partial least squares, classifications,
maximal separation criterion
microarray experiment, supervisor learning

Paweł BŁASZCZYK[*],
Katarzyna STĄPOR[**]

MAXIMUM SEPARATION PARTIAL LEAST SQUARES (MSPLS): A NEW METHOD FOR CLASSIFICATION IN MICROARRAY EXPERIMENT

The purpose of the paper is to propose a new method for classification. Our MSPLS method was deduced from the classic Partial Least Squares (PLS) algorithm. In this method we applied the Maximum Separation Criterion. On the basis of the approach we are able to find such weight vectors that the dispersion between the classes is maximal and the dispersion within the classes is minimal. In order to compare the performance of classifier we used the following types of dataset – biological and simulated. Error rates and confidence intervals were estimated by the jackknife method.

1. INTRODUCTION

Prediction, classification and clustering are the basic methods used to analyse and interpret microarray data in the microarray experiments. This kind of dataset contains vectors of genes expression, which belong to certain classes. Unfortunately, the number of vectors is usually much smaller than the number of genes. In this situation the classification results can be inadequate. That is the reason why it is so important to decrease the dimension of feature space, which could be done either by feature selection or by feature extraction. One of the feature extraction methods, which can be used, is the Partial Least Squares (PLS) Method.

In this paper we would like to propose a new method for feature extraction. Our method was deduced from the classic Partial Least Squares (PLS). This method was proposed by H. Wold (see [10], [11], [12]) and is often applied in chemometrics but it could be also applied for classification samples. PLS makes use of the ordinary least squares regression steps in the calculation of loadings, scores and regression coefficients. In the classic PLS method the objective criterion has the following formula: $w_k = \arg\max_{w^T w=1} \operatorname{cov}(Xw, Y)$. In this way we are able to find a weight vector for which the covariance between linear combination features of matrix X and the response matrix (vector in one dimensional case) Y is maximal. The covariance between the two elements denotes the degree of dependency between them. That is why when we find the weight

[*] University of Silesia, Institute of Mathematics, ul. Bankowa 14, 40-007 Katowice, pblaszcz@math.us.edu.pl
[**] The Silesian University of Technology, Institute of Computer Science, ul. Akademicka 12, 44-100 Gliwice, katarzyna.stapor@polsl.pl

vector for which covariance between the linear combination feature of matrix X and the response matrix Y is maximal, we can say that significant coefficients for each feature of matrix X have been found. It means that when we have sample matrix X, which contains n samples of p features, and response matrix Y, we know for which classes these samples belong and we can find, for every sample, the weight vector, which denotes significant coefficients. So that is the reason why the covariance between $\mathrm{X_i w_i}$ and Y_i must be maximal.

Unfortunately, this criterion does not provide suitable separation between classes, especially when dataset is not linearly separable and variables are high correlated. To provide better separation between classes, the objective criterion was modified. We used the maximal separation criterion. This criterion has a formula $tr(S_B - S_W)$, where S_B and S_W are between scatter matrix and within scatter matrix respectively. When maximizing this criterion matrix S_B is maximizing and matrix S_W is minimizing. It means that there is the distance between classes but the distance between samples in classes is minimizing. When applying this criterion to the classic PLS method, we obtain a new objective criterion.
This new criterion has formula

$$w_k = \arg\max \operatorname{cov}\ (X(S_B - S_W)^T w_k, Y) = \arg\max (N-1)^{-1} w_k^T (S_B - S_W) XY\ .$$

The weight vectors are computed in the same way as in the classic PLS method on the basis of matrix X but this matrix is transformed into a new space, which provides maximal separation between classes.

2. METHODS

2.1. CRITERION OF MAXIMUM SEPARATION

In this paper we consider a linear case but this method could be modified for nonlinear cases by using kernel function. Let us assume that we have L-classes problem and let $(x_i, y_i) \in X \times \{C_1, C_2, ..., C_n\}; x \in R^d$ where X is a matrix of sample vectors and $Y = \{C_1, C_2, ..., C_n\}$ is the vector of class labels. Firstly, however we should recall some basic facts. Let S_B and S_W denote a between scatter matrix and within scatter matrix respectively. It means that S_B and S_W are given by:

$$S_B = \sum_{i=1}^{L} p_i (M_i - M_0)(M_i - M_0)^T \qquad (1)$$

$$S_W = \sum_{i=1}^{L} p_i \Sigma_i \qquad (2)$$

where Σ_i denotes covariance matrix, p_i is a priori probability of appearance of i-th class, M_i means vector for i-th class and M_0 has the following formula:

$$M_0 = \sum_{i=1}^{L} p_i M_i \tag{3}$$

Let us define the following function:

$$y = \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} p_i p_j d(M_i,M_j) - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} p_i p_j (tr(\Sigma_i) + tr(\Sigma_j)) \tag{4}$$

where $d(M_i,M_j)$ denotes the distance between vectors M_i and M_j. It shows that when using the basic properties of trace of matrix and bearing in mind the fact that $S_W = \sum_{i=1}^{L} \Sigma_i$, the right side of formula (4) can be written in the following way:

$$\frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} p_i p_j (tr(\Sigma_i) + tr(\Sigma_j)) = tr(S_W). \tag{5}$$

