International Journal on
Advanced Science
Engineering
Information Technology

Vol.10 (2020) No. 4
ISSN: 2088-5334

Weighted L_1-norm Logistic Regression for Gene Selection of Microarray Gene Expression Classification

Aiedh Mrisi Alharthi[a,b,1], Muhammad Hisyam Lee[a,2], Zakariya Yahya Algamal[c]

[a] *Department of Mathematical Sciences, Universiti Teknologi Malaysia, Johor, Malaysia*
E-mail: [1]aiedh.harthi@gmail.com; [2]mhl@utm.my

[b]*Department of Mathematics, Taif University, Saudi Arabia.*

[c]*Department of Statistics and Informatics, University of Mosul, Mosul, Iraq*
E-mail: zakariya.algamal@uomosul.edu.iq

Abstract— **The classification of cancer is a significant application of the DNA microarray data. Gene selection methods are ordinarily used handle the issue of high-dimensionality of microarray data to enable experts to diagnose and classify cancer with high accuracy. The penalized logistic regression (PLR) technique is usually used in the dimensionality reduction of the high-dimensional gene expression data sets to remove irrelevant and redundant predictors from the binary logistic regression model. One of the regularization techniques used to achieve this goal is the least absolute shrinkage and selection operator (Lasso). However, this technique has been criticized for being biased in the selection of genes. The adaptive Lasso was usually proposed by assigning an initial weight to each gene to address the selection bias. This paper is concerned with adapting PLR to improve its capability in classification and gene selection, in the sense of accuracy, by introducing the one-dimensional weighted Mahalanobis distance (1-DWM) for each gene as an initial weight inside L_1-norm. By experiments, this proposed method, denoted by adaptive penalized logistic regression (APLR), gives more accurate results compared with other famous methods in this regard. The proposed method is applied to some real high-dimensional gene expression data sets in order to demonstrate its efficiency in terms of classification accuracy and selection of gene. Therefore, the proposed method could be utilized in other studies implementing gene selection in the area of classification of high dimensional cancer data sets.**

Keywords— **lasso; adaptive lasso; logistic regression; classification; weighted lasso.**

I. Introduction

Recently, new technologies have been developed to deal with the rapid growth of statistical data to help researchers analyze big data and convert it into useful information. Big data might have unnecessary features because most of them are irrelevant or redundant. Therefore, the purpose of selecting essential features is to choose a small subset of the significant features of the original data set. The selection of features not only speeds up the learning process but also improves the work of the model [1]. By using microarray technology, researchers can classify both cancerous and normal tissues, depending on gene expression profiles. Recently, many studies used gene expression data to determine types of cancer and predict clinical outcomes in order to diagnose patients with cancer[2]–[4].

Microarray data of gene expression has many properties that hinder the evolution of these techniques. One of these properties is the high-dimensionality of data sets. That is, the gene expression data set contains a considerable number of genes p, with a small number of observations n. This means that the gene expression data set matrix has columns very much larger than rows, $p > n$ [5]. Another problem is that microarray data usually suffer from a high level of technical noise. Therefore, it is very crucial to overcome these two problems in order to reasonably improve the accuracy of classification associated with microarray data [6].

In recent years, statisticians have developed many approaches for feature selection. These approaches fall into three main categories. First, the filter category that involves the most popular methods for feature selection, where each feature is independently examined regardless of its group performance. Second, the wrapper category. It uses various algorithms to evaluate the process of selecting feature groups. Although the wrapper methods are more efficient in feature selection than the filter methods, they are usually computationally costly, such as forward feature selection and backward feature elimination. Third, the embedded category, which combines the advantages of filter and wrapper approaches. It constitutes regularization (penalizing) methods that can concurrently perform both model selection and feature selection [1], [7], [8].

Associated with classification in gene expression data is the logistic regression. Computationally, the training time of applying logistic regression increases as the number of the genes in gene expression data sets increases and becomes complex to compute [9]–[11]. Logistic regression has some limitations. For example, it cannot automatically perform a selection of genes, although it outperforms other methods in classifying gene expression data [12]. Penalized methods are very effective embedded gene selection methods, which is connected with many popular classification methods. Recently, logistic regression, as a sparse classification method, received tremendous attention. It combines the logistic regression with a penalty for performing gene selection and classification simultaneously. Several logistic regression models can be applied with different penalties, among these penalties are, L_1-norm, which is called the least absolute shrinkage and selection operator (Lasso) [13], smoothly clipped absolute deviation (SCAD) [14], Elastic net [15], and adaptive L_1-norm [16].

