# Cancer Classification by Correntropy-Based Sparse Compact Incremental Learning Machine

Mojtaba Nayyeri[1,*] and Hossein Sharifi Noghabi[1,2]

[1]Department of Computer Engineering, Ferdowsi University of Mashhad, Iran

[2]Center of Excellence on Soft Computing and Intelligent Information Processing.

## Abstract

Cancer prediction is of great importance and significance and it is crucial to provide researchers and scientists with novel, accurate and robust computational tools for this issue. Recent technologies such as Microarray and Next Generation Sequencing have paved the way for computational methods and techniques to play critical roles in this regard. Many important problems in cell biology require the dense nonlinear interactions between functional modules to be considered. The importance of computer simulation in understanding cellular processes is now widely accepted, and a variety of simulation algorithms useful for studying certain subsystems have been designed. In this article, a Sparse Compact Incremental Learning Machine (SCILM) is proposed for cancer classification problem on microarray gene expression data which take advantage of Correntropy cost that makes it robust against diverse noises and outliers. Moreover, since SCILM uses $l_1$-norm of the weights, it has sparseness which can be applied for gene selection purposes as well. Finally, due to compact structure, the proposed method is capable of performing classification tasks in all of the cases with only one neuron in its hidden layer. The experimental analysis is performed on 26 well known microarray datasets regarding diverse kinds of cancers and the results show that the proposed method not only achieved significantly high accuracy but also because of its sparseness, final connectivity weights determined the value and effectivity of each gene regarding the corresponding cancer.

## 1. Introduction

Most of human diseases are influenced by genes, and identifying genetic landscape and profile of diseases is an undisputable fact especially when it comes to diseases such as cancer [1]. In the quest for determination of genetic causes of diseases, new technologies such as Next Generation Sequencing [2, 3] or Microarray expression [4] which are high-throughput procedures have paved the way to quantitate and record thousands of genes expression levels simultaneously [5-7]. These new technologies provide computational oncologists with valuable information for cancer prediction and cancer classification [1, 8, 9]. Making the best use of these valuable information and extracting it from datasets requires advanced, accurate and robust computational techniques because these datasets most of the time follow “large-p-small-n” paradigm which means they have high number of observed genes but low number of samples [10]. Cancer classification has been studied comprehensively with diverse methods from weighted voting scheme [11] and Partial Least Square (PLS) [6] to Support Vector Machines (SVM) [12] and Extreme Learning Machines (ELM) [13]. In addition to these methods, Artificial Neural Networks (ANNs) [14], Probabilistic neural networks (PNNs) [15] and soft computing approaches (hybrid of evolutionary computation and machine learning) were also applied and developed for cancer diagnosis and cancer classification [10, 16]. One of the well-known types of ANNs are constructive networks whose optimum structures (number of nodes in the hidden layer) are determined automatically [17-19]. In these networks, number of nodes and connectivity weights are gradually increased from the lowest to the optimum value and they are categorized in two types: compact [17, 18, 20] and non-compact [19]. Input parameters of the newly added node in the non-compact type are specified randomly whereas in the compact one, they are adjusted via an optimization process.

Most of these methods are suffering from “curse of dimensionality” which is related to high dimensions of these datasets. Another aspect of cancer classification is related to feature selection (gene selection) methods in order to prevent over-fitting in the learning process [21]. Model *et al.* [22] applied several feature selection methods for DNA methylation based cancer classification. Another comparative study for feature selection was performed by Li *et al.* [23] for tissue classification based on gene expression. Cawley *et al.* [24] proposed a sparse logistic regression with Bayesian regularization for gene selection in cancer classification and Zhang *et al.* [25] used SVM with non-convex penalty for the same problem. Piao *et al.* [26] take advantage of ensemble an correlation-based gene selection method for gene expression data regarding cancer classification. Interested readers can refer to five good surveys of feature selection in [27-30] and [31] and the references therein. However, feature selection comes with certain prices such as addition of another layer of complexity to the model or information loss [27].

In this article, we propose Sparse Compact Incremental Learning Machine (SCILM) which prevents over-fitting without feature selection due to its compact structure. Further, because of Correntropy cost SCILM is robust against noises and outliers. In addition to these advantages, since SCILM takes advantage of $l_1$-norm of the weights, it is sparse and this sparseness determines the most effective connectivity weights corresponding to all features. Therefore, the final weights of the generated model by SCILM can be utilized for gene selection purposes as well.

SCILM is a learning method for datasets with low sample size and high dimensions. These characteristics are highly important and medical and pharmaceutical research because numbers of genes or drug compounds are significantly lower than number of features and attributes one can find for them. SCILM is proposed for such problems and microarray profiles for cancer classification have both these characteristics. The presented method prevents over-fitting without feature selection due to its compact structure and also because of Correntropy cost SCILM is robust against noises and outliers. Authors in [32], investigated robustness of Correntropy objective function. In addition to these advantages, since SCILM takes advantage of $l_1$-norm of the weights, it is sparse and this sparseness determines the most effective connectivity weights corresponding to all features. Therefore, the final weights of the generated model by SCILM can be utilized for gene selection purposes as well.

The rest of the paper is organized as follows: section 2 presents the proposed method, section 3 describes the results and final section concludes the paper.

## 2. Methods and Materials

This section presents a new constructive network with sparse input side connections. The network has a single hidden layer in which the hidden nodes are added one by one until the network reaches a certain predefined performance. After the new hidden node is added and trained, its parameters are fixed and do not changed during training the next nodes. Each newly added node is trained in two phases: a) Input parameters adjustment, b) output parameter adjustment. The input parameters of the newly added node are trained based on Correntropy objective function. The output connection is adjusted by MSE objective function. In the rest of this section some preliminaries are described followed by the description of the proposed algorithm.

