Revista Colombiana de Estadística - Applied Statistics
Enero 2020, volumen 43, no. 2, pp. 233 a 249
http://dx.doi.org/10.15446/rce.v43n2.81811

PLS Generalized Linear Regression and Kernel Multilogit Algorithm (KMA) for Microarray Data Classification Problem

Regresión lineal generalizada por MCP y algoritmo kernel multilogit para la clasificación de datos de microarreglos

Adolphus Wagala[1,a], Graciela González-Farías[1,b], Rogelio Ramos[1,c], Oscar Dalmau[2,d]

[1]Department of Probability and Statistics, Centro de Investigación en Matemáticas A.C., Guanajuato, México

[2]Department of Computer Science, Centro de Investigación en Matemáticas A.C., Guanajuato, México

Abstract

This study involves the implentation of the extensions of the partial least squares generalized linear regression (PLSGLR) by combining it with logistic regression and linear discriminant analysis, to get a partial least squares generalized linear regression-logistic regression model (PLSGLR-log), and a partial least squares generalized linear regression-linear discriminant analysis model (PLSGLRDA). A comparative study of the obtained classifiers with the classical methodologies like the k-nearest neighbours (KNN), linear discriminant analysis (LDA), partial least squares discriminant analysis (PLSDA), ridge partial least squares (RPLS), and support vector machines(SVM) is then carried out. Furthermore, a new methodology known as kernel multilogit algorithm (KMA) is also implemented and its performance compared with those of the other classifiers. The KMA emerged as the best classifier based on the lowest classification error rates compared to the others when applied to the types of data are considered; the unpreprocessed and preprocessed.

***Key words*:** Generalized linear regression; Kernel multilogit algorithm; Partial least squares.

[a]PhD. E-mail: adolphus.wagala@cimat.mx
[b]PhD. E-mail: farias@cimat.mx
[c]PhD. E-mail: rramosq@cimat.mx
[d]PhD. E-mail: dalmau@cimat.mx

Resumen

Este estudio combina el modelo de regresión lineal generalizado por mínimos cuadrado parciales (RLGMCP), con regresión logística y análisis discriminante lineal, para obtener los modelos de regresión logística generalizada por mínimos cuadrados parciales, (RLGMCP) y regresión logística generalizada-discriminante por mínimos cuadrados parciales (RLGDMCP). Se realiza un estudio comparativo con clasificadores clásicos como, k-vecinos más cercanos (KVC), análisis discriminante lineal (ADL), análisis discriminante de por mínimos cuadrados parciales (ADMCP), regresión por mínimos cuadrados parciales (RMCP) y máquinas de vectores de soporte de soporte vectorial (MSV). Además, se implementa una nueva metodología conocida como algoritmo de kernel multilogit (AKM). Su desempeño es comparado con los de los otros clasificadores. De acuerdo con las tasas de error de clasificación obtenidas a partir de los diferentes tipos de datos, el KMA es el de mejor resultado.

***Palabras clave*:** Regresíon lineal generalizada; Algoritmo de kernel multilogit; Mínimos cuadrados parciales.

1. Introduction

The field of genomics has witnessed a tremendous increase in the amount of data generation due to biotechnological advances like microarrays and next-generation sequencing platforms. These biotechnological advances have made it possible to simultaneously monitor expression levels for thousands of genes, and thus help in solving particular problems related to the identification of molecular variants in them, and their relation to the classification, diagnosis, prognosis and treatment of different conditions. The high dimensional data generated from microarray technology involve many thousands of genes measured simultaneously, a different microarray for each individual. This definitely introduces some noise and unwanted variations that might stem from technical or unknown sources.

In a microarray experiment let n and p be the numbers of the samples and genes respectively, so that the generated data is a $n \times p$ matrix. The main challenge with these technologies is that the resultant data are noisy due to biological and technological variations, and at the same time they usually are high dimensional, i.e., they have more variables than cases due to a low sample size, so $n << p$. This condition makes the direct application of most classical statistical methodology implausible, leading researchers to propose new solutions for this type of problem.

Normally before the down stream analysis of the data generated from DNA microarrays, a preprocessing and normalization stage is performed to remove the noise, filtering out the genes with low expression values, addressing missing values, and standardizing the data via a log-transformation. One of the most used preprocessing procedures for microarray data was proposed by Dudoit, Fridlyand & Speed (2002), which entails three basic steps, namely: thresholding, filtering out of genes outside of a range of minimum/maximum intensities, and finally, standardization of the expression values by a log transformation (Alshamlan, Badr & Alohali 2013, Dudoit et al. 2002).

This work considers classification problems for microarray data sets under two conditions: un-preprocessed and preprocessed. In the un-preprocessed data all genes available in the study are included, while in the preprocessed only the subset of genes believed to play important roles in the biological problem of interest are used. We extend the Partial Least Squares Generalized Linear Regression (PLSGLR) algorithm of Bastien, Vinzi & Tenenhaus (2005) by combining it with Logistic Regression, to give PLSGLR-log, and with Linear Discriminant Analysis to come up with PLSGLRDA. Furthermore, we compare their performance with that of the kernel multilogit algorithm (KMA) proposed by Dalmau, Alarcón & González (2015), and of the classical methods: the k-Nearest Neighbour (KNN), Ridge Partial Least Squares (RPLS), Partial Least Squares-Linear Discriminant Analysis (PLSDA), the usual Linear Discriminant Analysis (LDA) and the Support Vector Machines (SVM), when applied to a set of microarray data, referred to in this work as the Colon data set by Alon, Barkai, Notterman, Gish, Ybarra, Mack & Levine (1999). We evaluate the classifiers with regard to their classification error rates in this data set and compare them.

Our work addresses problems similar to many studies involving classification in microarrays, with typically high dimensional data and low numbers of samples (or subjects). Following a two stage strategy, many involve the use of the original PLS to build the components, even though the response variables are discrete, for example the analysis of Nguyen & Rocke (2002*a*, 2002*b*); this is intuitively not correct since the original PLS is an algorithm best suited for continuous response variables (that is, variables with numeric values that have an infinite number of values between any two values). And in almost all of the procedures a variable (gene) selection step is implemented, with an accompanying computing cost. This paper describes a procedure suitable for categorical data, and its performance is studied with and without the gene selection step, and compared to that of each of the other classifiers used. An additional advantage of our approach is that the PLSGLR can deal with missing values, unlike the original PLS, commonly used in the literature.

