Semi-supervised Feature Extraction Method Using Partial Least Squares and Gaussian Mixture Model

Pawel Blaszczyk, *Member, IAENG*

***Abstract*—The aim of this paper is to present a new semi-supervised classification method based on modified Partial Least Squares algorithm and Gaussian Mixture Models. The economical datasets are used to compare the performance of the classification.**

***Index Terms*—Partial Least Square, Gaussian Mixture Model, Semi-Supervised Learning, Classification, Feature Extraction, Kernel Methods.**

I. Introduction

FEATURE extraction, classification, and clustering are the basic methods used to analyze and interpret multivariate data. For classification task, the datasets contain vectors of features belonging to certain classes. These vectors are called samples. On the other hand, for the purpose of clustering, we do not have information about proper classification of objects. In datasets for classification tasks, the number of samples is usually much smaller compared to the number of features. In this situation, the small number of samples makes it impossible to estimate the classifier parameters properly; therefore, the classification results may be inadequate. In the literature, this phenomenon is known as the Curse of Dimensionality. In this case, it is important to decrease the dimension of the feature space. This can be done either by feature selection or feature extraction. Some of the linear feature extraction methods are for example Principal Component Analysis (PCA) and Partial Least Squares (PLS). These methods are often applied in chemometrics, engineering, computer vision, and many other applied sciences. However, the classical approach to feature extraction is based on the mean and the sample covariance matrix. It means that these methods are sensitive to outliers. Moreover, when the features and the target variables are non-linearly related, linear methods cannot properly describe the data distribution. Different non-linear versions of PCA and PLS have been developed (see [13], [9], [14]). In real classification task, we often have the dataset with relatively small amount of labeled data and a huge amount of data without labels. In real applications, we frequently encounter the problem with obtaining labeled data, as it is both time-consuming and capital-intensive. Sometimes, it requires specialized equipment or expert knowledge. Labeled data is very often associated with intense human labor, as in most applications, each of the examples need to be marked manually. In such situations, semi-supervised learning can have a great practical value. The semi-supervised techniques allow us to use both labeled and unlabeled data. Including the information coming from unlabeled data and semi-supervised learning, we can improve the feature extraction task. Unlabeled data, when used in conjunction with a small amount of labeled data, can improve learning accuracy.

In this paper, we present a new semi-supervised method for nonlinear feature extraction. We propose to combine a kernel for modified Partial Least Squares method with a Gaussian Mixture Model (GMM) (see [2], [15]) clustering algorithm. The supervised kernel exploits the information conveyed by the labeled samples and the cluster kernel from the structure of the data manifold. The proposed semi-supervised method was successfully tested in economical datasets.

Manuscript received July 10, 2015; revised July 31, 2015.

Pawel Blaszczyk is with the Institute of Mathematics, University of Silesia, Bankowa 14, Katowice, 40-007 Poland (phone: +48-32-258-29-76; fax: +48-32-258-29-76; e-mail: pawel.blaszczyk@us.edu.pl).

II. Methodology

Let us assume that we have the L-classes classification problem and let $(x_i, y_i) \in X \times \{C_1, \ldots, C_L\}, x \in \mathbb{R}^p$ where matrix of sample vectors X and response matrix Y are given by the following formulas:

$$X = \begin{bmatrix} x_{11} & \ldots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \ldots & x_{np} \end{bmatrix}, \quad Y = \begin{bmatrix} 1 & 0 & \ldots & \ldots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \ldots & 0 & 1 \end{bmatrix}. \tag{1}$$

Each row of the matrix Y contain 1 in a position denoting the class label.