### 2.1 *Dataset representation*

The dataset with $N$ distinct samples is denoted by

$$\chi = \{x_j, t_j\}_{j=1}^{N},\ x_j \in R^d, t_j \in R \qquad (1)$$

### 2.2 *Network structure*

Let f be a continuous mapping, $f_L$ be the output of the network with L hidden nodes. The network is represented as

$$f_L(x) = \sum_{i=1}^{i=L} \beta_i g_i(x) \qquad (2)$$

Where

$$g_i(x) = g(\langle w_i, x\rangle + b_i), w_i \in R^d, b_i \in R \qquad (3)$$

Where $\langle .,. \rangle$ is inner product between two elements. **In this paper g is considered as tangent hyperbolic function**. The network (with L hidden nodes) error vector is defined as

$$\zeta_L = T - F \qquad (4)$$

Where $T = [t_1, ..., t_N]$ and $F = [f_L(x_1), ..., f_L(x_N)]$. The activation vector for the *i*th hidden node is

$$H_i = [H_{i1},...,H_{iN}], i = 1,...,L \tag{5}$$

where $H_{ij} = g_i\left(x_j\right), j = 1,...,N; i = 1,...,L$ .

## 2.3 *Correntropy*

Let v and u be two random variables with $\zeta = u - v$ . The Correntropy is a similarity measure between two random variables and defined as

$$V\left(E\right) = E\left(k\left(\zeta\right)\right) \tag{6}$$

Where $E(.)$ denotes the expectation in probability theory and k(.) denotes a kernel function which satisfy Mercer condition. In this paper only the Gaussian kernel is used. Regard to this,

$$V(E) = E\left(\frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-\left(\|u-v\|^2\right)}{2\sigma^2}}\right) \tag{7}$$

## 2.4 *Proposed method*

This subsection proposes a new incremental constructive network with sparse hidden layer connections. The hidden nodes are added to the network and trained one by one. When the new node parameters are tuned, they are frozen and do not change during training the next nodes. Fig. 1 illustrates the mechanism of the proposed method.

Training of the new node performs in two stages:

***Stage 1: input side optimization***

In the previous work [33], Input parameters of the new node are trained based on Correntropy objective function as follows:

$$V(H_L) = \arg\max_{H_L} E\left(\frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-\left(\|\zeta_{L-1} - s_L H_L\|^2\right)}{2\sigma^2}}\right) = \arg\max_{H_L} E(k(\zeta_{L-1}, s_L H_L)) = E(<\Phi(\zeta_{L-1}), \Phi(s_L H_L)>) \tag{8}$$

Where $s_L$ is a real number which is obtained by trial and error and $\zeta_{L-1}$ is the residual error for the network with *L*-1 hidden nodes. Regarding Eq.(8), the new node $H_L$ has most similarity to the residual error (regard to kernel definition). It is important to note that when the new node vector equals to the residual error vector (most similarity between the new node and the residual error), the training error becomes zero. Thus the optimal condition is [33]

$$\begin{aligned}
&\zeta_{(L-1)1} = s_L H_{L1} \\
&\zeta_{(L-1)2} = s_L H_{L2} \\
&\vdots \\
&\zeta_{(L-1)N} = s_L H_{LN}
\end{aligned} \tag{9}$$

Similarly, it is known that $H_{Li} = g_L(x_i)$, $i = 1,...,N$ and $g$ is $\tanh(.)$. Since $g$ is bipolar and invertible the system (9) can be rewritten as [33]

$$\begin{aligned} \tanh^{-1}\left(\frac{\zeta_{(L-1)1}}{s_L}\right) &= X_1 W_L \\ \tanh^{-1}\left(\frac{\zeta_{(L-1)2}}{s_L}\right) &= X_2 W_L \\ &\vdots \\ \tanh^{-1}\left(\frac{\zeta_{(L-1)N}}{s_L}\right) &= X_N W_L \end{aligned} \qquad (10)$$

Where $W_L = [w_L\, b_L]^T_{(d+1)*1}$, $w_L = [w_{L1},...,w_{Ld}]_{(d*1)}$ , $X_i = [x_i\ 1]_{1*(d+1)}$ , $i = 1,...,N$ and $w_L, b_L$ are input connections (input weight and bias) of the $L$th hidden node and $x_i$ is the $i$*th* training sample. To obtain added simple representation, let:

$$P_{Li} = \tanh^{-1}\left(\frac{\zeta_{(L-1)i}}{s_L}\right),\ i = 1,...,N \qquad (11)$$

Thus, we can write

$$\begin{aligned} P_{L1} &= X_1 W_L \\ P_{L2} &= X_2 W_L \\ &\vdots \\ P_{LN} &= X_N W_L \end{aligned} \qquad (12)$$

Regard to this, as mentioned in [33], and since system of equations (9) and (12) are equivalent, the following equation will be solved instead of (8):

$$V(W_L) = \arg\max_{W_L} E\left(e^{\frac{-\left\|P_L - XW_L\right\|^2}{2\sigma^2}}\right) \qquad (13)$$

Where

$$X = \left[X_1^T \quad ,... \quad ,X_N^T\right]^T \qquad (14)$$

The expectation can be approximated from data points, the constant term can be removed, and thus the following optimization problem is obtained [33]:

$$W_L = \arg\max_{W_L} E\left( e^{\frac{-\left\|P_L - XW_L\right\|^2}{2\sigma^2}} \right) \approx \arg\max_{W_L} \sum_{i=1}^{N} \left( e^{\frac{-\left\|P_{Li} - X_i W_L\right\|^2}{2\sigma^2}} \right) \qquad (15)$$