The proposed two stage strategy for the classification problem is described as follows.

To the best of our knowledge, the proposed combination of PLS generalized linear regression algorithm with logistic and discriminant analysis has not been used before in cases where $n << p$. The PLS generalized linear regression algorithm is simple, and a good performance when compred to the classical methods would make it an attractive alternative (See Table 1).

2. Kernel Multilogit Algorithm (KMA)

The KMA was recently proposed by Dalmau et al. (2015). This algorithm works by first transforming a categorical response variable to a continuous one via a multilogit transformation. A categorical variable in this case refers to the one that contain a finite number of categories or distinct groups for instance tumor and non-tumor. The transformed variable is then used with the explanatory variables

in a regression model for classification and prediction. Finally, the new predicted variables are transformed back using the inverse multilogit function to the original space to enable classification.

Table 1: Proposed strategy

Steps
Step 1: Dimension reduction In this stage, we propose to use PLSGLR to project the high dimensional data to a low dimension space thus resulting in new components (latent variables), which preserve the information in the intrinsic structure of the data.
Step 2: Use of latent variables for classification Analyze the obtained latent variables with the classical statistical classifiers: i PLSGLR components with logistics regression to get the PLSGLR-logistic model denoted as (PLSGLR-log) ii PLSGLR components with linear discriminant analysis model to get PLSGLR-Linear Discriminant Analysis model denoted as (PLSGLRDA)

Let the response variable vector $\boldsymbol{y}$ be categorical with class labels $\{1, 2, \ldots, C\}$. To classify a discrete variable from predictor variables $\boldsymbol{x}$, the first step is to transform the response variable $\boldsymbol{y}$ into a new space using the multilogit function. The multinomial logit model with C as the reference category can be given as

$$\begin{aligned} \Pr(\boldsymbol{y} = j \mid \boldsymbol{x}) &= \frac{\exp\{f(\boldsymbol{x}; \boldsymbol{\theta}_j)\}}{1 + \sum_{i=1}^{C-1} \exp\{f(\boldsymbol{x}; \boldsymbol{\theta}_i)\}}, \quad j = \{1, 2, \ldots, C-1\} \\ \Pr(\boldsymbol{y} = C \mid \boldsymbol{x}) &= \frac{1}{1 + \sum_{i=1}^{C-1} \exp\{f(\boldsymbol{x}; \boldsymbol{\theta}_i)\}}, \end{aligned} \tag{1}$$

where $f(\boldsymbol{x}; \boldsymbol{\theta}_i) = \boldsymbol{x}^T \boldsymbol{\theta}_i$. The expected value of $\boldsymbol{y}$ being a multinomial random variable is given by $E(\boldsymbol{y} \mid \boldsymbol{x}) = [\Pr(\boldsymbol{y} = 1 \mid \boldsymbol{x}), \Pr(\boldsymbol{y} = 2 \mid \boldsymbol{x}), \ldots, \Pr(\boldsymbol{y} = C \mid \boldsymbol{x})]^T$. Now, denoting $\boldsymbol{t} = E(\boldsymbol{y} \mid \boldsymbol{x})$, the original response variable $\boldsymbol{y}$ is not used but instead a transformed version $\boldsymbol{\vartheta} = \text{logit}(\boldsymbol{t})$ is used. The logit transformation is done with C as the reference category as follows

$$\vartheta_j = \text{logit}(t_j) = \log \frac{t_j}{t_C}, \quad j = \{1, 2, \ldots, C-1\} \tag{2}$$

where $\vartheta_j \in \boldsymbol{\vartheta}, t_j \in \boldsymbol{t}$.

In the second step a parametric linear model is proposed and its parameter estimates can be obtained via the standard Bayesian formula $\Pr(\boldsymbol{\vartheta} \mid \boldsymbol{x}) = \Pr(\boldsymbol{x} \mid \boldsymbol{\vartheta})\Pr(\boldsymbol{\vartheta})/\Pr(\boldsymbol{x})$ where $\Pr(\boldsymbol{\vartheta} \mid \boldsymbol{x})$ is the posterior probability distribution, $\Pr(\boldsymbol{x} \mid \boldsymbol{\vartheta})$ is the likelihood function and $\Pr(\boldsymbol{x})$ is the normalization constant, assuming that $\boldsymbol{\vartheta} \in \mathbb{R}^{C-1}$ for a given $\boldsymbol{x} \in \mathbb{R}^m$ follows a multivariate normal distribution $\boldsymbol{\vartheta} \mid \boldsymbol{x} \sim \mathcal{N}(\Theta^T \mathbf{x}, \alpha^2 \mathbf{I})$, $\Theta \in \mathbb{R}^{m \times C-1}$,$\Pr(\boldsymbol{\vartheta} \mid \boldsymbol{x})$ is also multivariate normally distributed. Furthermore, the prior parameters are assumed to follow a normal distribution, i.e. $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \beta^2 \mathbf{I})$ where β is known. The parameter matrix $\boldsymbol{\Theta}$ is

thus estimated by optimizing an equivalent of a regularized least squares function

$$\begin{aligned} \widehat{\boldsymbol{\Theta}} &= \arg\min_{\Theta} \mathbf{U}(\boldsymbol{\Theta}) \\ U(\boldsymbol{\Theta}) &= \|\boldsymbol{\vartheta} - \mathbf{X}\boldsymbol{\Theta}\|_F^2 + \lambda\|\boldsymbol{\Theta}\|_F^2, \end{aligned} \tag{3}$$

where $\boldsymbol{\vartheta} = [\boldsymbol{\vartheta}^{(i)}]_{i=1,2,\ldots,n}^T$, $\mathbf{X} = [\boldsymbol{x}^{(i)}]_{i=1,2,\ldots,n}^T$, $\|.\|_F$ is the Frobenius norm of a matrix and λ is the regularization parameter. The result is a closed form estimate given by $\widehat{\boldsymbol{\Theta}} = (\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\boldsymbol{\vartheta}$. To capture non-linearities which may be present, a dual representation $\boldsymbol{\Theta} = \mathbf{X}^T\boldsymbol{\Gamma}$ is taken so that

$$U(\boldsymbol{\Gamma}) = \|\boldsymbol{\vartheta} - \mathbf{X}\mathbf{X}^T\Gamma\|_F^2 + \lambda\|\mathbf{X}^T\boldsymbol{\Gamma}\|_F^2$$

then $U(\boldsymbol{\Gamma})$ is optimized to get $\widehat{\boldsymbol{\Gamma}} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\boldsymbol{\vartheta}$, where $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ is the Gram matrix, $K_{ij} = \langle \boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)} \rangle + 1$. However a more general kernel $K_{ij} = ((\phi(\boldsymbol{x}^{(i)}), \phi(\boldsymbol{x}^{(j)}))$ where $\phi(\cdot)$ is a nonlinear mapping, is preferred in practice.

