33rd Annual International Conference of the IEEE EMBS
Boston, Massachusetts USA, August 30 - September 3, 2011

Sparse Representation via ℓ_1-minimization for Underdetermined Systems in Classification of Tumors with Gene Expression Data

R. Sánchez, M. Argáez, P. Guillén

Abstract— **The development of cancer diagnosis models and cancer discovery from DNA microarray data are of great interest in bioinformatics and medicine. In pattern recognition and machine learning, a classification problem refers to finding an algorithm for assigning a given input data into one of several categories. Many natural signals are sparse or compressible in the sense that they have short representations when expressed in a suitable basis. Motivated by the recent successful algorithm developments for sparse signal recovery, we apply the selective nature of sparse representation to perform the above mentioned classification. In order to find such sparse representation we implement an ℓ_1-minimization algorithm. This methodology overcomes the lack of robustness with respect to outliers. In contrast to other classification algorithms, no model selection dependency is involved. The minimization algorithm is a convex relaxation-like that has been proven to efficiently recover sparse signals. To study its performance, the proposed method is applied to six tumor gene expression datasets and numerically compared with various support vector machine methods (SVM). The numerical results show that the ℓ_1-minimization algorithm proposed performs at least comparably and often better than SVMs.**

I. Introduction

DNA microarray technology is a powerful and efficient tool for measuring relative gene activity or expression in a variety of applications, and has the potential to provide accurate and objective cancer diagnosis due to its high performance capability of measuring expression levels of tens of thousands of genes simultaneously.

The challenging issue in microarray technique is to analyze and interpret the large volume of data. One statistical technique commonly applied to microarray data is classification. Classification problems arise in many different applications such as data mining and knowledge discovery, data compression, pattern recognition and pattern classification, in order to group similar genes in one class so that genes within the same class are similar to each other and different from genes in other class [4],[13].

Manuscript received April 15, 2011. This research was performed at the University of Texas at El Paso and supported in part by the U.S. Army Research Laboratory, through the Army High Performance Computing Research Center, Cooperative Agreement W911NF-07-0027.

R. Sánchez, Program in Computational Sciences, University of Texas at El Paso, TX, 79968, USA reinaldosanar@gmail.com

M. Argáez, Program in Computational Sciences, University of Texas at El Paso, TX, 79968, USA, margaez@utep.edu

P. Guillén, Program in Computational Sciences, University of Texas at El Paso, TX, 79968, USA, pguillen@utep.edu, Universidad de Los Andes, Merida, Venezuela.

Many engineering and science applications involve solving linear inverse problems that are usually ill-conditioned and for which the use of regularization techniques is required to be able to propose useful solutions. Recently, regularization via sparsity constraints has become very popular, where we look for an approximate solution to a linear system of equations, with the requirement that it has as few nonzero components as possible. This kind of problems can be found in several applications in machine learning, image and signal processing, coding and information theory, among others.

Motivated by the recent successful algorithms proposed for sparse signal recovery problems in [1], [7], [14], we apply the selective nature of sparse representation to solve classification problems. A test sample is represented in an overcomplete dictionary with the training samples as base elements. In case we have sufficient training samples available for each class, test samples can be expressed as a linear combination of only those training samples that belong to the same class, therefore providing a naturally sparse representation.

Support Vector Machines (SVM) are powerful automatic learning structures, based on the statistical theory of learning, capable of resolving classification, regression and estimation problems. They have been the aim of much research in recent years. The method was proposed by V. Vapnik [18] in the late seventies for solving pattern recognition problems. However, the performance of SVM relies upon careful choice of model parameters. Currently, SVM is implemented in the GEMS-SVM software [17], for automated cancer diagnosis from microarray gene expression data.

This work presents a new method of supervised machine learning for classification of tumors using gene expression data. Specifically, the strategy for classification consists in finding a sparse representation of test samples with respect to training samples. The sparse representation is computed via ℓ_1-minimization. The new method is applied to six tumor gene expression datasets and its performance is compared with the results obtained from the GEMS-SVM software.