As mentioned in [33], to avoid overfitting and achieve a better generalization performance, the regularization term should be added:

$$W_L = \arg\max_{W_L} \left( E(e^{\frac{-\left\|P_L - XW_L\right\|^2}{2\sigma^2}}) - \lambda \left\|W_L\right\|^2 \right) \qquad (16)$$

As mentioned in [33] and similar to [34], employing the half quadratic optimization problem, the local solution of (16) is obtained, using the following iterative process [33]:

$$\begin{cases} \alpha_i^{t+1} = -G\left(P_{Li} - X_i W_L{}^t\right) \\ W_L{}^{t+1} = \arg\max_{W_L{}^t} \left( \sum_{i=1}^{N} \left( \alpha_i^{t+1} \frac{\left\|P_{Li} - X_i W_L{}^t\right\|^2}{2\sigma^2} \right) - \lambda \left\|W_L{}^t\right\|^2 \right) \end{cases} \qquad (17)$$

Where $G(z) = \exp(-\frac{\left\|z\right\|^2}{2\sigma^2})$. It is obvious that auxiliary variables i.e., $\alpha_i^{t+1}$ help to reduce effect of data that are contaminated by noises. After some derivations (multiplying the optimization problem by constant term $\sigma^2$ and set $\lambda' = 2\lambda$ ) which are mentioned in [33], we obtain:

$$\begin{cases} \alpha_i^{t+1} = -G\left(P_{Li} - X_i W_L{}^t\right) \\ \begin{cases} \max_{W_L} \sum_{i=1}^{N} \frac{\left(\alpha_i^{t+1} \xi_i^{2}\right)}{2} - \frac{\lambda'}{2} \left\|W_L{}^t\right\|^2 \\ X_i W_L{}^t = P_{Li} - \xi_i \ , \ i = 1, \ldots, N \end{cases} \end{cases} \qquad (18)$$

Several literatures in machine learning and adaptive filters replaced l2 norm by l1 norm to provide sparse solution (optimum weight) [35, 36]. Accordingly, inspired by [36] and Different from previous work [33], to provide a sparse solution, the following iterative process should be performed instead of (18):

$$\begin{cases} \alpha_i^{t+1} = -G\left(P_{Li} - X_i W_L{}^t\right) \\ \begin{cases} \max_{W_L{}^t} \sum_{i=1}^{N} \left(\alpha_i^{t+1} \xi_i \ \right) - \lambda'' \left|W_L{}^t\right| \\ X_i W_L{}^t = P_{Li} - \xi_i \ , \ i = 1, \ldots, N \end{cases} \end{cases} \qquad (19)$$

Where $|.|$ denotes the $l_1$ norm. Consider the following optimization problem that is extracted from Eq. (19):

$$\begin{cases} \max_{W_L} \sum_{i=1}^{N} \left(\alpha_i^{t+1} \xi_i \right) - \lambda'' \left|W_L^{\ t}\right| \\ X_i W_L^{\ t} = P_{Li} - \xi_i \ , \ i = 1, \dots, N \end{cases} \qquad (20)$$

Let $r = [r_1, \dots, r_{d+1}]$, $s = [s_1, \dots, s_{d+1}]$, $p = [p_1, \dots, p_N]$ and $q = [q_1, \dots, q_N]$, where $r, s \geq 0$ and $p, q \geq 0$.

Let $W_L^{\ t} = r - s$ and $\xi_i = p_i - q_i, i = 1, \dots, N$ , inspired by [36], the optimization problem (20) can be rewritten as a linear programming problem:

$$\begin{cases} \max_{W_L} \sum_{i=1}^{N} \left(\alpha_i^{t+1} \left(p_i + q_i\right)\right) - \lambda'' \left(r + s\right) \\ X_i \left(r - s\right) = P_{Li} - \left(p_i - q_i\right), \ i = 1, \dots, N \end{cases} \qquad (21)$$

In order to solve the problem (21) by using Matlab toolbox, it needs to change optimization problem (21) to the standard form. The standard form of linear programming which is used in Matlab is:

$$\begin{cases} \min_x \ f^T x \\ s.t : A_{eq} x = B_{eq} \\ lb \leq x \leq ub \end{cases} \qquad (22)$$

Thus, passing from (21) to (22), we obtain:

$$f = [\alpha \quad \alpha \quad \Lambda \quad \Lambda]^T_{(2*(N+(d+1)))*1}$$

Where

$$\alpha = [\alpha_1^{t+1}, \dots, \alpha_N^{t+1}]$$

$$\Lambda = \left[-\lambda'' - \lambda'' \ \dots - \lambda''\right]_{1*(d+1)}$$

$$x = [p \ q \ r \ s]^T_{(2*(N+(d+1)))*1}$$

And

$$A_{eq} = \left[\Gamma_{(N*N)} \quad -\Gamma_{(N*N)} \quad X_{(N*(d+1))} \quad -X_{(N*(d+1))}\right]_{N*(2*(N+(d+1))}$$

where

$$\Gamma = \begin{bmatrix} 1 & \dots & 1 \\ 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}_{(N*N)}$$

$$B_{eq} = \left[ P_{L1}, \dots, P_{LN} \right]_{N*1}^{T}$$

$$lb = 0 \text{ and } ub = \infty$$

Thus, the optimum input parameters of the new node are adjusted using the following iterative process:

$$\begin{cases} \alpha_i^{t+1} = -G\left( P_{Li} - X_i W_L^{\ t} \right) \\ \begin{cases} \min_x \ f^T x \\ s.t: A_{eq} x = B_{eq} \\ lb \le x \le ub \end{cases} \end{cases} \qquad (23)$$

***Stage 2: Output side optimization:*** Similar to [17], the output weight is adjusted using the following equation

$$\beta_L = \frac{\left\langle \zeta_{(L-1)}, H_L \right\rangle}{\left\langle H_L, H_L \right\rangle} \qquad (24)$$

Where $H_L$ is obtained from previous stage.