The final step of the algorithm involves prediction/classification given a new set of response variables $\boldsymbol{x}^{new}$. This entails estimation of $\boldsymbol{\vartheta}^{new}$ by $\boldsymbol{\vartheta}^{new} = \widehat{\boldsymbol{\Gamma}}^T\widehat{\boldsymbol{x}}^{new}$, but $\widehat{\boldsymbol{x}}^{new} = K((\phi(\boldsymbol{x}^{(i)}), \phi(\boldsymbol{x}^{(new)}))$. The computed $\boldsymbol{\vartheta}^{new}$ is used to estimate $\mathbf{t}^{new}$ by using $\boldsymbol{t}^{new} = \text{logit}^{-1}(\boldsymbol{\vartheta}^{new})$. The inverse of a logit function is given by

$$\begin{aligned} t_j^{new} &= \frac{\exp\{\vartheta_j^{new}\}}{1 + \sum_{i=1}^{C-1} \exp\{\vartheta_j^{new}\}}, \quad j = \{1, 2, \ldots, C-1\} \\ t_J^{new} &= \frac{1}{1 + \sum_{i=1}^{C-1} \exp\{\vartheta_j^{new}\}}. \end{aligned} \tag{4}$$

The class labels associated with $\boldsymbol{x}^{new}$ are then computed using the estimated conditional distribution by finding the components that maximize those of $\boldsymbol{t}^{new}$ i.e. using the Bayes rule. The computed $\mathbf{t}^{new}$ is then used to get the class label ($\widehat{\boldsymbol{y}}^{new}$) of the new data; for details see (Dalmau et al. 2015).

3. Partial Least Squares (PLS) and Some of its Applications in Genomics

PLS is a very useful approach because it is able to analyze data with many, noisy, collinear as well as incomplete variables. PLS is usually utilized in data reduction when there is multicollinearity or when the data have more variables than the number of samples. Essentially, the PLS aims at maximizing the covariance between the response variables $\boldsymbol{Y}$ and the predictors $\boldsymbol{X}$, i.e., $cov(\boldsymbol{X}^T\boldsymbol{Y})$ of highly multidimensional data by finding a linear subspace of the explanatory variables (Wold, Sjöström & Erikson 2001, Höskuldsson 1988). Some literature on PLS can be found in (Wold et al. 2001, Wold, Ruhe, Wold & Dunn III 1984, Höskuldsson 1988), among others.

The research on PLS is still very active due to its ability to address problems associated with the high dimensional data such as multicollinearity and high dimensionality, among others. In the recent past, PLS has been utilized predominantly in high dimensional data in different fields like chemometrics and the "omics" like genomics, proteomics, metabolomics, and many other fields that generate large amounts of data, like spectroscopy (Gromski, Muhamadali, Ellis, Xu, Correa, Turner & Goodcare 2015). Recent applications of PLS in microarray studies include Huang, Tu, Huang, Lien, Lai & Chuang (2013), who applied PLS regression (PLSR) in breast cancer intrinsic taxonomy, for classification of distinct molecular sub-types by using PAM50 signature genes as predictive variables in PLS analysis and the latent binary gene component analyzed by a logistic regression for each molecular sub-type. Also, Telaar, Liland, Repsilber & Nürnberg (2013) extended the notion of PLS-discriminant analysis (PLS-DA) to Powered PLS-DA (PPLS-DA), introducing a 'power parameter' maximised towards the correlation between the components and the group-membership, thereby achieving a minimal classification error. Furthermore, Xi, Gu, Baniasadi & Raftery (2014) discussed the PLS-DA with applications to metabolites data. Other articles involving the usage of PLS include: Dong, Zhang, Zhu, Wang & Wang (2014) who used PLS to investigate the underlying mechanism of the post-traumatic stress disorder (PTSD) using microarray data; Gusnanto, Ploner, Shuweihdi & Pawitan (2013), who made gene selection based on partial least squares and logistic regression random-effects (RE) in classification models; gene selection involving PLS was also done by Wang, An, Chen, Li & Alterovitz (2015). The sparse PLS has also been utilized by many researchers; for instance, Chun & Keles (2009),Lee, Lee, Lee & Pawitan (2011) and Chung & Keles (2010) provided an efficient algorithm for the implementation of sparse PLS for variable selection in high dimensional data. Furthermore, Lê Cao, Rossouw, Robert-Granieé & Besse (2008) used sparse PLS for variable selection when integrating omics data. They implemented sparsity via lasso penalization of the PLS loading vectors when computing the singular value decomposition.

4. PLS Generalized Linear Regression Algorithm

In this section, we present an algorithm that can be applied to any Generalized Linear Regression which was developed by (Bastien et al. 2005). Consider the response data $\mathbf{y}$ with the explanatory variables $\mathbf{x_1}, \ldots, \mathbf{x_p}$; then a PLS-General Linear Regression (PLSGLR) can be written as

$$g(\theta) = \sum_{h=1}^{m} c_h \Big(\sum_{j=1}^{p} w_{hj}^{*} \mathbf{x}_j \Big), \tag{5}$$

where θ a conditional expectation of the variable $\mathbf{y}$ if its distribution is continuous, or a vector of probabilities if the variable $\mathbf{y}$ follows a discrete distribution with a finite support, while $g(.)$ is the link function chosen according to the probability distribution of $\mathbf{y}$. The PLS components are given by $t_h = \sum_{j=1}^{p} w_{hj}^{*} \mathbf{x}_j, j =$

$1, \ldots, p, h = 1, \ldots, m$. To compute the PLS components, let $\mathbf{X} = \mathbf{x_1} \ldots, \mathbf{x_p}$ be a matrix of p centred explanatory variables $\mathbf{x}_j$'s. The key objective is to determine m orthogonal PLS components defined as a linear combination of the $\mathbf{x}_j$'s. The algorithm is presented as follows:

1. Computation of the first PLS component t_1: First, the regression coefficients a_{1j} of $\mathbf{x}_j$ are computed using the usual Generalized Linear Model (GLM) procedure of $\mathbf{y}$ on $\mathbf{x}_j, j = 1 \ldots p$. The column vector $\mathbf{a}_1$ which contains a_{1j} is then normalized: $\mathbf{w}_1 = \mathbf{a}_1/\|\mathbf{a}_1\|$. Finally, the component t_1 is computed as $\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1/\mathbf{w}_1'\mathbf{w}_1$.

