}$ and $\{y_i\}_{i=1}^n$ and use it to predict the class labels of the test samples using $\mathbf{V}_{te}$.

Logistic Kernel Partial Least Square Regression Algorithm

Step 1: (see above)

Step 2: (see above)

Step 3: Call KPLS algorithm to find k component directions (Rosopal and trejo 2001).

1. Set $K^d = K^c$

2. for $i = 1, \ldots, k$, let $\mathbf{u}_i$, $\mathbf{t}_i$, and $\mathbf{c}_i$ be some latent vectors, we initialize $\mathbf{u}_i^{new}$ randomly
   Do

   - $\mathbf{u}_i^{old} = \mathbf{u}_i^{new}$
   - $\mathbf{t}_i = K^d\mathbf{u}_i^{old}$, $\mathbf{t}_i \leftarrow \mathbf{t}_i/||\mathbf{t}_i||$.
   - $\mathbf{c}_i = \mathbf{y}'\mathbf{t}_i$
   - $\mathbf{u}_i^{new} = \mathbf{y}\mathbf{c}_i$, $\mathbf{u}_i^{new} \leftarrow \mathbf{u}_i^{new}/||\mathbf{u}_i^{new}||$

   While $|\mathbf{u}^{new} - \mathbf{u}^{old}| > 10^{-6}$.

3. deflate K^c, $\mathbf{y}$ by $K^d \leftarrow \prod_{l=1}^i(I - \mathbf{t}_l\mathbf{t}_l')K^c\prod_{l=1}^i(I - \mathbf{t}_l\mathbf{t}_l')$ and $\mathbf{y} \leftarrow \mathbf{y} - \prod_{l=1}^i\mathbf{t}_l\mathbf{t}_l'\mathbf{y}$.

4. finally we have component matrix $U = [\mathbf{u_1}, \ldots, \mathbf{u_k}]$.

Step 4: Find the projections $\mathbf{V} = K^c U$ and $\mathbf{V}_{te} = K_{te}^c U$ for the training and test data respectively.

Step 5: Build a logistic regression model using $\mathbf{V}$ and $\{y_i\}_{i=1}^n$ and test the model performance using $\mathbf{V}_{te}$ and $\{y_t\}_{t=1}^{n_t}$. Let the estimated coefficients be $\boldsymbol{\beta} = (\beta_0, \ldots \beta_k)$, $\mathbf{v}_t$ be the k dimensional vector of sample t, and the logistic
function be g, we have $p(y = 1|\boldsymbol{\beta}, \mathbf{v}_t) = g(\beta_0 + \sum_{j=1}^k \beta_j \mathbf{v}_{tj})$

The dimension of projection (the number of components) k used in the model can be selected based on Akaike's information criteria (AIC):

$$AIC = -2 \log(\hat{L}) + 2(k + 1),$$

where $\hat{L}$ is the maximum likelihood. Let the estimated logistic regression
coefficients be $\boldsymbol{\beta} = (\beta_0, \ldots \beta_k)$, and $\mathbf{v}$ be the k PLS components of the training sample. Then the maximum likelihood $\hat{L}$ can be calculate as

$$\hat{L} = \prod_{j=1}^n \left(p(y = 1|\boldsymbol{\beta}, \mathbf{v}) \right)^{y_j} \left(1 - p(y = 1|\boldsymbol{\beta}, \mathbf{v}) \right)^{1-y_j}.$$

We choose k with minimum AIC value as our estimate of the number of PLS
components.

Acknowledgements

References

[1] Adam, B.L., Qui, Y., Davisi, J.W., Wardi, M.D., Clementsi, M.A., Cazares, L.H., et al. (2002) Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men. *Cancer Res*, 62, 3609-3614.

[2] Baggerly, K.A., Morris, J.S., and Coombes, K.R. (2004) Reproducibility of SELDI-TOF protein patterns in serum: comparing datasets from different experiments. *Bioinformatics* 20, 777-785.