The proposed method is specified in Algorithm 1 and Fig. 2 is the flow chart of SCILM:

**Algorithm 1** SCILM

```
Input: training samples $\chi = \{x_i, t_i\}_{i=1}^{N}$
Output: The optimal input and output weights $\beta_i, W_i, i = 1, \dots, L$
Initialization: Maximum number of hidden nodes $L$, regularization term $\lambda''$, Maximum number of iterations IT1
For i=1:L
    Stage 1: calculate $P_L$ and $X$ by (11) and (14)
    For k=1:IT1
            Update input parameters according to (23)
    END
  Stage 2: calculate the hidden node vector $\mathrm{H_L}$ using (5),
and the error vector, $\zeta_{L-1}$ , using (4):
          Adjust the output weight according to (24)
    END
    Update Error as $\zeta_L = \zeta_{L-1} - \beta_L H_L$
END
```

## 2.5 *Datasets*

The experiments were performed on 26 datasets adopted from [21] which contain 11 multi-class and 15 two-class datasets. We applied the same reference numbering system as [21] for convenience. Reference numbers less than or equal to 15 are for two-class datasets and reference numbers greater than 15 indicate multi-class datasets. Most of the datasets have dimensions in the range of approximately 2000 to 25000 (except dataset 19 about yeast which has 79 features) and the sample size varies approximately from 50 to 300. The detail for each dataset is as follow:

Six of the datasets have been studied in [37] and among them, datasets numbered 1 to 3 relate to breast cancer, 4 and 5 deal with lung cancer and dataset 6 is about hepatocellular carcinoma. Another group of six datasets studied in [38]. Datasets 7 and 8 deal with prostate cancer, 9 and 10 are about breast cancer and finally 16 and 17 are related to leukaemia. Five well-known bioinformatics datasets are about colon cancer (11) [39], ovarian cancer (12) [40], leukaemia (13) [11], lymphoma (18) [41] and yeast (19) [42]. There rest of the datasets are selected from NCBI GEO and their corresponding IDs are available in [21] and in this paper they are numbered in the range of {14,15} U {20,...,26}.

## *2.6 Experiments*

This paper used SVMs with Radial Basis Function (RBF) and sigmoid kernels and Correntropy based ELM (ELM-RCC) [34] to compare with SCILM. The RBF kernel used in SVM is defined as $\mathrm{K(u,v)} = \exp(-\gamma\|\mathrm{u}-\mathrm{v}\|^2)$ and the sigmoid kernel is defined as $K(u,v) = \tanh(\alpha.u.v + \beta)$ $\mathrm{K(u,v)} = \tanh(\alpha\mathrm{uv} + \beta)$. For SVM, we used the LIBSVM toolbox [43], the regularization parameter $\mathrm{C}$ is selected from the set $\{10^{-3}, 10^{-2}, \dots, 10^{5}\}$ and the RBF kernel parameter $\gamma$ is selected from$\{\{10^{-2}, \dots, 10^{2}\}\}$. For the SVM with sigmoid kernel, the parameters $\alpha$ and $\beta$ are selected from the sets $\{\{10^{-2}, \dots, 10^{2}\}\}$ and $\{10^{-1}, \dots, 10^{1}\}$ respectively. ELM-RCC [34] have a hyper parameter $\lambda$and the best of this parameter is selected from the set$\{\{10^{-3}, 10^{-2}, \dots, 10^{5}\}\}$. Similar to [44], the number of hidden units in ELM-RCC is set to 1000. This paper used additive nodes with sine activation function for ELM-RCC as $h(x) = \sin(\langle w, x\rangle + b)$ $\mathrm{h(x)} = \sin(< w, x > +b)$. The parameters $\mathrm{w, b}$ are randomly generated by uniform distribution between -1 and 1. Furthermore, the data samples are normalized into range -1 and 1. For the proposed method, the parameter $\lambda''$ in the optimization problem (20) is selected from the set $\{10^{-3}, 10^{-2}, \dots, 10^{5}\}$ and the number of hidden nodes is set to one (the most compact network i.e., one hidden layer network with one hidden node) and the kernel width $(\sigma)$ is selected from$\{10^{-2}, \dots, 10^{2}\}$. SVM used one against all strategy for multiclass classification datasets. To evaluate the performance of the proposed method in comparison with SVM and ELM-RCC, the testing accuracies are reported in table 1 and table 2. For the proposed method experiments are performed in 20 independent trials for each problem. In each trial, data samples are reshuffled and mean of accuracy is reported in the following tables.