2. Computation of the second PLS component t_2: Involves the computation of the linear model coefficients a_{2j} of $\mathbf{x}_j$ in the GLM setting of $\mathbf{y}$ on t_1 and $\mathbf{x}_j, j = 1, \ldots, p$. Since the main idea of PLS is to create the orthogonal components t_2, the component t_1 is added as a variable in estimating $\mathbf{y}$ on t_1 and $\mathbf{x}_j, j = 1, \ldots, p$. This is because the structure of PLSGLR does not allow the residuals of y to be obtained in each iteration that would aid in construction of orthogonal components. The column vector $\mathbf{a}_2$ which contains a_{2j} is normalized: $\mathbf{w}_2 = \mathbf{a}_2/\|\mathbf{a}_2\|$ and thereafter, the residual matrix $\mathbf{X}_1$ is obtained via the regression of $\mathbf{X}$ on t_1. The use of residual matrix in the attainment of the next component ensures orthogonality between the different components. The component t_2 is calculated by $\mathbf{t}_2 = \mathbf{X}_1\mathbf{w}_2/\mathbf{w}_2'\mathbf{w}_2$. Finally, $\mathbf{t}_2$ is expressed in pterms of $\mathbf{X}$: $\mathbf{t}_2 = \mathbf{X}\mathbf{w}_2^*$.

3. Computation of the h^{th} PLS Component t_h: Consider the already computed components $t_1, \ldots, t_{h-1}$; the final component t_h is computed by calculating the GLM coefficients a_{hj} of $\mathbf{x}_j$ by fitting $\mathbf{y}$ on $t_1, \ldots, t_{h-1}$ and $\mathbf{x}_j, j = 1, \ldots, p$. Next, the column vector $\mathbf{a}_h$, which contains a_{hj} is normalized as: $\mathbf{w}_h = \mathbf{a}_h/\|\mathbf{a}_h\|$. The residual matrix $\mathbf{X}_{h-1}$ of the regression of $\mathbf{X}$ on $t_1, \ldots, t_{h-1}$ is then computed. The use of the residual matrix and the previously obtained $t_1, \ldots, t_{h-1}$ as covariables in calculating the GLM coefficients helps the creation of orthogonal components, as previously explained. The final component t_h is thus computed as $\mathbf{t}_h = \mathbf{X}_{h-1}\mathbf{w}_h/\mathbf{w}_h'\mathbf{w}_h$. Finally, $\mathbf{t}_h$ is expressed in terms of $\mathbf{X}$: $\mathbf{t}_h = \mathbf{X}\mathbf{w}_h^*$.

While computing the components t_h, the nonsignificant elements in a_h can be set to zero in order to simplify calculations, since only the significant response variables are needed to build the PLS components. The number of m components to be used can be determined through cross-validation or by hard thresholding. The iteration can be stopped once there are no more significant coefficients in a_h (Bastien et al. 2005).

Consider $x_{h-1,i}$, a column vector of the transpose of the ith row of X_{h-1}; then $t_{hi} = x_{h-1,i}'w_h/w_h'w_h$ of the ith case on the component t_h. This is basically the slope of the fitted line of the univariate OLS linear regression without intercept for $x_{h-1,i}$ on w_h, which can be estimated even with some data missing. Consequently, the component is computed based on the available data. Therefore the PLSGLR algorithm by (Bastien et al. 2005) effectively copes up with missing data.

5. Error Rate as a Measure of Classification Accuracy

In literature, there exist numerous metrics for the classification accuracy for instance the error rate, sensitivity and specificity among others. Following the methodology of Boulesteix, Strobl, Augustin & Daumer (2008), consider a random vector $\boldsymbol{X} \in \mathbb{R}^p$ and the response vector $\boldsymbol{y} \in \{0, \ldots, k-1\}$. If $\boldsymbol{f}$ denotes the joint distribution of $\boldsymbol{X}$ and $\boldsymbol{y}$ then a classifier is a function say C that maps from $\mathbb{R}^p$ to $\{0, \ldots, k-1\}$ thereby assigning classes to a vector of some matrix $\boldsymbol{X}$ corresponding to the p-dimensional gene expression vector while $\widehat{\boldsymbol{y}}$ is the predicted class.

$$\begin{aligned} C : \mathbb{R}^p &\to \{0, \ldots, k-1\} \\ \boldsymbol{X} &\to \widehat{\boldsymbol{y}} \end{aligned} \tag{6}$$

For $\boldsymbol{f(X, y)}$ and $\boldsymbol{y}$ are known then the Bayes classifier can be constructed by

$$C_{bayes}(\boldsymbol{X}) = \arg\max_k P(\boldsymbol{Y} = k/\boldsymbol{X}) \tag{7}$$

from the derivation of the posterior distribution $P(\boldsymbol{y/X})$. The error rate is thus given by the distribution

$$\begin{aligned} Err(C) &= P_{\boldsymbol{f}}(C(\boldsymbol{X}) \neq \boldsymbol{y}) \\ &= E_{\boldsymbol{f}}(IC((\boldsymbol{X}) \neq \boldsymbol{y})) \end{aligned} \tag{8}$$