If we assume that $d(M_i,M_j)$ denotes Euclidean distance, it means that $d(M_i,M_j) = (M_i - M_0)(M_i - M_0)^T$, it shows that the left side part of formula (4) can be written by:

$$\frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} p_i p_j d(M_i,M_j) = tr(S_B). \tag{6}$$

Our formula (4) can be written in the following way:

$$y = tr(S_B - S_W). \tag{7}$$

This function is called the criterion of maximum separation. When transposing our vector X through our function $y = W^T X$ to vector Y, we can say that the projection of S_B and S_W has the following formula:

$$S_W^Y = W^T S_W W, \tag{8}$$

$$S_B^Y = W^T S_B W. \tag{9}$$

We get the following criterion function:

$$J(W) = tr(W^T (S_B - S_W) W). \tag{10}$$

It can be shown that the maximized function $\max \sum_{k=1}^{L} W_k^T (S_B - S_W) W_k$, under condition $W_k^T W_k = 1$, when using the Lagrange multiplier method and applying basic properties of the eigenvalues criterion function (10), can be written in the following way:

$$J(W) = \sum_{k=1}^{d} \lambda_k \tag{11}$$

where λ_k are eigenvalues of the matrix $(S_B - S_W)$ and d is a number of eigenvalues of this matrix.

2.2. NEW ALGORITHM

By using the maximal separation criterion we were able to design a new algorithm. The algorithm is based on classical PLS algorithm (see [10], [11], [12], [9]). It is used for two class problem only, but it can be easily generalized for multiclass problem. In our algorithm, as well as in the classic one, the objective criterion for constructing components is sequentially maximizing the covariance between the response variable and the linear combination of the predictors. Thus, we found the weight vector W satisfying the following objective criterion:

$$w_k = \arg\max \operatorname{cov}\ (X(S_B - S_W)^T w_k, Y) = \arg\max (N-1)^{-1} w_k^T (S_B - S_W) XY \tag{12}$$

subject to

$$\begin{aligned} & w_k^T w_k = 1; \quad for\ 1 \le k \le k \\ & t_k^t t_k = 0 \end{aligned} \tag{13}$$

Our MSPLS algorithm can be summarized in the following way:

1. FOR $k = 1$ to d set u to Y and DO:
2. $w = \dfrac{(S_B - S_W)X^T u}{u^T u}$ and scale w to be unit length
3. $t = Xw$
4. $c = \dfrac{Y^T}{t^T t}$ and scale c to be unit length
5. $u = Yc$
6. $p = \dfrac{X^T t}{t^T t}$
7. $b = \dfrac{u^T t}{t^T t}$
8. Residual matrices: $X_{(k+1)} = X_k - tp^T$ and $Y_{(k+1)} = Y_k - btc^T$
9. END FOR

Next, we calculate regression coefficients by the following formula:

$$Q = w \cdot \left[\frac{p^T w}{c^T} \right]^{-1} \tag{14}$$

These coefficients are used to classification. Because of the fact that, in this paper, we are using this method for two class problem, the decision function has the following formula:

$$testY = \mathrm{sgn}(testX \cdot Q - 1) \tag{15}$$

where $testX$ is matrix of test vectors and $testY$ is the vector of predicted class labels.

2.3. ESTIMATING ERROR RATES

In order to estimate error rates for each dataset, we use the Jackknife resampling, which is the special case of the bootstrap procedure [4]. We used the Jackknife method as follows: we randomly selected a single sample of the test dataset, learned the classifier on the remaining samples and classified the chosen sample obtaining the error rate. The error rate was equal either to 0, if the sample was classified correctly or to 1, if it was not. This procedure has been repeated for 1000 times. Each time, we selected the sample from a full set of samples. The bootstrap (jackknife) error rate ER_J was the mean error rate from each step. The error rate was as follows:

$$ER_J = \frac{1}{n} \cdot \sum_{i=1}^{n} er(i) \tag{16}$$

where n was the number of repetitions and $er(i)$ was the error rate in i-th iteration. In this paper, we used n being equal to 1000. We used percentiles of the ER_J error distribution in order to find the end points of the confidence intervals. We used 1000 ER_J values estimated in the way given above to estimate the distribution. For the significant level α the confidence interval was bounded at $\alpha/2$ and $1-\alpha/2$ percentiles, so the confidence interval was defined by:

$$CI = \left(p_{\alpha/2}; p_{1-\alpha/2}\right) \tag{17}$$

where $p_{\alpha/2}$ was $\alpha/2$ percentile.