II. Materials and Methods

A. Dataset

In this numerical experimentation we use 6 different datasets from the GEMS-SVM software, freely available in MATLAB .mat format (http://www.gems-system.org/). A short description of the datasets used follows:

- 9_Tumors. The dataset comes from a study of 9 human tumor types: NSCLC, colon, breast, ovary, leukemia, renal, melanoma, prostate, and CNS.
- 11_Tumors. Consists of gene expression data of 11 various human tumor types: ovary, bladder/ureter, breast, colorectal, gastro-esophagus, kidney, liver, prostate, pancreas, adeno lung, and squamous lung.
- Prostate_Tumor. Binary dataset contains gene expression data of prostate tumor and normal tissues.
- Lung_Cancer. Dataset of 4 lung cancer types and normal tissues.
- SRBCT. Small, round blue cell tumors (SRBCT) of childhood.
- Brain_Tumor. Dataset from a study of 5 human brain tumor types: medulloblastoma, malignant glioma, AT/RT, normal cerebellum, and PNET.

In the Table 1, the number of samples and genes for each dataset is described.

Table 1. Information about the six datasets.

Dataset	# Samples	# Genes	# Classes
9_Tumors	60	5726	9
11_Tumors	174	12533	11
Prostate_Tumor	102	10509	2
Lung_Cancer	203	12600	5
SRBCT	83	2308	4
Brain_Tumor	90	5920	5

B. Problem Formulation

We say that a vector $\mathbf{c}\in\mathbb{R}^n$, has a sparse representation if it can be expressed as a linear combination of as few *base* elements as possible from a fixed collection matrix $B\in\mathbb{R}^{m\times n}$, called *dictionary*. In other words, given a dictionary $B\in\mathbb{R}^{m\times n}$ where the columns b_i have unit Euclidean norm, we say that the vector $\mathbf{c}\in\mathbb{R}^n$, has a *k-sparse* representation if $\mathbf{c}=\sum_{i=1}^{n} c_i b_i$ and $\|\mathbf{c}\|_0 \le k$.

The counting function $\|\cdot\|_0 : \mathbb{R}^n \to \mathbb{R}_0^+$ known as the ℓ_0 norm [11], gives the number of nonzero elements in its argument. That is, $\|\mathbf{c}\|_0 = \text{cardinal}\{i : c_i \neq 0\}$. Even though we call it the ℓ_0 norm, one can easily verify that it does not satisfy the positive homogeneity (positive scalability) property in the definition of a norm, namely we have that $\|\lambda\mathbf{c}\|_0 \neq |\lambda|\,\|\mathbf{c}\|_0$, for any given nonzero scalar λ. One of the research topics in Compressed Sensing consists of finding the vector $\mathbf{c}\in\mathbb{R}^n$, such that the $\|\mathbf{c}\|_0$ is as small as possible, subject to some restrictions. A vector $\mathbf{c}$ is said to be *nearly-sparse* if the rearranged entries of $\mathbf{c}$ decay rapidly when sorted in decreasing order of magnitude. Since compressible vectors are well approximated by sparse ones [6], the framework of Compressed Sensing applies to this class too.

In this work, we look for the sparsest vector $\mathbf{c}$ satisfying an underdetermined linear system of equations $B\mathbf{c} = \mathbf{y}$. Formally, we want to solve the following optimization problem

$$\min \|\mathbf{c}\|_0 \quad \text{subject to} \quad B\mathbf{c} = \mathbf{y}. \tag{1}$$

However, problem (1) is a combinatorial minimization problem and NP-hard (non-deterministic polynomial-time) [11], and therefore any algorithm that is intended to solve (1) given the matrix B and the vector $\mathbf{y}$, will be computationally intractable.