[3] Baggerly, K.A., Morris, J.S., Edmonson, S.R., Coombes, K.R. (2005) Signal in Noise: evaluating Reported Reproducibility of Serum Proteomic Test for Ovarian Cancer. *Journal of National Cancer Institute*, 97, 307-309.

[4] Burges, C.J.C. (1998) A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2, 955-974.

[5] Colantuoni C, Henry G, Zeger S, Pevsner J. (2002). Local mean normalization of microarray element signal intensities across an array surface: quality control and correction of spatially systematic artifacts. *Biotechniques*. 32(6):1316-20.

[6] Farid, H., and Simoncellii, E.P. (2004) Differentiation of discrete multidimensional signals. *IEEE Transactions on Image Processing*, 13, NO. 4. pp, 486-508.

[7] Froning, J.N., Olson, M.D., Froelicher, V.F. (1988) Problems and limitations of ECG baseline estimation and removal using a cubic spline technique during exercise ECG testing: recommendations for proper implementation. *J Electrocardiol*,21 Suppl:S149-57.

[8] Lee, S., Kwon, D., Lee, S. (2003) Dimensionality reduction for indexing time series based on the minimum distance, *Journal of Information Science and Engineering*, 19, 697-711.

[9] Lilien, R.H., Farid, H., and Donald, B.R. (2003) Probabilistic Disease Classification of Expression-Dependent Proteomic Data from Mass Spectrometry of Human Serum. *Journal of Computational Biology*, 10, 925-946.

[10] Liu, Z., and Chen, D. (2004) Gene Expression Data Classification with Revised Kernel Partial Least Squares Algorithm. *FLAIRS Conference*, pp. 104-108.

[11] Liu, Chen, D. and Bensmail, H. Gene expression data classification with kernel principal component analysis, J. of Biomedicine and Biotechnology, 2005:2, 155-159.

[12] Petricoin, E.F. III, Ornstein, D.K., Paweletzi, C.P., Ardekani, A., Hacketti, P.S., Hitti, B.A., et al. (2002) Serum proteomic patterns for detection of prostate cancer. *J Natl Cancer Inst* 1576-1578.

[13] Qu, Y., Adam, B.L., Thornquist, M., Potter, J.D., Thompson, M.L., Yasui, Y., Davis, J., Schellhammer, P.F., Cazares, L.H., Clements, M.A., Wright, G.L. Jr., and Feng, Z. (2003) Data Reduction Using a Discrete Wavelet Transform in Discriminant Analysis of Very High Dimensionality Data. *Biometrics*, 59, 143-151(9).

[14] Roboz, J. (2002) Mass Spectrometry in Cancer Research. *CRC Press*

[15] Rosipal, R. and Trejo, L.J. (2001) Kernel partial least squares regression in RKHS, Theory and empirical comparison. *Technical report # 14*, University of Paisley, UK.

[16] Samuelsson, J., Dalevi, D., Levander, F., and Rgnvaldsson, T. (2004) Modular, scriptable and automated analysis tools for high-throughput peptide mass fingerprinting. *Bioinformatics*, 20, 3628-3635.

[17] Silapachote, P., Karuppiah, D., and Hanson, A. (2004) Feature Selection Using AD-ABOOST for Face Expression Recognition. *The 4th IASTED International Conference on Visualization, Imaging, and Image Processing,* Marbella, Spain, pp, 452, 273-279.

[18] Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K., and Zhao, H. (2003) Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics*, 19, 1636-1643

[19] Zhu J. and Hastie, T. Kernel logistic regression and the import vector machine, J. of Computational and Graphical Statistics, vol 14, no. 1, March, 2005, 185-205

[20] Zhu, W., Wang, X., Ma, Y., Rao, M., Glimm, J., and Kovach J.S. (2003) Detection of cancer-specific markers amid massive mass spectral data. *Proceedings of the National Academy of Sciences USA*, 100, 14666-14671.