The L_1-norm penalty model is one of the most popular procedures in the class of sparse methods. One of the drawbacks of the L_1-norm penalty model is that it equally penalizes all genes. For this reason, it inconsistently selects genes [14], [16]. To improve the process of gene selection, the present study proposes an adaptive logistic regression method by employing a certain weight inside the L_1-norm to classify patients concerning having cancer correctly. The used weight in the proposed method is proportional to the importance of each gene. This paper experimentally compares the proposed method and several other methods used in gene selection. The experimental results show that the proposed method outperforms the other methods in terms of classification accuracy. Besides this introduction, this paper is organized as follows. Section 2 is dedicated to the description of material and methodology. Section 3 presents the results and the experimental study that is intended to evaluate the efficiency of APLR compared to Lasso and ALasso. Finally, Section 4 concludes this paper.

II. Materials and Method

A. Penalized Logistic Regression

The logistic regression is a statistical classification method that is used to predict the value of the response variable when it is categorical with only two possible values that can be denoted by 0 and 1. The logistic regression works very well when the number of predictors is limited; however, when dealing with high-dimensional data sets such as gene expression data sets, the logistic regression method becomes inefficient in the sense that the errors of prediction increase as the number of predictors increase, and the computation of the predictors coefficients becomes cumbersome. Another problem that limits the use of logistic regression is overfitting, which happens when the number of predictors is by far larger than the number of observed values [17]. A famous method that has emerged recently to increase the efficiency of the logistic regression and improve its capability of classification is by regularization. The regularization technique can be done by penalizing the predictors and shrinking their coefficients. Three techniques are used in this regard: Lasso [13], ridge [18], and elastic net [15]. The Lasso algorithm can be used to sparse the high-dimensional gene expression datasets by shrinking most of the predictors' coefficients to zero. Although Lasso can do a good job in overcoming the overfitting problem, it is biased and lacks the oracle properties. It also does not encourage group selection. That is, it chooses only one or just few of the highly correlated predictors and shrinks the coefficients of the rest of the predictors to zero. Several authors adapted Lasso to overcome these problems and improve its efficiency in classification and collection. They introduced a new technique called ALasso (Adaptive Lasso) [16]. One of the ways to adapt Lasso is by introducing weights inside the L_1-norm. In this paper, a new weight inside the L_1-norm is introduced in the context of ALasso as we discuss in details in the next section.

Logistic regression is used to model a binary classification problem. This paper concentrates on a generic binary classification problem, where we have a set of data X representing the observed values of certain predictors. The data X can be represented as a $n \times p -$dimensional matrix. That is, $X = (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ where n is the sample size, the p-dimensional vector $x_i = (x_{i1}, x_{i2}, ..., x_{ip})$ represents the values of the p expression genes corresponding to the i^{th} observation of the response variable, and x_{ij} represents the value of the expression of the j^{th} gene for the i^{th} observation. y_i is i^{th} observation. It takes the value 0 or 1. The n observation can be classified into exactly two class. y_i represents the class of the i^{th} observation. In other words, $y_i = 0$ or 1 according to whether the i^{th} observation belongs to the first or second class, respectively. In logistic regression, the response variable y has a Bernoulli distribution. The probability of y is equal to 1, given *x*, denoted as $\pi(x)$, is given by

$$p(y_i = 1 | x_{ij}) = \pi(x_j) = \frac{e^{x'_j \beta}}{1+e^{x'_j \beta}}, j = 1,2,...,p \tag{1}$$

$$f(y_i) = \pi_i^{y_i} (1-\pi_i)^{1-y_i}, i = 1,2,...,n \tag{2}$$

The likelihood function is given as

$$L(\beta, y_i) = \prod_{i=1}^{n} f(y_i) = \prod_{i=1}^{n} \pi_i^{y_i} (1-\pi_i)^{1-y_i} \tag{3}$$

Then, the log- likelihood function becomes

$$\begin{aligned} \ell(\beta, y_i) = \sum_{i=1}^{n} \{ & y_i \log \pi(x_i) \\ & + (1-y_i) \log(1-\pi(x_i))\} \end{aligned} \tag{4}$$