A. Partial Least Squares

One of the commonly used feature extraction methods is the Partial Least Squares (PLS) Method (see [16], [4], [8]). PLS uses of the least squares regression method [7] in the calculation of loadings, scores and regression coefficients. The idea behind the classic PLS is to optimize the following objective function:

$$(w_k, q_k) = \arg\max_{w^T w=1; q^T q=1} \operatorname{cov}(X_{k-1} w, Y_{k-1} q) \tag{2}$$

under conditions:

$$w_k^T w_k = q_k q_k^T = 1 \quad \text{for } 1 \leq k \leq d, \tag{3}$$

$$t_k^T t_j = w_k^T X_{k-1}^T X_{j-1} w_j = 0 \text{ for } k \neq j, \tag{4}$$

where $\operatorname{cov}(X_{k-1} w, Y_{k-1} q)$ is a covariance matrix between $X_{k-1} w$ and $Y_{k-1} q$, vector t_k is the k-th extracted component, w_k is the vector of weights for k-th component and d denotes the number of extracted components. The matrices X_k, Y_k arise from X_{k-1}, Y_{k-1} by using so called deflation technique which removes the k-th component using the following formulas:

$$X_{(k+1)} = X_k - t_k t_k^T X_k \tag{5}$$

$$Y_{(k+1)} = Y_k - t_k t_k^T Y_k \tag{6}$$

ISBN: 978-988-14047-2-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Extracted vector w_k corresponds to the eigenvector connected with the largest eigenvalue of the following eigenproblem:

$$X_{k-1}^T Y_{k-1} Y_{k-1}^T X_{k-1} w_k = \lambda w_k \tag{7}$$

Let S_B denote the between scatter matrix and S_W within scatter matrix, given by:

$$S_B = \sum_{i=1}^{L} p_i \left(M_i - M_0\right)\left(M_i - M_0\right)^T, \tag{8}$$

$$S_W = \sum_{i=1}^{L} p_i E\left[\left(X - M_i\right)\left(X - M_i\right)^T | C_i\right] = \sum_{i=1}^{L} p_i S_i, \tag{9}$$

where S_i denotes the covariance matrix, p_i is a priori probability of the appearance of the i-th class, M_i is the mean vector for the i-th class, and M_0 is given by:

$$M_0 = \sum_{i=1}^{L} p_i M_i. \tag{10}$$

These matrices are often used to define separation criteria for evaluating and optimizing the separation between classes. For the PLS, a separation criterion is used to find vectors of weights that provide an optimal separation between classes in the projected space. In PLS method the matrix in each k-th step is:

$$X_k^T Y_k Y_k^T X_k = \sum_{i=1}^{L} n_i^2 \left(M_i - M_0\right)\left(M_i - M_0\right)^T \tag{11}$$

This matrix is almost identical to the between class scatter matrix S_B. Hence, we can say that the separation criterion in the PLS method is based only on the between scatter matrix. It means that the classic PLS method is that it does not properly separates the classes. To provide a better separation between classes we can use weighted separation criterion (see [1]) denoted by:

$$J = \operatorname{tr}\left(\gamma S_B - (1-\gamma) S_W\right). \tag{12}$$

where γ is a parameter from interval $[0, 1]$, S_B and S_W are between scatter matrix and within scatter matrix, respectively. Applying a linear transformation criterion, condition (12) can be rewritten in the following form:

$$J\left(w\right) = \operatorname{tr}\left(w^T \left(\gamma S_B - (1-\gamma) S_W\right) w\right). \tag{13}$$

which is more suitable for optimization. The next step is to optimize the following criterion:

$$\max_{w_k} \sum_{k=1}^{d} w_k^T \left(\gamma S_B - (1-\gamma) S_W\right) w_k, \tag{14}$$

under the conditions:

$$w_k^T w_k = 1 \quad \text{for } 1 \le k \le p. \tag{15}$$

The solution to this problem can be found using the Lagrange multipliers method. To find the correct value of the parameter γ, we used the following metric:

$$\rho(C_1, C_2) = \min_{c_1 \in C_1, c_2 \in C_2} \rho(c_1, c_2), \tag{16}$$

where C_i is the i-th class for $i \in \{1, 2\}$. The value of the parameter γ was chosen by the using the following formula:

$$\gamma = \frac{\min_{i,j=1,\ldots,L, i \neq j} \left\{\rho(C_i, C_j)\right\}}{1 + \min_{i,j=1,\ldots,L, i \neq j} \left\{\rho(C_i, C_j)\right\}}. \tag{17}$$