## 3. Result and Discussion

In this section, we discuss the results from two viewpoints of accuracy and feature selection. In the accuracy part in comparison with the stated methods, SCILM achieved significantly better results in 10 datasets, performed equally in 6 datasets and only lost in 7 datasets. In table 1, the average accuracy of each two-class dataset over 20 independent runs is reported. According to this table, in the case of dataset 1 SCILM achieved the accuracy of 88.75% and among the compared methods ELM-RCC achieved 76.875%. For dataset 2, SCILM has an accuracy of

71.58% while SVM with RBF kernel has 66.94%. In the third datasets we achieved 71.33% but both SVMs were not able to perform better than 64.21%. In dataset 4, SCILM and SVM with RBF had accuracy levels of 61.27% and 58.09%, respectively. For the fifth dataset, SVM with sigmoid kernel and SCILM performed equally, however, in dataset 6 the same SVM performs better than the proposed method. Both SVMs achieved almost the same result as SCILM for dataset 7 in spite of the fact that SCILM was slightly better. In dataset 8, SCILM achieved 97.85% and ELM-RCC was not better than 94.28%. In dataset 9, SCILM shows the performance of 87.5% while SVM with RBF had the accuracy of 80%. In datasets 10, 12, 13 and 15 the proposed methods and the best of the compared methods achieved almost the same accuracy level; however, in datasets 11 and 14 SVM with RBF kernel performs better than the proposed method.

The results of the average test accuracies for multi-class datasets are reported in table 2. According to this table, for datasets 16 and 17 the proposed method achieved significantly better results than the best of the other compared methods, however, in datasets 18 and 19 SVM with sigmoid kernel achieved slightly better results than SCILM and for datasets 20 and 21, ELM-RCC achieved higher accuracy than SCILM. For the rest of the datasets, SCILM obtained competitive results and outperforms other methods especially in the case of datasets 22, 23 and 25.

It is important to note than SCILM has only 1 neuron in its hidden layer for all datasets, while ELM-RCC has 1000 neurons in its hidden layer and SVMs also take advantage of cross validation for parameter optimization based on LIBSVM library.

Concerning the aspect of feature selection, because SCILM is taking advantage of the $l_1$-norm, it has sparseness and the final connectivity weights also tend to be sparse. Therefore, after generating the model for each dataset these final weights can be analyzed from the feature selection point of view. According to this approach, each feature has a corresponding weight which indicates the value of that feature i.e. more valuable features have higher values for their corresponding connectivity weights and less vital features have approximately equal to zero values for their final weights. Due to sparsity of the generated model, most of these weights tend to be near or equal to zero. The values of these weights for 8 randomly selected datasets are illustrated in figure 3. In order to save space the rest of the datasets are not considered in this figure. As shown in this figure, the weights are clearly specifying valuable features and separate them from less informative ones. The advantage of this feature selection approach is that it does not have additional computational cost or complexity for the model since it is within the learning process.

As for another advantage of SCILM, since the auxiliary variables ( $\alpha$ ) that appeared in optimization problem (20), have a small value for the outliers and these data points have a small role in the optimization of the network parameters, the proposed method is also robust to outliers as well.
The proposed network has an advantage of having a more compact architecture i.e., only has one hidden node in single layer, while the experiments demonstrated that SVM and ELM have several nodes (from 15 to 200 support vectors for SVM and 1000 hidden nodes for ELM). Furthermore, SCILM has the most sparse input side connection (in most cases sparsity rate is up to 99%). This simpler structure and the lowest number of degrees of freedom (parameters) lead to a substantially better generalization performance compared with the other methods.

## 4. Conclusion

In this paper, we have proposed a new classification method named SCILM based on incremental learning machines for cancer classification based on gene expression microarray data, which has two main significant advantages. First, because of Correntropy cost function, it is robust against uncertainty and noisy data. Second, since it uses the $l_1$-norm of the weights in its optimization process, it can be seen as a feature selection method as well. This norm provides the generated model with sparseness in the weights, which can be exploited for feature selection purposes.

In the proposed method, the network structure is determined automatically, leading to a better generalization performance. Furthermore, due to optimization of the hidden layer parameters, the network has a compact architecture and fast training and testing speed.

We demonstrated that SCILM significantly improved the performance of other compared methods in both two-class and multi-class datasets. The capability of the SCILM for feature selection and selecting meaningful genes still requires more experiments and studies which is in center of our future research.

## References

[1] R. Upstill-Goddard, D. Eccles, J. Fliege, A. Collins, Machine learning approaches for the discovery of gene–gene interactions in disease data, Briefings in Bioinformatics, (2012).

[2] O. Morozova, M.A. Marra, Applications of next-generation sequencing technologies in functional genomics, Genomics, 92 (2008) 255-264.

[3] I. Gobernado, A. Sanchez-Herranz, A. Jimenez-Escrig, Chapter 3 - Next-Generation Sequencing: New Tools to Solve Old Challenges, in: A.C. Carolina Simó, G.-C. Virginia (Eds.) Comprehensive Analytical Chemistry, Elsevier, 2014, pp. 47-79.

[4] T. Muangsub, J. Samsuwan, P. Tongyoo, N. Kitkumthorn, A. Mutirangura, Analysis of methylation microarray for tissue specific detection, Gene, 553 (2014) 31-41.

[5] A. Taylor, J. Steinberg, T.S. Andrews, C. Webber, GeneNet Toolbox for MATLAB: a flexible platform for the analysis of gene connectivity in biological networks, Bioinformatics (Oxford, England), 31 (2015) 442-444.

[6] D.V. Nguyen, D.M. Rocke, Multi-class cancer classification via partial least squares with gene expression profiles, Bioinformatics (Oxford, England), 18 (2002) 1216-1226.

[7] L. Dailey, High throughput technologies for the functional discovery of mammalian enhancers: New approaches for understanding transcriptional regulatory network dynamics, Genomics.

[8] P. Larranaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J.A. Lozano, R. Armananzas, G. Santafe, A. Perez, V. Robles, Machine learning in bioinformatics, Brief Bioinform, 7 (2006) 86-112.