Strategies to overcome this difficulty have been developed giving rise to different algorithmic approaches with remarkable results in different applications. *Convex Relaxation* approaches where the objective function in problem (1) is replaced by the ℓ_1-norm convex function [10] have been proven to find the sparsest solution to the linear system of equations. That is, under some mild conditions [7], problem (1) is equivalent to the ℓ_1-minimization problem:

$$\min \|\mathbf{c}\|_1 \quad \text{subject to} \quad B\mathbf{c} = \mathbf{y}, \tag{2}$$

where the ℓ_1 norm is defined as

$$\|\mathbf{c}\|_1 = \sum_{i=1}^{n} |c_i|.$$

Now we have an optimization problem whose objective function is convex, unlike the ℓ_0-norm in problem (1). The motivation for this approach comes from studying the theory of *compressed sensing* (compressive sampling) which has been a research topic of interest in the last years. The work in this area initiated in late 2004 by Emmanuel Candès, Justin Romberg and Terence Tao [7], and independently by David Donoho [10]. The ℓ_1 convex relaxation approach has been proven to successfully find sparse solutions to linear system of equations.

In [1] the authors propose a Path Following Signal Recovery (PFSR) algorithm that finds solutions of a sequence of linear equality constrained multiquadratic problems, depending on a regularization parameter that converges to zero, approximating the ℓ_1-norm in a homotopic manner. Numerical experimentations have shown that the PFSR algorithm is capable of recovering sparse signals, with results comparing favorably - in both accuracy and CPU running time - with the state-of-the-art algorithms for finding sparse solutions to linear systems of equations. The MATLAB implementation of the PFSR algorithm can be found at http://www.math.utep.edu/Student/rsanchez/

The PFSR algorithm we propose in [1] finds an approximate solution of problems of type (2), by solving a sequence of subproblems of the form

$$\min \sum_{i=1}^{n} \left(c_i^2 + \mu\right)^{1/2} \quad \text{subject to } B\mathbf{c} = \mathbf{y}, \tag{3}$$

as the regularization parameter μ goes to zero. The algorithm generates two sequence of iterates. The first sequence (inner loop) generates a series of iterates for obtaining an approximate solution of subproblem (3) for a fixed regularization parameter $\mu>0$. The second sequence

(outer loop) creates a series of the approximate solutions for the subproblems (3) that converges to an optimal solution of the ℓ_1-minimization problem (2).

Algorithm 1. Path Following Signal Recovery (PFSR)

The PFSR Algorithm

Task: Find an approximate solution $\mathbf{c}$ to the problem
$\min \|\mathbf{c}\|_1$ subject to $B\mathbf{c} = \mathbf{y}$

Parameters: We are given the matrix B and the vector $\mathbf{y}$

Step 1. **Initialization:** Set: $\sigma,\ \tau,\ \mu,\ \epsilon_1$

Step 2. Initial approximate solution $\mathbf{c} = B^T\mathbf{y}$

Step 3. **Outer Loop** : **for** $j = 1, ..., maxiter$

Step 4. **Inner Loop** : Set $\mathbf{c}_- = \mathbf{c}$

Step 5. Update weight matrix: $D_\mu(\mathbf{c}_-) = diag(\mathbf{c}_-^2 + \mu)$

Step 6. Solve the fixed-point equation:

$$\begin{bmatrix} D_\mu(\mathbf{c}_-)^{-1/2} & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{s} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix}$$

Step 7. **Check proximity to the central path:**
if $\frac{\|\mathbf{c}-\mathbf{c}_-\|}{1+\|\mathbf{c}\|} \geq \sqrt{\mu}$ **go to** step 4

Step 8. Set $\tilde{\mathbf{c}} = |\mathbf{c}|$, $w = B^T\mathbf{s}$, $\tilde{z} = (\mathbb{1} - |w|)$

Step 9. Compute $error_{primal} = \frac{\|B\mathbf{c}-\mathbf{y}\|}{1+\|\mathbf{y}\|}$, $gap = \frac{\tilde{\mathbf{c}}^T\tilde{z}}{n}$