Parameter γ equals 0 if and only if certain i and j classes exist for which $\rho(C_i, C_j) = 0$. This means that at least one sample which belongs to classes C_i and C_j, exist. If distance between classes increase, the value of γ also increases. Therefore the importance of the component S_W becomes greater.

To improve separation between classes in classic PLS method, we replace the matrix (11) with the matrix from our separation criterion (13) (see [1]) to optimize the objective criterion

$$w_k = \arg\max_w \left(w^T (\gamma S_B - (1-\gamma) S_W) w\right), \tag{18}$$

under the following conditions:

$$w_k^T w_k = 1 \quad \text{for } 1 \le k \le d \tag{19}$$

$$t_k^T t_j = w_k^T X_{k-1}^T X_{j-1} w_j = 0 \quad \text{for } k \neq j, \tag{20}$$

We call this extraction algorithm, i.e., Extraction by applying Weighted Criterion of Difference Scatter Matrices (EWCDSM). One can prove that the extracted vector w_k corresponds to the eigenvector connected with the largest eigenvalue for the following eigenproblem:

$$(\gamma S_B - (1-\gamma) S_W) w = \lambda w. \tag{21}$$

Additionally, the k-th component corresponds to the eigenvector related to the largest eigenvector in the following eigenproblem:

$$X_{k-1} X_{k-1}^T \left(D - (1-\gamma) I\right) t = \lambda t. \tag{22}$$

Matrix $D = [D_j]$ is an $n \times n$ block-diagonal matrix where D_j is a matrix in which all elements equals $1/nn_j$, and n_j is the number of samples in the j-th class.

A proper features extraction for nonlinear separable is difficult and could be inaccurate. Hence, for this problem we designed a nonlinear version of our extraction algorithm. We used the following nonlinear function $\Phi : x_i \in \mathbb{R}^N \to \Phi(x_i) \in F$ which transforms the input vector into a new, higher dimensional feature space F. Our aim is to find an EWCDSM component in F. In F, vectors w_k and t_k are given by the following formulas:

$$w_k = (D - (1-\gamma) I) \mathbf{K}_k w_k \tag{23}$$

$$t_k = \mathbf{K}_k w_k \tag{24}$$

where $\mathbf{K}$ is the kernel matrix. One can prove that the extracted vector w_k corresponds to the eigenvector connected with the largest eigenvalue using the following formula:

$$\left(D_k - (1-\gamma) I\right) \mathbf{\Phi}_k \mathbf{\Phi}_k^T w_k = \lambda w_k. \tag{25}$$

Furthermore, the k-th component corresponds to the eigenvector connected with largest eigenvector in the following eigenproblem:

$$\mathbf{K}_{k-1} (D_{k-1} - (1-\gamma) I) t = \lambda t. \tag{26}$$

ISBN: 978-988-14047-2-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

B. Classification using PLS method

Let us assume that X_{train} and X_{test} are the realizations of the matrix X for train and test datasets respectively. The idea of a training step is to extract vectors of weights w_k and components t_k by using the train matrix X_{train} and to store them as a columns in matrices W and T respectively. To classify samples into classes, we use train matrix X_{train} to compute the regression coefficients by using the least squares method [7] given by:

$$Q = W\left(P^T W\right)^{-1} U^T, \tag{27}$$

where,

$$U = YY^T T\left(T^T T\right)^{-1}, \tag{28}$$

$$W = X^T U, \tag{29}$$

$$P = X^T T\left(T^T T\right)^{-1}. \tag{30}$$

We then multiply test matrix X_{test} by the coefficients of the matrix Q. To classify samples corresponding to the Y_{test} matrix, we use the decision rule:

$$y_i = \arg\max_{j=1,\ldots,L} Y_{test}(i,j). \tag{31}$$

The final form of the response matrix is the following:

$$Y_{test} = \left[\ y_1 y_2 \cdots y_L\ \right]^T. \tag{32}$$

Like for the linear version of the algorithm, if want make a prediction, first we must compute the regression coefficient using the formula (33)

$$Q = \mathbf{\Phi}^T U(T^T \mathbf{K} U)^{-1} T^T Y, \tag{33}$$

where T is matrix of the components and matrix U has the following form

$$U = YY^T C. \tag{34}$$

We make a prediction by multiplying the test matrix data $\mathbf{\Phi}_{test}$ by matrix Q, i.e.

$$\hat{Y} = \mathbf{\Phi}_{test} Q, \tag{35}$$

and then by using the decision rule

$$y_i = \arg\max_{j=1,\ldots,L} \hat{Y}(i,j). \tag{36}$$

Finally, the response matrix has the following form

$$Y_{test} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_L \end{bmatrix} \tag{37}$$

Like in classic kernel PLS algorithm, if we want make a prediction for the data from test dataset, we use the following formula.

$$\hat{Y} = \mathbf{K} U\left(T^T \mathbf{K} U\right)^{-1} T^T Y = TT^T Y, \tag{38}$$

and the decision rule has the following formula

$$y_i = \arg\max_{j=1,\ldots,L} [TT^T Y](i,j). \tag{39}$$

Finally, the response matrix is given by

$$Y_{test} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_L \end{bmatrix}. \tag{40}$$

III. Gaussian Mixture Model

Gaussian mixture model (GMM) (see [2], [15]) is a kind of mixture density model, which assumes that each component of the probabilistic model is a Gaussian density component, i.e., given by formula

$$p(x|\theta_k) = \frac{1}{\sqrt{2\pi}^p |\Sigma_k|} exp - \frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k) \tag{41}$$

where $\theta_k = (\mu_k, \Sigma_k)$ are the parameters of the Gaussian distribution, including the mean μ_k and positive defined covariance matrix Σ_k. Hence, the Gaussian Mixture Model is the probability density on $\mathbb{R}^p$ given by formula

$$p(x|\theta) = \sum_{k=1}^{M} p(x|\theta_k) p(k) \tag{42}$$

where $\theta = (\mu_1, \Sigma_1, \mu_2, \Sigma_2, \ldots, \mu_M, \Sigma_M,)$ is the vector of the model parameters, $p(k)$ represents a priori probabilities, which sum to one. In GMM method, we assume that the covariance matrices are diagonal; hence, the GMM is specified by $(2p+1)M$ parameters. The parameters are learned form the training dataset by classical Expectation-Maximization (EM) algorithm (see [12]). With Gaussian components, we have two steps in one iteration of the EM algorithm. E-step is the first step in which we re-estimate the expectation based on the previous iteration

$$p(k|x) = \frac{p(k)p(x|k)}{\sum_{i=1}^{M} p(k)p(x|k)} \tag{43}$$

$$p(k)^{new} = \frac{1}{n}\sum_{i=1}^{n} P(k|x). \tag{44}$$

The second step is so-called M-step in which we update the model parameters to maximize the log-likelihood

$$\mu_i = \frac{\sum_{j=1}^{n} p(k|x_j) x_j}{\sum_{j=1}^{n} p(k|x_j)} \tag{45}$$

$$\Sigma_i = \frac{\sum_{j=1}^{n} p(k|x_j)(x_j-\mu_i)(x_j-\mu_i)^T}{\sum_{j=1}^{n} p(k|x_j)}. \tag{46}$$