Now suppose that we have data for n sample observations (say patients) denoted by $\boldsymbol{D} = \boldsymbol{d}_1, \ldots, \boldsymbol{d}_n$ which are identically independently distributed observations $\boldsymbol{d}_i = (y_i, \boldsymbol{x}_i)$. Here $y_i \in \{0, \ldots, k-1\}$ denotes the membership of the response vector while $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^T$ the p-vector of the expression data for the i^{th} patient. Given the learning data set is given by $\boldsymbol{D}_1 = (\mathbf{d}_1^*, \ldots, \mathbf{d}_L^*)$ where L is the number of observations chosen for the learning set. If the classifier defined by equation 6 and learnt using the data $\boldsymbol{D}_1$ using the classification method M is denoted by $C^M_{\boldsymbol{D}_1}$ then the error rate given $\boldsymbol{D}_1$ is given by

$$C^M_{\boldsymbol{D}_1} = E_{\boldsymbol{f}}(I(\boldsymbol{y} \neq C^M_{\boldsymbol{D}_1}(\boldsymbol{X}))/\boldsymbol{D}_1) \tag{9}$$

which is unknown because the joing distribution $\boldsymbol{f}$ is unknown. However, for $\boldsymbol{D}_t = (\mathbf{d}_1^*, \ldots, \mathbf{d}_T^*)$ the estimator for error can be obtained as

$$\widehat{Err}(C^M_{\boldsymbol{D}_1}, \boldsymbol{D}_t) = \frac{1}{T} \sum_{i=1}^{T} I(y_{ti} \neq C^M_{\boldsymbol{D}_1}(x_{ti})) \tag{10}$$

where $\boldsymbol{X}_{ti} = (x_{t_i 1}, \ldots, x_{t_i p})$ is the p-vector giving the t^{th} observation's gene expression. The sensitivity and specificity are a consequence of the estimated error rate given by equation 9, for more details see Boulesteix et al. (2008). In this paper we therefore compare the different classifiers based on the error rates because it is deemed adequate in identifying the best classifier. Furthermore, the sensitivity and specificity of a classifier are not fixed characteristics but are influenced by the type of misclassification scheme utilized.

6. Applications to a Real Data Set

We describe in detail the analysis of the Colon data by Alon et al. (1999), obtained from the `R` package `plsgenomics`, which consists of a (62×2000) matrix giving the expression levels of 2000 genes for 62 colon tissue samples.

An exploratory analysis of the data was done in order to visualize the differences in the un-preprocessed and preprocessed microarray data sets. The preprocessing is done using the `R` package `plsgenomics` see `https://rdrr.io/cran/plsgenomics/`, that implements the recommendations of Dudoit et al. (2002). To visualize the differences between the preprocessed and un-preprocessed data sets, we consider the pairs of box plots, relative log expression (RLE), and principal components analysis (PCA) plots presented in Figures 1,2, 3, 4 and 5 respectively.

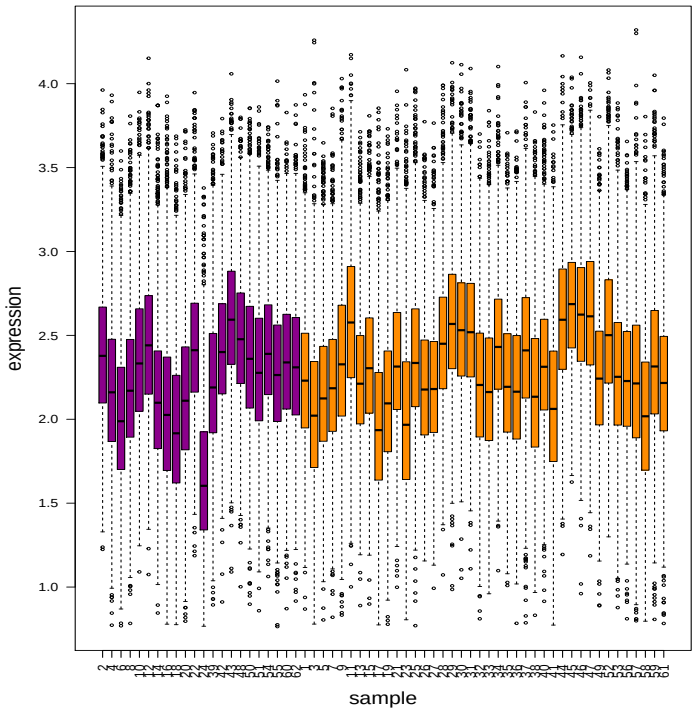


FIGURE 1: Box plot for the un-preprocessed colon data. The box plot for un-preprocessed data clearly shows that the data are noisy and have a lot of variations. The data have some unwanted variations that are expected to affect the analysis. They also lack symmetry.

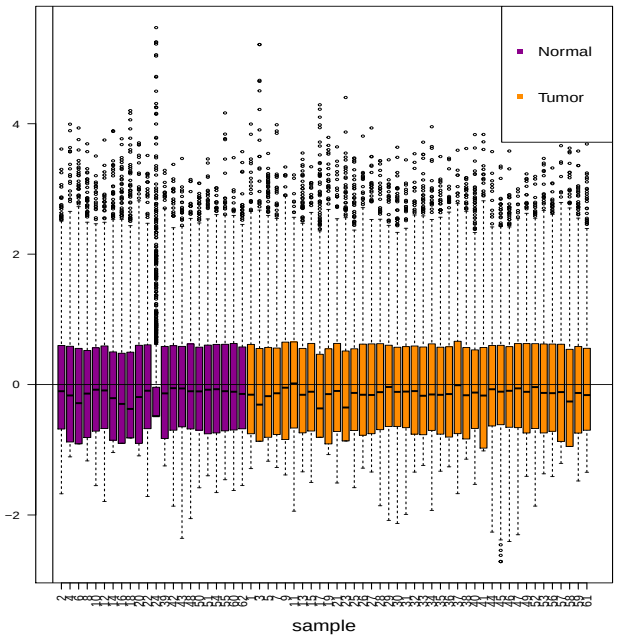
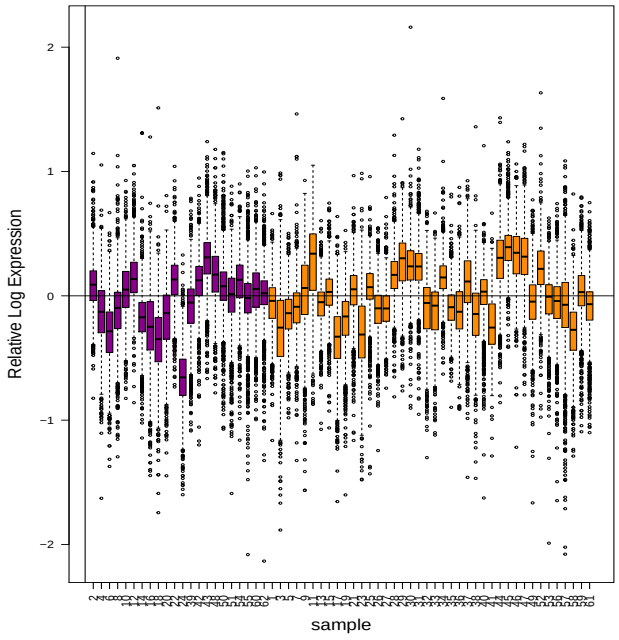


FIGURE 2: Box plot for the preprocessed colon data. This plot presents less variations. The data seem to have a symmetric distribution and do not show the presence of unwanted variation. From the two figures, it is expected that the preprocessed data would be easier to analyze.