Step 10. **Stopping criteria for the problem:**
if $error_{primal} > \epsilon$ or $gap > \epsilon_1$
Update $\mu = \min\{\sigma gap, \tau\mu\}$, **go to** step 3
else
display '$\mathbf{c}$ is an optimal solution'

In Step 6 of Algorithm 1, $\mathbf{s}$ represents the Lagrange multiplier associated to the equality constraint in problem (3). It is important to notice that in our algorithm, Step 6 is reformulated and solved using a specially designed Conjugate Gradient (CG) algorithm. Specifically, for a current point $\mathbf{c}_-$, the first block of equations of the system in Step 6, gives $\mathbf{c} + D_\mu(\mathbf{c}_-)^{1/2} B^T\mathbf{s} = 0$, and since $D\mathbf{c} = \mathbf{y}$, we obtain the weighted normal equation

$$B\, D_\mu(\mathbf{c}_-)^{1/2}\, B^T\mathbf{s} = -\mathbf{y} \tag{4}$$

In order to solve (4) we apply a CG method and then compute the new approximation for $\mathbf{c}$ as in $\mathbf{c} = -D_\mu(\mathbf{c}_-)^{1/2} B^T\mathbf{s}$.

The initialization parameters σ, τ, μ, and ϵ_1 are used for defining the tolerance and regularization parameter within the algorithm.

C. Classification Problem

Let us consider a *training dataset* $\{(\mathbf{x}_i, t_i) : i=1,..., n\}$, $\mathbf{x}_i \in \mathbb{R}^d$, $t_i \in \{1, 2,..., N\}$, where n is the number of samples and N the number of classes. The vector $\mathbf{x}_i$ represents the i-th sample, containing *gene expression* values and t_i denotes the label of the i-th sample.

The sparse representation problem is formulated as follows: For a testing sample $\mathbf{y} \in \mathbb{R}^d$, find the sparsest representation from the training dataset. That is,

$$\begin{aligned} &\min \|\mathbf{c}\|_0 \\ &\text{s.t} \quad \mathbf{c}_1\mathbf{x}_1 + \mathbf{c}_2\mathbf{x}_2 + \ldots + \mathbf{c}_n\mathbf{x}_n = \mathbf{y}. \end{aligned} \tag{5}$$

We show that indeed a valid test sample $\mathbf{y}$ can be represented using only the training samples from the same class, therefore inducing a sparse representation

Let us rearrange the given n_i training samples from the i-th class as the columns of a submatrix $A_i = [\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,n_i}] \in \mathbb{R}^{d \times n_i}$. That is, we group all those samples with the same label into a submatrix A_i. In case we have sufficient training samples of the i-th class, any test sample $\mathbf{y}$ from the same class will be represented as a linear combination of the training samples associated with class i:

$$\mathbf{y} = c_{i,1}\mathbf{x}_{i,1} + c_{i,2}\mathbf{x}_{i,2} + \ldots + c_{i,n_i}\mathbf{x}_{i,n_i}, \tag{6}$$

for some values of $c_{i,j} \in \mathbb{R}$, $j = 1,\ldots,n_i$. Now, making use of the whole training data set, we define the matrix A of size $d \times n$ by concatenating all of the n training samples of the different N classes, that is $A = [A_1, A_2, \ldots, A_N]$. Then, the linear representation of the test sample $\mathbf{y}$ can be rewritten using all training samples, as in $\mathbf{y} = Av$, where $v = [0, \ldots, 0, c_{i,1}, c_{i,2}, \ldots, c_{i,n_i}, 0, \ldots, 0]^T \in \mathbb{R}^n$ is the vector whose entries are all zero except for those associated with the i-th class. We emphasize that for practical problems, the matrix A does not have to be rearranged, since the sparse representation will identify the nonzero components we are interested in.