3. EXPERIMENTAL RESULTS

3.1. DATASETS

The method described in the previous section was compared with the standard PLS algorithm on several simulated DNA microarray datasets. Firstly, we prepared datasets consisting of two groups of 15 arrays with 2000 genes. Two datasets IS01, IS05 with 1% (20) and 5% (100) differentially expressed genes were generated according to the article by Broberg [2], using normal distributions with parameters given in Table 1. Only the last three rows represented different expression.

Table 1. Means and standard deviations used in IS01, IS05

Group 1		Group 2	
μ_1	σ_1	μ_2	σ_2
-8	0,2	-8	0,2
-10	0,4	-10	0,4
-12	1	-12	1
-6	0,1	-6,1	0,1
-8	0,1	-8,5	0,2
-10	0,4	-11	0,7

We assumed an equal probability of every model from the first three rows for non-differentially expressed genes and from the second three rows of the table for DEGs. We also generated another datasets including two groups of arrays with 21 arrays belonging to the first group and 19 arrays to the second one. Each array includes 2000 genes; the proportion of DEGs was set equal to 1% that is we had 20 differentially expressed genes in these datasets. Firstly, we independently generated each entry of the 2000×40 matrix from the standard normal distribution. Secondly, we added a value of 2 to the first 100 genes in the first group to model differentially expressed genes. Thus, first 100 genes in the first group were normally distributed with mean 2 and all the elements of the whole matrix were stochastically independent. Afterwards, we independently generated 40 random numbers $a_1,...,a_{40}$ from the standard normal distribution. Then, for the fixed correlation value ρ we applied the following transformation for each entry of the generated matrix: $x_{ij} := \sqrt{\rho} a_j + \sqrt{1-\rho} x_{ij}$, where $i = 1,...,2000$ was the number of gene and $j = 1,...,40$ was the number of sample, so that for any $i_1 \neq i_2$ and j we had $Corr\left(x_{i_1 j}, x_{i_2 j}\right) = \rho$. Using the procedure described above, we generated training and test datasets called CS02 and CS06, with the chosen correlation strength at the level of 0.2 and 0.6 respectively.

We also compared all of these methods on the leukemia dataset, available at [16]. The dataset came from a study of gene expression in two types of acute leukemias: acute lybphoblastic leukemia (ALL) and acute myeloid leukemia (AML). The training set contained 38 cases (27 ALL and 11 AML) and test dataset contained 34 cases (20 ALL and 14 AML) both with 7129 genes.

3.2. RESULTS

For the comparison of the classification performance on simulated dataset we applied exactly the same scheme in each case. Firstly, the dataset was standardized. Next, we built the classifier and estimated the error rates by using the jackknife method. The procedure varied for the leukemia and the colon datasets. We calculated raw p-values from the t-test, then we used 100000 permutations in order to estimate them. Secondly, we controlled FDR at level α equal to *0,01*. So, for leukemia and colon dataset we chose respectively 861 and 180 genes for which p-values was less than 0,01. We reduced the original dataset to this selected genes. Afterwards, the experiment schemes were exactly the same as in the case of the simulation dataset.

3.2.1. BIOLOGICAL AND SIMULATED DATA

The results for the biological data were very satisfactory. For the leukemia dataset our classifier did not make any mistake. These results were better than in the case of the classic PLS algorithm, where mean error rate equalled 0,027 and the confidence interval was [0,026, 0,028]. The details are presented in table 2.

Table 2. Classification results for the leukemia dataset

No. sample	Class	MSPLS	PLS	Sample no.	Class	MSPLS	PLS
1	0	0	0	18	0	0	0
2	0	0	0	19	0	0	0
3	0	0	0	20	0	0	0
4	0	0	0	21	1	1	1
5	0	0	0	22	1	1	1
6	0	0	0	23	1	1	1
7	0	0	0	24	1	1	1
8	0	0	0	25	1	1	1
9	0	0	0	26	1	1	1
10	0	0	0	27	1	1	1
11	0	0	0	28	1	1	1
12	0	0	0	29	1	1	1
13	0	0	0	30	1	1	1
14	0	0	0	31	1	1	0
15	0	0	0	32	1	1	1
16	0	0	0	33	1	1	1
17	0	0	0	34	1	1	1

As far as the colon dataset is concerned, the results were also very good. The mean error rate was 0,048 and the confidence interval was [0.048-0.049]. These results were better than in the case of the classic PLS, where the mean error rate was 0,104 and the confidence interval equalled [0,096-0,104]. The details are presented in table 3.