[9] G.B. Fogel, Computational intelligence approaches for pattern discovery in biological systems, Brief Bioinform, 9 (2008) 307-316.

[10] R.M. Luque-Baena, D. Urda, J.L. Subirats, L. Franco, J.M. Jerez, Application of genetic algorithms and constructive neural networks for the analysis of microarray cancer data, Theoretical biology & medical modelling, 11 Suppl 1 (2014) S7.

[11] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science (New York, N.Y.), 286 (1999) 531-537.

[12] T.S. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, D. Haussler, Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics (Oxford, England), 16 (2000) 906-914.

[13] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing, 70 (2006) 489-501.

[14] L.J. Lancashire, C. Lemetre, G.R. Ball, An introduction to artificial neural networks in bioinformatics-application to complex microarray and mass spectrometry datasets in cancer studies, Brief Bioinform, 10 (2009) 315-329.

[15] A. Statnikov, C.F. Aliferis, I. Tsamardinos, D. Hardin, S. Levy, A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis, Bioinformatics (Oxford, England), 21 (2005) 631-643.

[16] J.J. Liu, G. Cutler, W. Li, Z. Pan, S. Peng, T. Hoey, L. Chen, X.B. Ling, Multiclass cancer classification and biomarker discovery using GA-based algorithms, Bioinformatics, 21 (2005) 2691-2697.

[17] T.-Y. Kwok, D.-Y. Yeung, Objective functions for training new hidden units in constructive neural networks, Neural Networks, IEEE Transactions on, 8 (1997) 1131-1148.

[18] S.E. Fahlman, C. Lebiere, The cascade-correlation learning architecture, (1989).

[19] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, Neural Networks, IEEE Transactions on, 17 (2006) 879-892.

[20] G. Huang, S. Song, C. Wu, Orthogonal least squares algorithm for training cascade neural networks, Circuits and Systems I: Regular Papers, IEEE Transactions on, 59 (2012) 2629-2637.

[21] L. Song, J. Bedo, K.M. Borgwardt, A. Gretton, A. Smola, Gene selection via the BAHSIC family of algorithms, Bioinformatics (Oxford, England), 23 (2007) i490-498.

[22] F. Model, P. Adorján, A. Olek, C. Piepenbrock, Feature selection for DNA methylation based cancer classification, Bioinformatics (Oxford, England), 17 (2001) S157-S164.

[23] T. Li, C. Zhang, M. Ogihara, A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression, Bioinformatics (Oxford, England), 20 (2004) 2429-2437.

[24] G.C. Cawley, N.L.C. Talbot, Gene selection in cancer classification using sparse logistic regression with Bayesian regularization, Bioinformatics (Oxford, England), 22 (2006) 2348-2355.

[25] H.H. Zhang, J. Ahn, X. Lin, C. Park, Gene selection using support vector machines with non-convex penalty, Bioinformatics (Oxford, England), 22 (2006) 88-95.
[26] Y. Piao, M. Piao, K. Park, K.H. Ryu, An ensemble correlation-based gene selection algorithm for cancer classification with gene expression data, Bioinformatics (Oxford, England), 28 (2012) 3306-3315.
[27] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics (Oxford, England), 23 (2007) 2507-2517.
[28] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, A. Nowe, A survey on filter techniques for feature selection in gene expression microarray analysis, IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM, 9 (2012) 1106-1119.
[29] E. Hemphill, J. Lindsay, C. Lee, Mandoiu, II, C.E. Nelson, Feature selection and classifier performance on diverse bio- logical datasets, BMC bioinformatics, 15 Suppl 13 (2014) S4.
[30] S. Ma, J. Huang, Penalized feature selection and classification in bioinformatics, Brief Bioinform, 9 (2008) 392-403.
[31] B. Duval, J.-K. Hao, Advances in metaheuristics for gene selection and classification of microarray data, Briefings in Bioinformatics, 11 (2010) 127-141.
[32] H. Sharifi Noghabi, M. Mohammadi, Robust Group Fused Lasso for Multisample CNV Detection under Uncertainty, bioRxiv, (2015).
[33] M. Nayyeri, M. Rohani, H. Sadoghi Yazdi, A new Correntropy Based Constrctive Neural network, Neral Networks and Learning Systems, IEEE Transactions on, Submitted, (2015).
[34] H.-J. Xing, X.-M. Wang, Training extreme learning machine via regularized correntropy criterion, Neural Computing and Applications, 23 (2013) 1977-1986.
[35] L. Zhang, W. Zhou, On the sparseness of 1-norm support vector machines, Neural Networks, 23 (2010) 373-385.
[36] O.L. Mangasarian, Exact 1-norm support vector machines via unconstrained convex differentiable minimization, The Journal of Machine Learning Research, 7 (2006) 1517-1530.
[37] L. Ein-Dor, O. Zuk, E. Domany, Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer, Proceedings of the National Academy of Sciences of the United States of America, 103 (2006) 5923-5928.
[38] P. Warnat, R. Eils, B. Brors, Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes, BMC bioinformatics, 6 (2005) 265.
[39] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A.J. Levine, 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 (1999) 6745-6750.
[40] A. Berchuck, E.S. Iversen, J.M. Lancaster, J. Pittman, J. Luo, P. Lee, S. Murphy, H.K. Dressman, P.G. Febbo, M. West, J.R. Nevins, J.R. Marks, Patterns of gene expression that characterize long-term survival in advanced stage serous ovarian cancers, Clinical cancer research : an official journal of the American Association for Cancer Research, 11 (2005) 3686-3696.
[41] A.A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma, I.S. Lossos, A. Rosenwald, J.C. Boldrick, H. Sabet, T. Tran, X. Yu, J.I. Powell, L. Yang, G.E. Marti, T. Moore, J. Hudson, Jr., L. Lu, D.B. Lewis, R. Tibshirani, G. Sherlock, W.C. Chan, T.C. Greiner, D.D. Weisenburger, J.O. Armitage, R. Warnke, R. Levy, W. Wilson, M.R. Grever, J.C. Byrd, D. Botstein, P.O. Brown, L.M. Staudt, Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, Nature, 403 (2000) 503-511.
[42] M.P. Brown, W.N. Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, M. Ares, Jr., D. Haussler, Knowledge-based analysis of microarray gene expression data by using support vector machines, Proceedings of the National Academy of Sciences of the United States of America, 97 (2000) 262-267.
[43] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol., 2 (2011) 1-27.
[44] H. Guang-Bin, Z. Hongming, D. Xiaojian, Z. Rui, Extreme Learning Machine for Regression and Multiclass Classification, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 42 (2012) 513-529.