The penalized logistic regression (PLR) is given by

$$PLR = \ell(\beta, y_i) = \sum_{i=1}^{n}\{y_i \log \pi(x_i) + (1-y_i)\log(1-\pi(x_i))\} + \lambda g(\beta) \tag{5}$$

where $g(\beta)$ is the penalty term and λ is a tuning parameter $(\lambda \geq 0)$. It regulates the amount of penalty (strength of shrinkage) that predictor variables are exposed to. That is, when λ increases, the magnitude of the penalty term increases. The value of the tuning parameter λ depends on the data. Therefore, its value can be determined using a cross-validation method [17], which can be implemented using the R package glmnet. Before solving the PLR maximization problem, it is assumed that the genes are standardized so that

$$\sum_{i=1}^{n} y_i = 0, \frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0, \text{ and } \sum_{i=1}^{n} x_{ij}^2 = 1,$$

for $j = 1, 2, ..., p$, in order to make the intercept (β_0) equal to zero.

B. Lasso Regression

Lasso is one of the most popular penalties. It was introduced by Tibshirani [13] to employ L$_1$–norm penalty to the coefficients of predictors. That is; it performs variable selection by shrinking the coefficients of some predictors exactly to zero. Lasso performs continuous shrinkage and variable selection at the same time. To obtain the estimates of Lasso, the log-likelihood is maximized when the penalty term is given by

$$g(\beta) = \sum_{j=1}^{p} |\beta_j| \tag{6}$$

The equation of the penalized logistic regression using Lasso is

$$PLR = \sum_{i=1}^{n}\{y_i \log \pi(x_i) + (1-y_i)\log(1-\pi(x_i))\} + \lambda \sum_{j=1}^{p} |\beta_j| \tag{7}$$

By choosing the appropriate penalty term described in (6), the coefficients of some predictors become exactly zero. This means that Lasso performs variable selection. Therefore, Lasso gives sparse solutions. The maximum likelihood solution of Eq. (7) is

$$\hat{\beta}_{LASSO} = \arg\min_{\beta}\left[\sum_{i=1}^{n}\{y_i \log \pi(x_i) + (1-y_i)\log(1-\pi(x_i))\} + \lambda \sum_{j=1}^{p} |\beta_j|\right] \tag{8}$$

In other words,

$$\hat{\beta}_{LASSO} = \arg\min_{\beta}\left[\sum_{i=1}^{n} \ell(\beta, y_i) + \lambda \sum_{j=1}^{p} |\beta_j|\right]. \tag{9}$$

Lasso can do a good job in selecting genes, but it has three deficiencies [9], [19]. The first deficiency is related to the number of genes that Lasso selects. In gene expression data sets, the number of features, p, is usually by far larger than the number of observations, n, in the sample. Unfortunately, Lasso cannot select more genes than the number of observations; that is, the number of genes Lasso selects is bounded above by n. The second deficiency is related to the way the genes work. Naturally, genes work as clusters or groups. Each group constitutes highly correlated genes. It is expected that Lasso takes this into account when selecting genes. That is, it is expected either to select the whole group of highly correlated genes (if they really are related to the disease) or to leave it all (if they are unrelated). Unfortunately, Lasso selects only one or a few members of each highly correlated group of genes that are related to the disease. Zou and Hastie [15] proposed a regularization method (called the elastic net) to overcome the first and second deficiencies. The elastic net method employs L$_1$–norm and L$_2$–norm penalties. The third deficiency of Lasso is that it is biased in gene selection. The reason is that it employs the same penalty to all gene coefficients. As a result of this shortcoming, Lasso lacks the oracle properties (see [14]). Concerning the last deficiency of Lasso with respect to the lack of the oracle properties, Zou [16] developed a new regularization technique called the adaptive Lasso technique, where different weights are employed inside the L$_1$–norm penalty to penalize different coefficients. In the adaptive Lasso, adaptive weights are used for penalizing different coefficients in the L$_1$–norm penalty.