We consider a construction error vector $\mathbf{e}$ on the measurements; therefore we obtain the following underdetermined linear system

$$A\mathbf{c} + \mathbf{e} = \mathbf{y}. \tag{7}$$

This system can be rewritten as $B\mathbf{d} = \mathbf{y}$, where

$$B = [A, I],\ \mathbf{d} = [\mathbf{c}, \mathbf{e}]^T.$$

Here I represents the $d \times d$ identity matrix, $B \in \mathbb{R}^{d \times (d+n)}$, and $\mathbf{d} \in \mathbb{R}^{n+d}$.

Now to find the solution to the sparse representation problem, we solve the ℓ_1-norm minimization problem

$$\begin{aligned} &\min \|\mathbf{d}\|_1 \\ &\text{s.t} \quad B\mathbf{d} = \mathbf{y}, \end{aligned} \tag{8}$$

for a given testing sample $\mathbf{y}$. One of the advantages of this formulation is that robustness with respect to outliers and noise in the dataset is gained when using the ℓ_1-norm [3], [9]. Also, we do not need to care for model selection, as in support vector machine approaches, since we take advantage of the selective nature of the sparse representation approach in order to perform classification.

D. Discriminant Functions and Classifier

Once the sparse vector $\hat{\mathbf{c}}$ has been found as the solution to (8), we identify the class to which the testing sample $\mathbf{y}$ belongs. The approach consists in associating the nonzero entries of $\hat{\mathbf{c}}$ with the columns of A corresponding to those

training samples having the same category of the testing sample $\mathbf{y}$.

The solution vector $\hat{\mathbf{c}}$ is decomposed as the sum of d-dimensional vectors $\hat{\mathbf{c}}_k$, where $\hat{\mathbf{c}}_k$ is obtained by keeping only those entries in $\hat{\mathbf{c}}$ associated with category k and assigning zeros to all the other entries. Later, we define the N discriminant functions

$$g_k(\mathbf{y}) = \| \mathbf{y} - A\,\hat{\mathbf{c}}_k \|_2, \quad k = 1,\ldots, N. \tag{8}$$

Thus g_k represents the approximation error when $\mathbf{y}$ is assigned to the category k. Finally we classify $\mathbf{y}$ in the category with the smallest approximation error. That is

$$t = \arg\min_{k=1,\ldots,N} g_k(\mathbf{y}). \tag{9}$$

E. K-fold-cross Validation

Performance is commonly measured by the error rate of the classifier on the entire population. Cross validation is a statistical method for evaluating machine learning algorithms in which the data is divided in two sets: one used for the training stage, and the second one used for testing (validation). These two training and testing sets should cross-over in consecutive rounds in such way that each sample in the data set has a chance of being validated.

In the case of K-fold cross validation, a K-fold partition of the dataset is created by splitting the data into K equally (nearly equal) sized subsets (folds), and then for each of the K experiments, K-1 folds are used for *training* and the remaining one for *testing*. A common choice for K-fold cross-validation is $K = 10$ (see [15]).

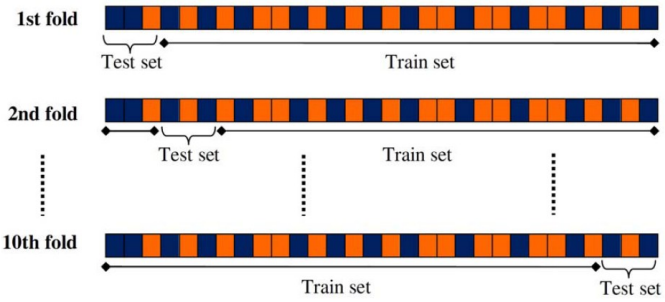

Fig. 1. A 10-fold cross validation partition for a binary dataset

III. RESULTS

We solve the classification problem using the solution of the sparse representation in (6). The matrix A is built using the dataset elements, and for our numerical experiment we normalize the columns of A in such a way that they all have unit norm.