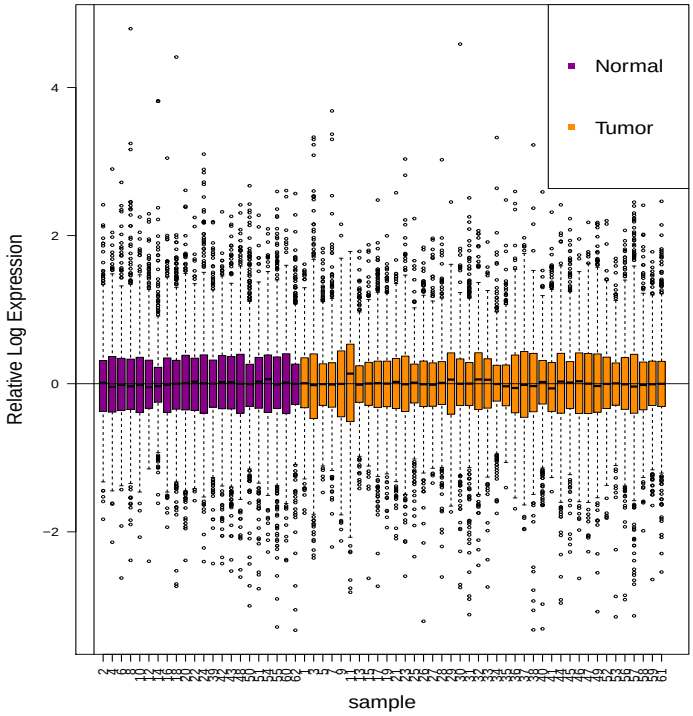


FIGURE 3: RLE plots for the un-preprocessed and preprocessed colon data. The RLE plot for the un-preprocessed data shows the presence of a lot of heterogeneity, implying that the data have variations that do not necessarily come from biological factors. However, the RLE plot for the processed data shows homogeneity and lack of unwanted noise, and should give better results when analyzed statistically.

The same pair of data sets is examined using RLE plots, to show how the preprocessed data compares with the un-preprocessed data set with regard to the batch effect or any other abnormality. The RLE plots have been extensively used in studies of microarray data to reveal the effectiveness of data normalization; for an example see Gagnon-Bartsch & Speed (2011). The RLE plots are simple yet very powerful in the visualization of data to detect unwanted variations. To understand how an RLE plot is constructed, consider a data matrix $\boldsymbol{X}_{p\times n}$ where p is the number of genes while n the number of microarray samples, and so the element of the data matrix x_{ij} represents the i^{th} gene in the j^{th}sample. The RLE plot is then constructed by first calculating the median across each of the p rows, and then substracting the respective median across each row of the data matrix $\boldsymbol{X}$, i.e $(x_{ij} - \text{median}x_{i*})$. The median is used because it is robust and not affected by outliers. A box plot is then generated for each of the n samples, and a good one will be centered around zero and its width (interquartile range) should be equal to or less than 0.2 see Gagnon-Bartsch & Speed (2011).

Finally we compare the ease of classification between the un-preprocessed and preprocessed data. The simplest way to visualize the separability of categories in a given data set is the use of principal components analysis (PCA) plots.

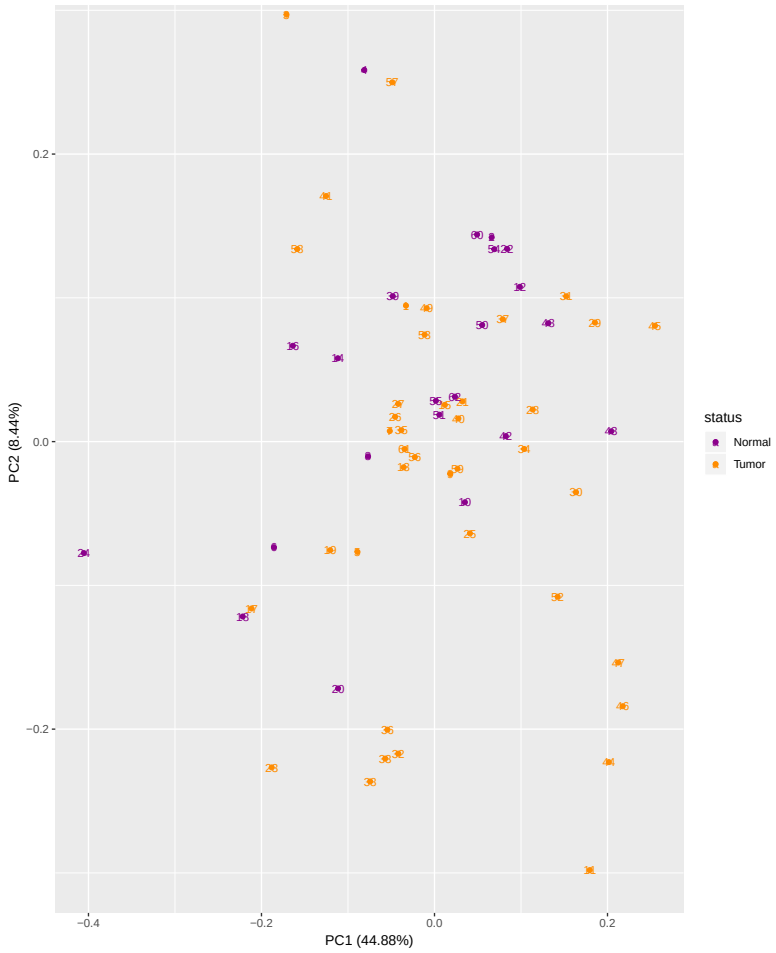


FIGURE 4: PCA plot for the un-preprocessed Colon data. The PCA plots show that it is harder to separate/classify the un-preprocessed data.

According to Gagnon-Bartsch & Speed (2011), one of the key challenges of the removal of unwanted variation is the difficulty in distinguishing the unwanted variations from the biological variation of interest. Furthermore, they note that the most appropriate way to deal with unwanted variation depends on the final objective of the analysis, for instance: differential expression (DE), classification, or clustering.