The adaptive Lasso technique was first introduced by Zou [16] to correct the Lasso's overestimation behavior by replacing the L$_1$-penalty by a re-weighted version [20]. That is, Zou amended the L$_1$-penalty by assigning different weights to different coefficients. The assigned weights can be based on Ridge, Lasso, or other shrinkage techniques. In this paper, the lasso resulting from the first stage is used as the initial estimator for the coefficients. The penalized logistic regression model using adaptive Lasso (ALasso) is defined as

$$\hat{\beta}_{LASSO} = \arg\min_{\beta}\left[\sum_{i=1}^{n}\{y_i \log \pi(x_i) + (1-y_i)\log(1-\pi(x_i))\} + \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{\left(|\hat{\beta}_j^{initial}|\right)^{\gamma}}\right] \tag{10}$$

where $\lambda, \gamma \geq 0$ and $\hat{\beta}_j^{initial}$ is an initial estimate for each β_j estimated using the Lasso technique. Here we set $\gamma = 1$, for simplicity. Eqs. (8)-(10) can be solved by using a popular method called "coordinate descent algorithm" [21].

C. The Proposed Method

The work in this paper is motivated by the fact that although the PLR method can be applied to the high-dimensional data sets using L_1-norm penalty, this method may lead to the selection of irrelevant and redundant predictors (genes), because the L_1-norm is inconsistent concerning variable selection. In other words, the estimates of the PLR using L_1-norm penalty might be biased for large coefficients because they receive larger penalties [7]. This work is also motivated by the fact that in PLR, the genes are usually standardized, although the standardization process may be unreasonable when the variances of genes are essential.

In order to extend the effect of the individual gene to joint effect of multigene Peng *et al.* [22] used the one-dimensional weighted Mahalanobis distance (1-DWM) as the criterion of gene effectiveness, that is defined as

$$J(x_{.j}) = \frac{(\bar{x}_{1j} - \bar{x}_{2j})^2}{\sigma_{wj}^2}, \tag{9}$$

where x^{j} is a column vector, denote the expression level of gene j across samples, and $\sigma_{wj}^2 = w_{1j}.\sigma_{1j}^2 + w_{2j}.\sigma_{2j}^2$, denotes the weighted variance of gene j, σ_{kj}^2 denotes the expression level variance of gene j in class k, w_k is the prior probability or weight of class k, where k in this paper is 2; i.e., we have exactly two classes and $w_1 = w_2 = 0.5$. Therefore, in order to improve selection of genes and ensure high classification accuracy, this paper uses the (1-DWM) of Peng *et al.* for each gene as an initial weight inside L_1-norm.

The j^{th} component of the *p*-dimensional vector of genes is given by

$$w_j = \frac{1}{|J(x_{.j})|}, \quad j = 1, 2, ..., p, \tag{10}$$

where $J(x_{.j})$ is the weight for every gene j that is defined as equation (11).

To reduce inconsistency in feature selection, the proposed weight in this paper gives the gene with a low value of ratio a relatively large amount of weight, while it gives the gene with a high value of ratio a small weight. Upon appropriately assigning weights to features, the PLR becomes capable of accurately selecting related features. The algorithm of implementing the APLR method is given in Algorithm. The existence of a global maximum point of the APLR solution is guaranteed by the fact that the APLR equation is convex. The APLR solution can be obtained by coordinate descent method.

Algorithm: The Computation of APLR

```
Step 1.  Split each gene $x_{.j}$ on the basis of the value of $y$
           into two classes $x_{1j}$ and $x_{2j}$ .
Step 2.  Find mean of $x_{1j}$ and $x_{2j}$
Step 3.  Find variance of $x_{1j}$ and $x_{2j}$
Step 4.  Compute $\sigma_{wj}^2 = (0.5)(\sigma_{1j}^2) + (0.5)(\sigma_{2j}^2)$
Step 5.  Compute $J(x_{.j}) = \frac{(\bar{x}_{1j} - \bar{x}_{2j})^2}{\sigma_{wj}^2}$
Step 6.  Find $w_j, j = 1, 2, \ldots, p$
Step 7.  Define $\tilde{\mathbf{x}}_i = w_j \mathbf{x}_i$
Step 8.  Solve the APLR
```

$$\hat{\beta}_{APLR} = \arg\min_{\beta} \left[\sum_{i=1}^{n} \{ y_i \log \pi(x_i) + (1 - y_i) \log(1 - \pi(x_i)) \} + \lambda \sum_{j=1}^{p} w_j \, |\beta_j| \right]. \tag{11}$$

D. Dataset Description

The proposed method (APLR) is applied to five binary logistic cancer classification data sets in order to evaluate its performance and demonstrate its advantages over the other competitive methods. The detailed information of these data sets is summarized in Table I.