The PFSR algorithm is applied to solve each of the problems of the form (6), which are needed at every iteration of a 10-fold cross-validation test. The PFSR algorithm and the complete validation experiment for each dataset are implemented in MATLAB. Notice that even though in our problem formulation we use the augmented matrix $B = [A, I]$, we do not need to store the complete matrix B but only A since

$$\begin{aligned} B\,\mathbf{d} &= A\mathbf{c} + \mathbf{e}, \\ B^T\mathbf{y} &= [A^T\mathbf{y}, \mathbf{y}]^T. \end{aligned} \tag{9}$$

Thus, we implement in a fast way the matrix-vector multiplications operations required by the PFSR algorithm. With this strategy we are able to estimate the error vector on the measurements, which can be also used to identify those genes that may correspond to biological markers. The method used in [12] uses an ℓ_1-regularized least squares algorithm, with a regularization parameter that must be carefully chosen to balance the tradeoff between sparsity and reconstruction error.

We compare our numerical results using the Sparse Representation (SR) approach proposed in this work with the classification method of Support Vector Machines (SVM) in [18]. Support vector machines are a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. The original SVM algorithm was invented by Vladimir Vapnik and the current standard implementation was proposed by Corinna Cortes and Vladimir Vapnik [8]. In order to perform this comparison, we use the software GEMS-SVM which has implemented several multiclass SVMs including one-versus-rest (OVR), one-versus-one (OVO), and directed acyclic graph (DAG). Polynomial and radial basis functions (RBF) kernels can also be used. For SVM, we report the best numerical result obtained from the different combinations available.

In Table 2 we show the performance measure results for each of the datasets tested in this experimentation, where the error rate of the classifier is computed and compared.

Table 2. Classification results for SR and SVM

Dataset	SR	SVM
9_Tumors	66.67%	67.01%
11_Tumors	96.55%	94.99%
Prostate_Tumor	94.12%	93.27%
Lung_Cancer	95.07%	96.05%
SRBCT	100%	100%
Brain_Tumor	91.11%	90%

The Sparse Representation (SR) results reported in Table 2 are encouraging when compared with the SVM approach for classification problems. We can see that SR meets the best performance reached by SVMs method.

We also see the *low* rate of accuracy for the dataset 9_Tumors which is probably related with the number of samples available, since from a total of 60 samples only 2 are available for category 7 corresponding to the prostate tumor case. This contrasts with the 9 samples available for non-small cell lung cancer (NSCL); 8 samples for breast, renal and melanoma cases; 7 for colon and 6 for ovarian, leukemia, and central nervous system (CNS) cases.

Therefore, in the situation when the only two samples available for category 7 happen to be in the testing dataset, generated by the random 10-fold cross validation stage, we will not have any samples of this category for training, i.e., these samples do not have any natural sparse linear representation using those elements in the training dataset.

In Figure 2, the sparse representation for the last cross-validation experiment on the binary dataset Prostate_Tumor is presented. One can notice the contrast between the large and small coefficients of the solution vector **c**, suggesting that the given test sample belongs to exactly one of the two classes in this dataset. In this case, the last test sample corresponds to one of the prostate tumor samples which are represented in red color, while the normal tissue samples are shown in blue. This fact confirms the idea of expressing any test sample as a linear combination of only those training samples belonging to the same class.

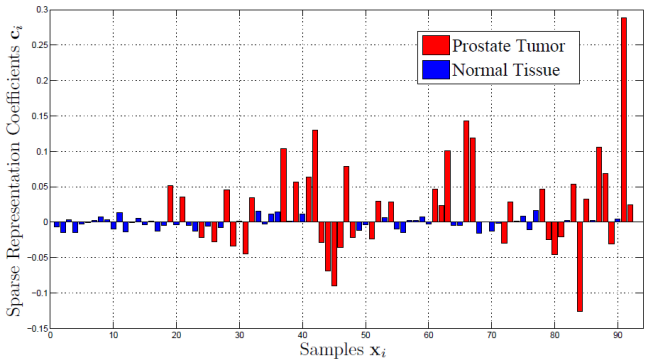

Fig. 2. The sparse representation of a given test sample.