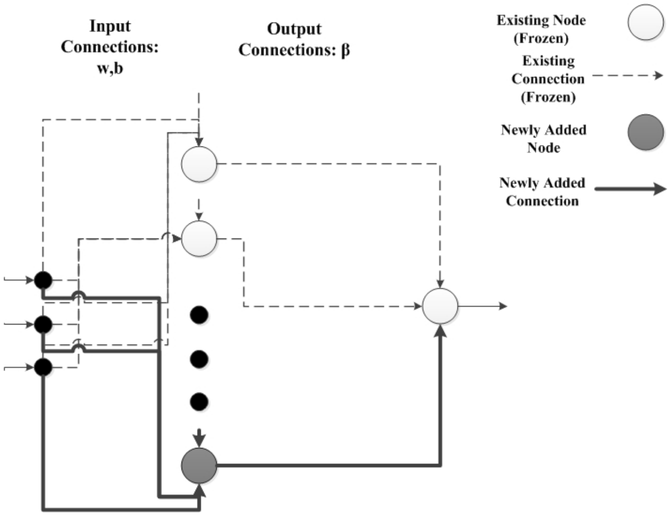

**Fig 1. Each newly added node is trained in two stages: a) Input weights adjustment b) Output weight adjustment. Existing nodes and connections do not change during training the new node.** (R1.1), (R1.2), (R2.2)

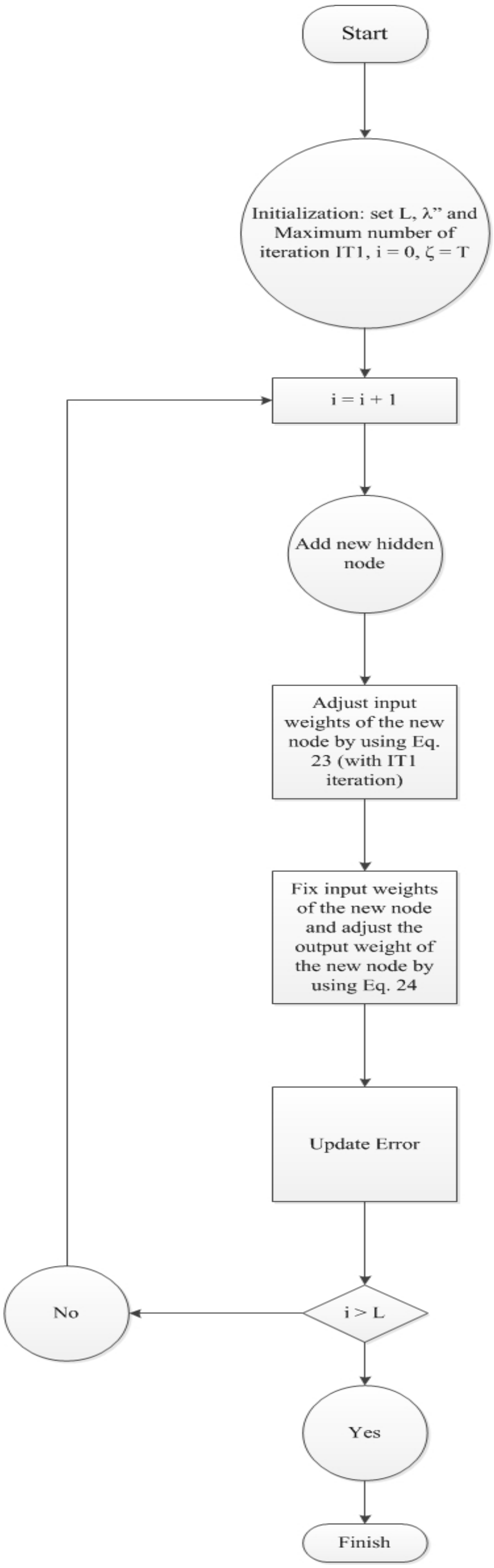
Start
Initialization: set L, λ'' and Maximum number of iteration IT1, i = 0, ζ = T
i = i + 1
Add new hidden node
Adjust input weights of the new node by using Eq. 23 (with IT1 iteration)
Fix input weights of the new node and adjust the output weight of the new node by using Eq. 24
Update Error
i > L
No
Yes
Finish

Fig 2. Flow chart of the algorithm. (R1.2)