The initial values of μ_i are randomly chosen from a normal distribution with the mean $\mu_0 = \frac{1}{n}\sum_{i=1}^{n} x_i$ and the covariance $\Sigma_0 = \frac{1}{n}\sum_{i=1}^{n}(xi\mu_0)(xi\mu_0)^T$. Using the Bayes rule, it is possible to obtain *a posteriori* probability $\pi_{i,k}$ of x belonging to cluster k by the following formula

$$\pi_{i,k} = \frac{p(x|k)p(k)}{p(x)} \tag{47}$$

where $p(x|k)$ is the conditional probability of x given the cluster k. It means that GMM is a linear combination of Gaussian density functions. The GMM clustering is fast and provides posterior membership probabilities.

IV. Proposed Semi-supervised modified PLS method

Like in [6], in this paper, we propose using a Gaussian Mixture Model (GMM) to perform the clustering, which is fast and provides posterior probabilities that typically lead to smoother kernels (see [6], [5]). The proposed cluster kernel will be the combination of a kernel computed from labeled data and a kernel computed from clustering unlabeled data (using GMM), resulting in following algorithm:

ISBN: 978-988-14047-2-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

1) Compute the kernel for labeled data using the following formula

$$\mathbf{K}_s(x_i, x_j) = \Phi(x_i)^T \Phi(x_j) \tag{48}$$

2) Run the GMM algorithm n times with different initial values and number of clusters This results in $q \cdot t$ cluster assignments where each sample has its corresponding posterior probability vector $\pi_i \in \mathbb{R}^m$, where m is the number of clusters.
3) Compute the kernel for all (labeled and unlabeled) data. The kernel is the mean of inner products maximum posterior probabilities π_i and π_j. The kernel is given by following formula:

$$\mathbf{K}_u(x_i, x_j) = \frac{1}{N} \sum_{k=1}^{t} \sum_{l=1}^{q} \pi_i^T \pi_j \tag{49}$$

where m is the number of clusters, N is normalization factor.
4) Compute the final kernel using the following formula

$$\mathbf{K}(x_i, x_j) = \delta \mathbf{K}_s(x_i, x_j) + (1 - \delta)\mathbf{K}_u(x_i, x_j) \tag{50}$$

where $delta \in [0, 1]$ is a scalar parameter tuned during validation.
5) Use the computed kernel into kernel PLS method.

Because the kernel in (49) corresponds to a summation of inner products in $t \cdot q$-dimensional spaces, the above kernel in (49) is a valid kernel. Additionally the summation of (50 leads also to valid Mercers kernels.

V. Experiments

A. Dataset

We applied the new extraction method to commonly accessible economical datasets: *Australian Credit Approval* and *German Credit Data*.We compared our method with PLS based on the Australian Credit Approval available at [17]. The Australian Credit Approval was introduced in papers [10], [11]. This dataset contains information from credit card application form divided into two classes denoted as 0 and 1. Class 1 contains information about people who receive positive decision regarding credit card application. Class 0 contains information about people who receive negative decision regarding credit card application. This dataset contains 690 samples, where 307 samples are those taken from class 0. The remaining 383 samples belong to class 1. Each sample is represented by 14 features. The second dataset, German Credit Data available at [17] contained 1000 samples divided into two classes: class 0 and class 1. Each sample is represented by 30 features.Both datasets contained some non-numerical features. In order to apply extraction algorithm to those datasets, the data had to be relabeled. We assigned natural numbers as new values of non-numerical features.

B. Experimental scheme and Results

To examine the classification performance of proposed method, we used the following experimental scheme. First, we normalized each dataset. For each dataset, we randomly chose 10% of samples as a labeled data (5% from each class). To define the $(q \cdot t)$ cluster centers and the posterior probabilities for each of them, we used 200 samples as unlabeled samples per each class in both datasets. In all cases, we tuned the parameter δ from 0 to 1 in 0.05 intervals. When the mixture models were computed, we chose the most probable Gaussian mode and computed the K_c kernel. We used the nonlinear version of EWCDSM with the Gaussian kernel and parameter σ. The result for all datasets are presented in the Table I. We used the jackknife method [3] to find the proper value of parameters δ and σ. Classification performance is computed by dividing the number of samples classified properly by the total number of samples. This rate is know as a standard error rate [3].