TABLE I
THE USED DATASETS

Dataset	Samples	Genes	Classes
Colon	62	2000	Tumor / Normal
Prostate	102	12600	Tumor / Non-tumor
DLBCL	77	7129	DLBCL / FL
Breast	168	2905	Benign /Malignant
Sco	54	22283	Sick/Normal

The first data set is the colon cancer, where the number of observations (gene expression levels) is 62 (40 cancerous tumors and 22 noncancerous tissues), and the number of features is 6500 genes. The colon cancer data set is obtained with the use of Affymetrix oligonucleotide array technology. In this data set, only 2000 gene expressions was used, according to the highest minimal intensity across the samples [23]. The second data set is prostate cancer. It contains 12600 genes. The sample contains 52 patients with cancerous prostate tumor and 50 other patients with non-cancerous tumor tissues [24]. The third data set is known as the diffuse large B-cell lymphoma (DLBCL), which consists of 77 observations for gene expressions, which is divided into 58 samples of diffuse large B-cell lymphomas (DLBCL) and 19 samples of follicular lymphoma (FL). The number of gene expression values in each sample is 7,129 [25]. The fourth data set is breast cancer (Breast) that includes the microarray data from 189 invasive breast *carcinomas*, and three published gene expression datasets from breast carcinomas [26]. The fifth data set is known as the Sarcoma data (Sco). The Sarcoma data set consists of expression profiles of 22,283 human genes from 54 patients, where 15 are normal, and 39 are sick with the disease [26].

E. Performance Evaluation

In this subsection, the predictive performance of the proposed method is evaluated and then compared with other sparse methods; three performance metrics are measured for both the training and testing datasets. These metrics include classification accuracy (*CA*), sensitivity (*SEN*), and specificity (*SPE*).

These criteria are defined as

$$CA = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \tag{12}$$

$$SEN = \frac{TP}{TP + FN} \times 100\% \tag{13}$$

$$SPE = \frac{TN}{FP + TN} \times 100\% \tag{14}$$

where TP is the number of true positive, FP is the number of false positive, TN is the number of true negative, and FN is the number of false negative. The higher the values of the used evaluation criteria, the better the classification performance is.

III. Results and Discussion

A. Experimental Setting

The proposed method (APLR) was shown to be effective through comparative experiments with three other methods, namely Lasso, SCAD, and ALasso. These methods along with APLogiR are applied to the data sets described above. We proceed as follows. To perform cross-validation (CV), each data set is randomly split into two partitions, namely the training set and the testing set. The training set consists of 70% of the data and the testing set consists of the rest 30% of the data. For examining the effect of the data partitions, the above methods are evaluated along with the proposed method, for their performance in classification, using 10-fold CV. The result is the average of 100 replications of the experiment. The value of the tuning parameter λ for each method was allowed a value in the interval [0, 100]. For the SCAD penalty, the constant a was set to 3.7 as Fan and Li [14] suggest it. All the replications were implemented in R using glmnet.

B. Experimental Results

Table II below shows the average of the following numerical measures for the five high-dimensional gene expression microarray data sets used in this study (colon, prostate, DLBCL, breast, and Sarcoma): the number of genes selected by each method, the accuracy of classification (%) (CA), the sensitivity (%) (SEN), and the specificity (%) (SPE) in both the training and testing data sets. To compare the different methods, the performance of Lasso, smoothly clipped absolute deviation (SCAD), and Adaptive Lasso (ALasso) have been also evaluated for the five data sets.