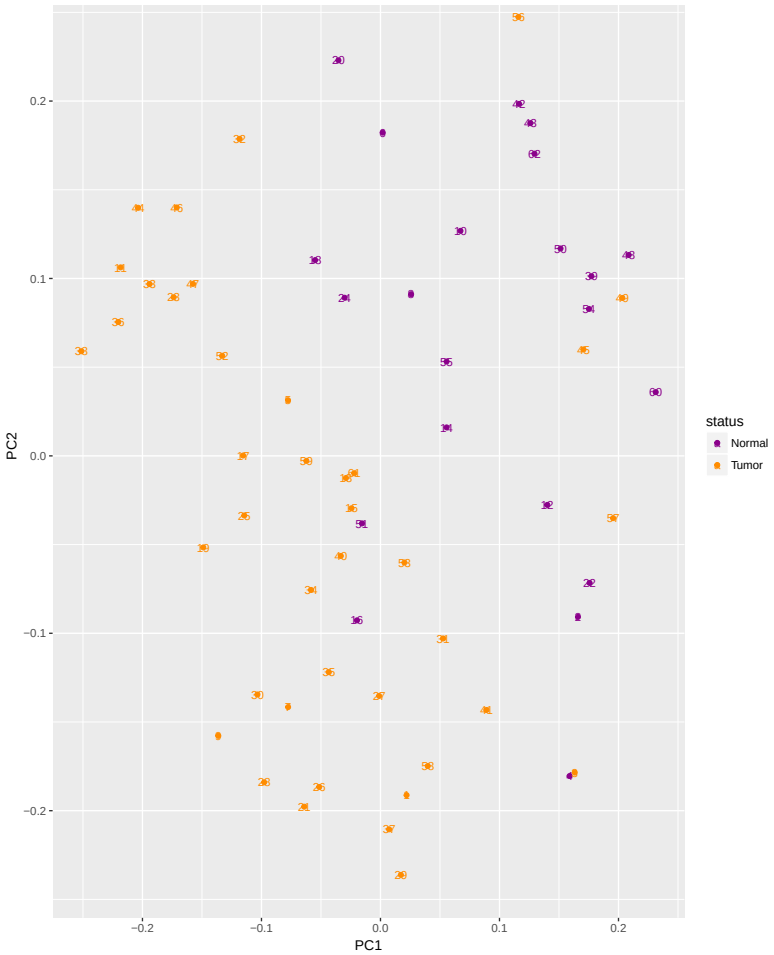


FIGURE 5: PCA plots for the preprocessed Colon data.It is relatively easier to separate/classify preprocessed data.

6.1. Analysis of the Un-Preprocessed Data

In this analysis, we compare the performance of our proposed model extensions PLSGLR-log, PLSGLRDA and the KMA (Dalmau et al. 2015) to that of the classical methods when the data has neither been preprocessed nor variables been selected, thus testing the performance of the classification algorithms in the presence of noise. The performance of the methodologies is then compared using a 10 fold cross validation (10-CV) and the corresponding classification error rates are computed. The results are presented in Table 2.

We note that there exist several metrics for measuring the performance of various classifiers. A particular method is judged to be the "best" if it has a lower classification error rate relative to the other methods, otherwise it is a poor classifier.

Table 2: Rate of classification error for the different methods when applied to the unpreprocessed data set

DATA	PLSGLR-log	PLSGLRDA	KNN	LDA	PLSDA	RPLS	SVM	KMA
Colon	38.3	31.7	60.0	25.0	11.7	15.0	18.3	1.7

The results based on minimal cross validation classification error rates indicate that for the Colon data, the KMA emerges as the best, followed by PLSDA, and RPLS, while the worst were KNN and PLSGLR-log.

6.2. Analysis of Preprocessed Data

During the preprocessing of microarray data the feature selection step is usually performed. This is because out of the thousands of variables (genes expression levels) generated, only a handful may play an important role towards the biological problem of interest. The thousands of data points are likely to be noisy due to biological or technical reasons. Thus the feature selection extracts a subset of the genes that are most informative (optimum subset of features). This reduces the noise by removing irrelevant or redundant features (Awada, Khoshgoftaar, Dittman, Wald & Napolitano 2012, Dudoit et al. 2002). Most commonly used feature selection methods involve ranking the genes based on some value of a univariate statistic, like the t-statistic, the F-statistic, or the Wilcoxon and Kruskal-Wallis statistics. A cut-off point based on either the number of genes or the p-value is imposed, to determine the number of variables to be used. Dudoit et al. (2002) suggest a gene selection method based on ranking. This is achieved by finding the ratio of between-group to within-group sum of squares (BSS/WSS) so that for a gene j,

$$BSS_j/WSS_j = \frac{\sum_i \sum_k I(y_i = k)(\bar{x}_{kj} - \bar{x}_{.j})^2}{\sum_i \sum_k I(y_i = k)(x_{ij} - \bar{x}_{kj})^2} \tag{11}$$

where $\bar{x}_{.j}$ and $\bar{x}_{kj}$ are the average expression levels of gene j and across all samples in class k, respectively. The p genes with the biggest ratio are selected. In this study, we adopted the (Dudoit et al. 2002) method of feature selection.

The preprocessing and the gene selection were performed using the recommendations of Dudoit et al. (2002). This decision stems from the fact that this method has been proven work relatively well in literature of several studies involving such as by Fort & Lambert-Lacroix (2005), Dudoit et al. (2002) among others and seem to perform relatively well. The top p genes were thus selected using Equation 11 for the implementation of the classification methods.

The classification error rates for the various methodologies when applied to the data under consideration are presented in Table 3.

The results indicate that KMA was the best, followed by RPLS, PLSDA. PLSGRDA performed equally well, while KNN emerged as the worst classifier, also in every comparison.

Table 3: classification error rates for the different methods when applied to the preprocessed data set

DATA	PLSGLR-log	PLSGLRDA	KNN	LDA	PLSDA	RPLS	SVM	KMA
Colon	16.4	13.3	26.7	15.0	11.7	11.7	14.8	11.2

7. Summary and Conclusions

In this study, two extensions of the PLSGLR were considered in addition to the KMA for a comparative study with some classical classification methodologies, namely KNN, LDA, PLSDA, RPLS and SVM, when applied to one commonly used microarray data set. The data were considered when un-preprocessed and when preprocessed. For both the un-preprocessed and preprocessed cases, the KMA emerged as a clear "winner" based on lower classification error rates. The KMA algorithm can therefore be recommended for classification problems involving noisy and non-noisy data. This could be due to the fact that the chosen kernels map the samples to a higher dimensional space, where they become linearly separable. This leads to a better classification ability by the KMA. Furthermore, the three new algorithms can therefore be considered as an addition to the existing literature for the microarray data classification problems.