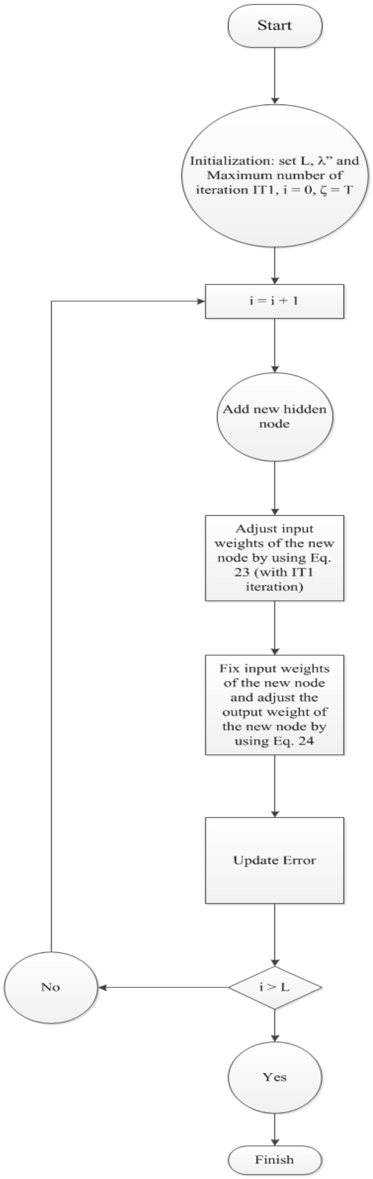
Start
Initialization: set L, λ" and Maximum number of iteration IT1, i = 0, ζ = T
i = i + 1
Add new hidden node
Adjust input weights of the new node by using Eq. 23 (with IT1 iteration)
Fix input weights of the new node and adjust the output weight of the new node by using Eq. 24
Update Error
i > L
No
Yes
Finish

**Fig 1. Flow chart of the proposed algorithm.** (R1.2)

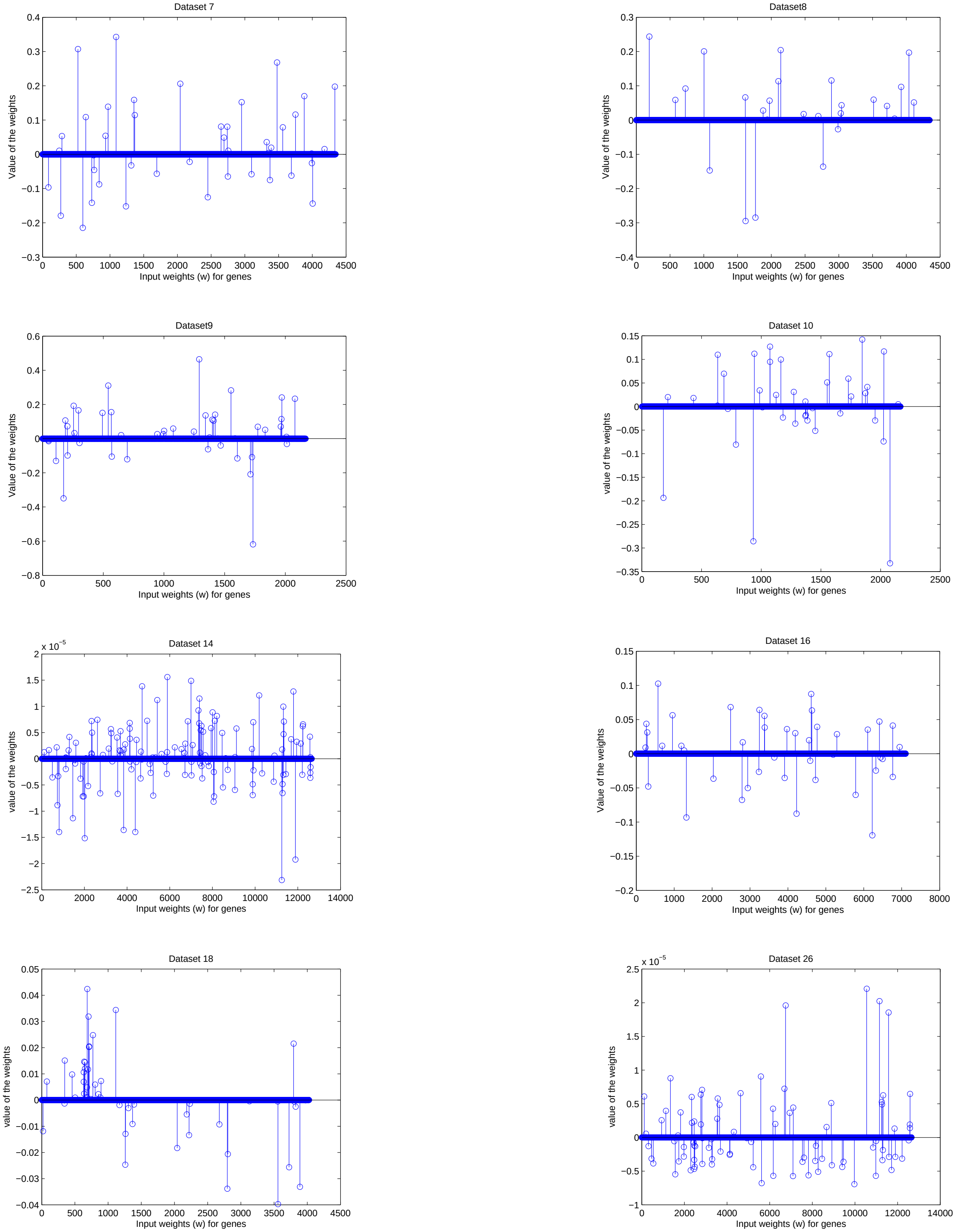

Fig 3. Connectivity weights (w) of the hidden layer are sparse and as these weights are coefficients of the genes, hardly informative genes are eliminated and only effective genes are considered for the model.