TABLE I
Classification performance (per cent) of economic datasets

	Australian	German
SS Kernel EWCDSM	95,65	94,21
PLS	63,91	83,78

VI. Conclusions

We introduced a new kernel version of an algorithm for semi-supervised feature extraction. Our algorithm uses weighted separation criterion to find the weights vector, which allows for the scatter between the classes to be maximal and for the scatter within the class to be minimal. When comparing the new criterion with the other well known ones, it can be seen that the new one can be used in a situation where the number of samples is small and the costs of computation are lowered. The new extraction algorithm can distinguish between between high-risk and low-risk samples for two different economical datasets. Moreover, we have shown that our method had significantly higher classification performance compared to classical the PLS method. The presented method performs well in solving classification problems. However, to draw some more general conclusions, further experiments should be conducted using other datasets.

References

[1] P. Blaszczyk and K. Stapor *A new feature extraction method based on the Partial Least Squares algorithm and its applications*, Advances in Intelligent and Soft Computing, 179-186, 2009.
[2] K. Chatfield, V.S. Lempitsky, A. Vedaldi and A. Zisserman *The devil is in the details: an evaluation of recent feature encoding methods*. In BMVC Vol. 2, No. 4, p. 8, SPringer 2011.
[3] R. Duda and P. Hart *Pattern Classification*. John Wiley & Sons. New York 2000.
[4] P. H. Garthwaite *An interpretation of Partial Least Squares. In: Journal of the American Statistical Association*, 89:122, 1994
[5] L. Gomez-Chova, G. Camps-Valls, L. Bruzzone and J. Calpe-Maravilla *Mean map kernel methods for semisupervised cloud classification*, IEEE Trans. Geosci. Rem. Sens., vol. 48, no. 1, pp. 207220, 2010.
[6] E. Izquierdo-Verdiguier, L. Gomez-Chova, L. Bruzzone and G. Camps-Valls *Semisupervised nonlinear feature extraction for image classification*, IEEE Workshop on Machine Learning for Signal Processing, MLSP12.
[7] J. Gren *Mathematical Statistic*, PWN, Warsaw, 1987, in polish.
[8] A. Höskuldsson *A PLS Regression methods*, Journal of Chemometrics, 2:211-228, 1988.
[9] M. A. Kramer *Nonlinear principal component analysis using autoassociative neural networks*, AIChE Journal, vol. 37, no. 2, pp. 233243, 1991.
[10] J. R. Quinlan *Simplifying decision trees*. Int. J. Man-Machine Studies, vol. 27, pp.221-234, 1987.
[11] J. R. Quinlan *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

ISBN: 978-988-14047-2-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[12] S.J. Roberts, *Parametric and non-parametric unsupervised cluster analysis*, Pattern Recognition 30(2), 261-272, 1997.
[13] S. T. Roweis and L. K. Saul *Nonlinear dimensionality reduction by locally linear embedding*, Science, vol. 290, no. 5500, pp. 2323–2326, December 2000.
[14] J. Shawe-Taylor and N. Cristianini *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.
[15] J. Wang, J. Lee and C. Zhang *Kernel GMM and its and its application to image binarization*, Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on. Vol. 1. IEEE, 2003.
[16] H. Wold *Soft Modeling by Latent Variables: The Non-Linear Iterative Partial Least Squares (NIPALS) Approach. In: Perspectives in Probability and Statistics. Papers in Honour of M. S. Bartlett*, 117-142, 1975.
[17] UC Irvine Machine Learning Repository. Available: http://archive.ics.uci.edu/ml/.

ISBN: 978-988-14047-2-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)