TABLE II
THE AVERAGED OF THE EVALUATION METRICS OVER 100 TIMES FOR THE FIVE DATA SETS

			Training set			**Testing set**		
Dataset	**Methods**	**Genes**	**% CA**	**% SEN**	**% SPE**	**% CA**	**% SEN**	**% SPE**
Colon	Lasso	14	94.14	92.20	94.64	79.53	78.43	76.92
	SCAD	14	94.81	92.51	95.51	79.51	80.13	67.90
	ALasso	14	94.83	93.22	95.22	78.40	81.41	81.91
	Proposed	12	96.12	95.85	96.83	82.91	82.42	83.33
Prostate	Lasso	29	99.82	99.63	100.00	91.14	88.34	91.74
	SCAD	28	99.82	99.71	100.00	60.13	45.63	72.54
	ALasso	28	99.91	99.72	100.00	82.11	56.41	91.91
	Proposed	24	100.00	100.00	100.00	93.53	91.82	94.43
DLBCL	Lasso	24	99.80	99.60	99.60	88.33	83.44	91.52
	SCAD	24	99.81	100.00	99.10	74.02	52.23	92.32
	ALasso	24	99.93	100.00	99.81	84.41	59.02	94.33
	Proposed	22	100.00	100.00	100.00	91.32	85.72	95.22
Breast	Lasso	30	93.80	97.50	89.91	71.34	67.12	75.44
	SCAD	31	93.82	97.62	90.01	64.11	50.03	75.23
	ALasso	31	93.90	97.71	90.03	68.53	58.02	75.42
	Proposed	27	95.61	98.83	92.42	75.44	72.74	81.42
Sco	Lasso	20	99.71	100.00	99.52	88.50	87.73	89.32
	SCAD	20	99.74	100.00	99.31	75.51	51.04	91.54
	ALasso	20	99.64	100.00	99.24	84.62	67.24	92.82
	Proposed	15	100.00	100.00	100.00	93.34	88.23	97.43

Our first observation is that our proposed method, APLR, has the lowest average number of selected genes among all other methods. For instance, in prostate cancer, the number of genes selected by APLR is 24 genes compared to 29, 28, and 28 genes selected by Lasso, SCAD, and ALasso, respectively. On the other hand, we found that Lasso gives the highest number of selected genes.

We also observe that in each of the data sets used in this study, the average classification accuracy, sensitivity, and specificity in both the training and testing sets of APLR are slightly larger than that of Lasso, SCAD, and ALasso. For example, in breast data, the classification accuracy of APLR in the training set is approximately (96%), which is higher than (94%), nearly, for Lasso, SCAD, and ALasso. Moreover, in colon data, the sensitivity of APLR is 95.85%, which is greater than that of Lasso, SCAD and ALasso, 92.2%, 92.5%, and 93.2%, respectively. The same observation can be concluded in the testing sets.

Another important note is that the specificity of APLR is much better than that of Lasso, SCAD, and ALasso in all the datasets. For the colon dataset, the specificity of APLR is better than that of ALasso and Lasso. For example, in DLBCL, the specificity of APLR in the testing set is 95.22%, which is greater than 94.33% for ALasso, 92.32% for SCAD, and 91.52% for Lasso. Therefore, in terms of the specificity in training sets in all data sets, either APLR is better than the other methods or almost the same. Overall, it is clear that the selection and classification performance of our proposed method APLR is the best compared to Lasso, SCAD, and Alasso. This asserts that our proposed method takes the weight of each gene into consideration during the selection and classification process.

IV. Conclusions

Accuracy of prediction is a desired goal of classification methods regarding high-dimensional microarray gene expressions data sets. The accuracy increases as the number of selected features (genes) decrease. This paper proposes an effective feature selection method, APLR, that simultaneously increases accuracy and decreases the number of selected features. It has been experimentally shown in this paper that the proposed method outperforms the other competitive methods in terms of accuracy, namely Lasso, SCAD, and ALasso. The proposed method, which was implemented with R, was successfully tested on five different publicly-known data sets. The results of the experiments assert that APLR is an efficient method of feature selection and classifications. APLR also can be a starting point for developing other regularizations and feature selection methods.

References

[1] X. Y. Liu, Y. Liang, S. Wang, Z. Y. Yang, and H. S. Ye, “A Hybrid Genetic Algorithm with Wrapper-Embedded Approaches for Feature Selection,” *IEEE Access*, vol. 6, pp. 22863–22874, 2018.

[2] K. Kourou, T. P. Exarchos, K. P. Exarchos, M V Karamouzis, and D. I. Fotiadis, “Machine learning applications in cancer prognosis and prediction,” *Comput. Struct. Biotechnol. J.*, vol. 13, pp. 8–17, 2015.

[3] H. R. Arabnia and Q. N. Tran, *Emerging trends in applications and infrastructures for computational biology, bioinformatics, and systems biology: systems and applications*. Morgan Kaufmann, 2016.