Table 3. Classification results for the colon dataset

No. sample	Class	MSPLS	PLS	Sample no.	Class	MSPLS	PLS
1	0	0	0	17	0	0	0
2	0	0	0	18	0	0	0
3	0	0	0	19	0	0	0
4	0	0	0	20	0	0	0
5	0	0	0	21	1	1	1
6	0	0	0	22	1	1	1
7	0	0	0	23	1	1	1
8	0	0	0	24	1	1	0
9	0	0	0	25	1	1	1
10	0	0	0	26	1	1	1
11	0	0	0	27	1	1	0
12	0	0	0	28	1	1	1
13	0	0	1	29	1	1	1
14	0	0	0	30	1	1	1
15	0	0	0	31	1	1	1
16	0	0	0				

We compared the performance of classification for four simulated datasets. For the simulated dataset IS01 and IS05 we received very good results. For each number of components, every sample was classified properly. For the next two simulated datasets CS02 and CS06 the results were not as good as in the case of the IS datasets. In the CS06 case, the mean error rate was 0.075 in the confidence interval [0,074; 0,076]. For CS02 dataset the results were even worse. The mean error rate equaled 0.15 in the confidence interval [0.14; 0.16]. When comparing these results with the ones which we received after exactly the same procedure for the classic PLS algorithm, it came out that the results were far better for the new algorithm. The error rates were smaller about 5%. The confidence interval was also much smaller. For the PLS algorithm, the length of the confidence interval was three times bigger than for the MSPLS algorithm.

4. CONCLUSION

We have introduced statistical analysis method for the classification in microarray experiment. The method was able to distinguish between normal and tumor samples for two different biological microarray data and other four different simulated microarray data. Our methods used a Maximum Separation Criterion to find the weights vector so that the dispersion between the classes would be maximal, whereas the dispersion within the class-minimum. The Proposed method was not restricted to any specific microarray technology. For the computable reasons we made pre-selection by choosing genes with the smallest p-values. The p-values were estimated by using raw p-values method (see [3]). By comparing our method with the classic partial Least Squares Algorithm, it was possible to show that the classification performance estimated by the Jackknife procedure was significantly higher. The restriction of only one dimensional class vector in the dataset was the main disadvantage. In the future, we would like to propose a multivariate algorithm and a new kernel based MSPLS algorithm.

BIBLIOGRAPHY

[1] BASTIEN P., VINZI V. E., TENENHAUS M., PLS generalized linear regression, Computational Statistics \& Data Analysis, 48 (2005), pg. 17-46.
[2] BROBERG P.: Statistical methods for ranking differentially expressed genes. Genome Biology 2003, 4:R41.
[3] DUDOIT S., YANG Y. H., CALLOW M. J., SPEED T .P.: Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. Technical Report 578, Department of Statistics, UC Berkeley, CA, 2000.
[4] EFRON B.: Estimating the error rate of prediction rule improvement on cross-validation. Journal of the American Statistical Association, 1983, Vol. 78, No. 382.
[5] FUKUNAGA K., Introduction to statistical pattern recognition, Academic Press Professional, New York, 1990.
[6] GARTHWAITE P. H., An interpretation of Partial Least Squares, Journal of the American Statistical Association, Mar 1994, 89, 425, ABI/INFORM Global, pg. 122.
[7] HÖSKULDSSON A., PLS Regression methods, Journal of Chemometrics, vol. 2, 211-228 (1988).
[8] HÖSKULDSSON A., Variable and subset selection in PLS regression, Chemometrics and Intelligent laboratory Systems, 55 (2001), 23-38.
[9] NGUYEN D. V., ROCKE D. M., On Partial Least Squares dimension reduction for a microarray-based classification: a simulation study, Computational Statistic and Data Analysis, 46 (2004) pg. 407-425.

[10] WOLD H., Soft Modeling by Latent Variables: The Non-Linear Iterative Partial Least Squares (NIPALS) Approach, Perspectives in Probability and Statistics. Papers in Honour of M. S. Bartlett, London 1975, pg. 117-142.
[11] WOLD S., MARTENS H., WOLD H, The multivariate calibration problem in chemistry solved by the PLS method, Proc. Conf. Matrix Pencils, (A. Ruhe and B. Kagström, eds.), March 1982, Lecture Notes in Mathematics, Springer Verlag, Heidelberg, pg. 286-293.
[12] WOLD S., RUHE A., WOLD H., The co-linearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses, SIAM J. Sci. Stat. Comput. Vol. 5, no 3, September 1984.
[13] WOLD S., SJÖSTRÖM M., ERIKSSON L., PLS-regression: a basic tool of chemometrics, Chemometrics and Intelligent laboratory Systems, 58, 2001, pg. 109-130.
[14] VAPNIK V. N., Statistical Learning Theory, Wiley, New York 1998.
[15] VAPNIK V. N., The nature of statistical learning theory 2ed., Springer, 2000.
[16] http://www.broad.mit.edu