Acknowledgements

We acknowledge the partial support from the Mexico's Consejo Nacional de Ciencias y Tecnología (CONACyT) project number 252996. Part of this work was done when A.W was a PhD Candidate at CIMAT, AC. Guanajuato, Gto, México (Wagala 2018).

[Recibido: agosto de 2019 — Aceptado: enero de 2020]

References

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D. & Levine, A. J. (1999), 'Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays', *Proceedings of the National Academy of Sciences of the United States of America* **96**(12), 6745–6750.

Alshamlan, H. M., Badr, G. & Alohali, Y. (2013), A study of cancer microarray gene expression profile: Objectives and approaches, *in* 'Proceedings of the World Congress on Engineering', Vol. II, London.

Awada, W., Khoshgoftaar, T. M., Dittman, D., Wald, R. & Napolitano, A. (2012), A review of the stability of feature selection techniques for bioinformatics data, *in* '2012 IEEE 13th International Conference on Information Reuse & Integration (IRI)', IEEE, pp. 356–363.

Bastien, P., Vinzi, E. V. & Tenenhaus, M. (2005), 'PLS generalised linear regression', *Computational Statistics and Data Analysis* **48**, 17–46.

Boulesteix, A. L., Strobl, C., Augustin, T. & Daumer, M. (2008), 'Evaluating microarray-based classifiers: an overview', *Cancer informatics* **6**, 77–97.

Chun, H. & Keles, S. (2009), 'Sparse partial least squares regression for simultaneous dimension reduction and variable selection', *Journal of the Royal Statistical Society. Series B, Statistical Methodology* **72**(1), 325. *http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2810828/

Chung, D. & Keles, S. (2010), 'Sparse partial least squares classification for high dimensional data', *Statistical Applications in Genetics and Molecular Biology* **9**(1), 17.

Dalmau, O., Alarcón, T. E. & González, G. (2015), 'Kernel multilogit algorithm for multiclass classification', *Computational Statistics and Data Analysis* **82**, 199–206.

Dong, K., Zhang, F., Zhu, Z., Wang, Z. & Wang, G. (2014), 'Partial least squares based gene expression analysis in posttraumatic stress disorder', *European Review for Medical and Pharmacological Sciences* **18**, 2306–2310.

Dudoit, S., Fridlyand, J. & Speed, T. (2002), 'Comparison of discrimination methods for the classification of tumors using gene expression data', *Journal of the American Statistical Association* **97**(457), 77–86.

Fort, G. & Lambert-Lacroix, S. (2005), 'Classification using partial least squares with penalized logistic regression', *Bioinformatics* **7**, 1104–1111.

Gagnon-Bartsch, J. A. & Speed, T. P. (2011), 'Using control genes to correct for unwanted variation in microarray data', *Biostatistics* **13**(3), 539–552. *http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3577104/

Gromski, S., Muhamadali, H., Ellis, D., Xu, Y., Correa, E., Turner, M. & Goodcare, R. (2015), 'A tutorial review: Metabolomics and partial least squares-discriminant analysis a marriage of convenience or a shotgun wedding', *Analytica Chimica Acta* **879**, 10–23.

Gusnanto, A., Ploner, A., Shuweihdi, F. & Pawitan, Y. (2013), 'Partial least squares and logistic regression random-effects estimates for gene selection in supervised classification of gene expression data', *Journal of Biomedical Informatics* pp. 697–709.

Höskuldsson, A. (1988), 'PLS regression methods', *Journal of Chemometrics* **2**, 211–228.

Huang, C. C., Tu, S. H., Huang, C. H., Lien, H. H., Lai, L. H. & Chuang, E. (2013), 'Multiclass prediction with partial least square regression for gene expression data: Applications in breast cancer intrinsic taxonomy', *BioMed Research International* pp. 1–9.

Lê Cao, K., Rossouw, D., Robert-Granieé, C. & Besse, P. (2008), 'A Sparse PLS for variable selection when integrating omics data', *Statistical Applications in Genetics and Molecular Biology* **7**(1).

Lee, D., Lee, W., Lee, Y. & Pawitan, Y. (2011), 'Sparse partial least-squares regression and its applications to high-throughput data analysis', *Chemometrics and Intelligent Laboratory Systems* **109**(1), 1–8.

Nguyen, D. V. & Rocke, D. M. (2002*a*), 'Multi-class cancer classification via partial least squares with gene expression profiles', *Bioinformatics* **18**(9), 1216–1226.

Nguyen, D. V. & Rocke, D. M. (2002*b*), 'Tumor classification by partial least squares using microarray gene expression data', *Bioinformatics* **18**(1), 39–50.

Telaar, A., Liland, K., Repsilber, D. & Nürnberg, G. (2013), 'An extension of PPLS-DA for classification and comparison to ordinary PLS-DA', *PLoS ONE 8* **2**, e55267.

Wagala, A. (2018), Problems in Statistical Genetics: Classification and Testing for Network Changes, PhD thesis, Centro de Investigación en Matemáticas A. C., Department of Probability & Statistics.
*https://cimat.repositorioinstitucional.mx

Wang, A., An, N., Chen, G., Li, L. & Alterovitz, G. (2015), 'Improving plsrfe based gene selection for microarray data classification', *Computers in Biology and Medicine* **62**, 14–24.

Wold, S., Ruhe, A., Wold, W. & Dunn III, W. J. (1984), 'The collinearity problem in linear regression, the partial least squares approach to generalized inverses', *SIAM Journal on Scientific and Statistical Computing* **5**(3), 735–743.

Wold, S., Sjöström, M. & Erikson, L. (2001), 'PLS-regression: A basic tool of chemometrics.', *Chemometrics and Intelligent Laboratory Systems* **58**, 109–130.

Xi, B., Gu, H., Baniasadi, H. & Raftery, D. (2014), 'Statistical analysis and modeling of mass spectrometry-based metabolomics data', *Methods Mol Biol.* **1198**, 333–353.