IV. Conclusion

In this work, we have presented a new method of supervised machine learning for classification via ℓ_1-optimization. Through the ℓ_1-optimization it is possible to obtain a sparse representation of each testing sample as a linear combination of all the training samples, so the sparse representation approach does not contain separate training and testing stages. The advantage of the sparse representation technique based on ℓ_1-optimization is twofold: it is not necessary to care for model selection as in SVM, and on the other hand robustness with respect to outliers and noise is gained using the ℓ_1-norm. The numerical results obtained show that the performance of the proposed method is comparable with or better than that obtained from the GEMS-SVM software.

Since the treatment of cancer greatly depends on the accurate classification of tumors, the development of numerical techniques from data mining like the one presented in this work could be used in the process of classification using gene expression data, and has the potential to provide a more accurate and objective cancer diagnosis.

Acknowledgment

This work was supported in part by the U.S. Army Research Laboratory, through the Army High Performance Computing Research Center, Cooperative Agreement W911NF-07-0027, Universidad de Los Andes, Venezuela, and the Program in Computational Science at the University of Texas at El Paso.

References

[1] M. Argáez, C. Ramirez, R. Sanchez, "An ℓ_1-algorithm for underdetermined systems and applications", *IEEE Conference Proceedings of North American Fuzzy Information Processing Society*, 2011, submitted for publication.

[2] M. Argáez, "Solving overdetermined systems in ℓ_q quasi-norms". *Special SCAN '08 Issue of Reliable Computing*, 2010.

[3] R. Aster, B. Borchers, C. Thurber, *Parameter Estimation and Inverse Problems*, Elsevier Academic Press, New York, 2005.

[4] M. Asyali, M. Alci, "Reliability analysis of microarray data using fuzzy c-means and normal mixture modeling based classification methods". *BIOINFORMATICS*, 2005, 21, (5), pp. 644-649.

[5] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge, U.K. Cambridge University Press, 2004.

[6] E. Candès, "Compressive sampling", *Proceedings of the International Congress of Mathematicians*, Madrid, Spain, 2006.

[7] E. Candès, J. Romberg and T. Tao, "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information", *IEEE Transactions on Information Theory*, Vol. 52, pp. 489-509, 2006.

[8] C. Cortes and V. Vapnik, "Support Vector Networks", *Machine Learning*, vol. 20, No. 3, pp. 273-297, 1995.

[9] M. Debruyne, "An Outlier Map for Support Vector Machine Classification", *Annals of Applied Statistics,* vol. 3, No.4, pp. 1566–1580, 2009.

[10] D. Donoho, "Compressed sensing", *IEEE Transactions on Information Theory*, Vol. 52, No. 4, pp. 1289-1306, 2006.

[11] M. Elad, *Sparse and redundant representations*, Springer 2010.

[12] X. Hang, F. Wu, "Sparse Representation for Classification of Tumors using gene expression data", *Journal of Biomedicine and Biotechnology*, 2009.

[13] H. Kamber, *Datamining Concepts and Techniques*, Elsevier publications, 2005.

[14] S. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinvesky, "An interior-point method for large-scale ℓ_1-regularized least squares", *IEEE Selected Topics in Signal Processing*, Vol. 1, No. 4, pp. 606-617, 2007.

[15] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection", *Proceedings of International Joint Conference on AI*, pp 1137-1145, 1995.

[16] C. Miosso, R. Von-Borries, M. Argáez, L. Velázquez, C. Quintero, C. Potes, "Compressed sensing reconstruction with prior information using penalized reweighted normal equations", *IEEE Transactions on Signal Processing*, Vol. 52, No. 4, pp. 1289-1306, 2009.

[17] A. Statnikov, I. Tsamardinos, Y. Dosbayev, C. Aliferis, "GEMS: A system for automated cancer diagnosis and biomarker discovery from microarray gene expression data". *International Journal of Medical Informatics*, 2005, 74, pp. 491-503.

[18] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. Springer-Verlag, New York, 2000.

[19] J. Wright, Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, "Robust Face Recognition via Sparse Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 61, No. 2, pp 210-227, 2009.