[4] Z. Y. Algamal and M. H. Lee, “A two-stage sparse logistic regression for optimal gene selection in high-dimensional microarray data classification,” *Adv. Data Anal. Classif.*, 2018.

[5] Z. Y. Algamal and M. H. Lee, “Penalized logistic regression with the adaptive LASSO for gene selection in high-dimensional cancer classification,” *Expert Syst. Appl.*, vol. 42, no. 23, pp. 9326–9332, 2015.

[6] Z.-Y. Yang, Y. Liang, H. Zhang, H. Chai, B. Zhang, and C. Peng, “Robust Sparse Logistic Regression With the Lq (0 < q < 1) Regularization for Feature Selection Using Gene Expression Data ZIYI,” *IEEE Access*, vol. 6, pp. 68586–68595, 2018.

[7] Z. Y. Algamal and H. T. Mohammad Ali, “An efficient gene selection method for high-dimensional microarray data based on sparse logistic regression,” *Electron. J. Appl. Stat. Anal.*, vol. 10, no. 1, pp. 242–256, 2017.

[8] H. H. Hsu, C. W. Hsieh, and M. Da Lu, “Hybrid feature selection by combining filters and wrappers,” *Expert Syst. Appl.*, vol. 38, no. 7, pp. 8144–8150, 2011.

[9] Z. Y. Algamal, “Classification of gene expression autism data based on adaptive penalized logistic regression,” *Electron. J. Appl. Stat. Anal.*, vol. 10, no. 2, pp. 561–571, 2017.

[10] Y. Asar and A. Genç, “New shrinkage parameters for the Liu-type logistic estimators,” *Commun. Stat. Comput.*, vol. 45, no. 3, pp. 1094–1103, 2016.

[11] D. Inan and B. E. Erdogan, “Liu-type logistic estimator,” *Commun. Stat. Comput.*, vol. 42, no. 7, pp. 1578–1586, 2013.

[12] Y. Liang *et al.*, “Sparse logistic regression with a L 1/2 penalty for gene selection in cancer classification,” *BMC Bioinformatics*, vol. 14, no. 1, p. 198, 2013.

[13] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. R. Stat. Soc. Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[14] J. Fan and R. Li, “Variable selection via nonconcave penalized likelihood and its oracle properties,” *J. Am. Stat. Assoc.*, vol. 96, no. 456, pp. 1348–1360, 2001.

[15] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *J. R. Stat. Soc. Ser. B (statistical Methodol.*, vol. 67, no. 2, pp. 301–320, 2005.

[16] H. Zou, “The adaptive lasso and its oracle properties,” *J. Am. Stat. Assoc.*, vol. 101, no. 476, pp. 1418–1429, 2006.

[17] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, vol. 112. Springer, 2013.

[18] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[19] S. Wang, B. Nan, S. Rosset, and J. Zhu, “Random lasso,” *Ann. Appl. Stat.*, vol. 5, no. 1, p. 468, 2011.

[20] Bühlmann, Geer, P. and Van De, and Sara, *Statistics for High-Dimensional Data: Methods, Theory and Applications.* 2011.

[21] J. Friedman, T. Hastie, and R. Tibshirani, “Regularization paths for generalized linear models via coordinate descent,” *J. Stat. Softw.*, vol. 33, no. 1, p. 1, 2010.

[22] H. Peng, Y. Fu, J. Liu, X. Fang, and C. Jiang, “Optimal gene subset selection using the modified SFFS algorithm for tumor classification,” *Neural Comput. Appl.*, vol. 23, no. 6, pp. 1531–1538, 2013.

[23] U. Alon *et al.*, “Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays,” *Proc. Natl. Acad. Sci.*, vol. 96, no. 12, pp. 6745–6750, 1999.

[24] D. Singh *et al.*, “Gene expression correlates of clinical prostate cancer behavior,” *Cancer Cell*, vol. 1, no. 2, pp. 203–209, 2002.

[25] M. A. Shipp *et al.*, “Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning,” *Nat. Med.*, vol. 8, no. 1, p. 68, 2002.

[26] Q. Shen, Z. Mei, and B. X. Ye, “Simultaneous genes and training samples selection by modified particle swarm optimization for gene expression data classification,” *Comput. Biol. Med.*, vol. 39, no. 7, pp. 646–649, 2009.