01 Jan 2006

Gene Expression Data for DLBCL Cancer Survival Prediction with a Combination of Machine Learning Technologies

Rui Xu
Missouri University of Science and Technology

Xindi Cai

Donald C. Wunsch
Missouri University of Science and Technology, dwunsch@mst.edu

Recommended Citation

R. Xu et al., "Gene Expression Data for DLBCL Cancer Survival Prediction with a Combination of Machine Learning Technologies," *Proceedings of the IEEE Engineering in Medicine and Biology 27th Annual Conference, 2005*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2006.
The definitive version is available at https://doi.org/10.1109/IEMBS.2005.1616559

Proceedings of the 2005 IEEE
Engineering in Medicine and Biology 27th Annual Conference
Shanghai, China, September 1-4, 2005

Gene Expression Data for DLBCL Cancer Survival Prediction with A Combination of Machine Learning Technologies

Rui Xu, Xindi Cai, and Donald C. Wunsch II
Applied Computational Intelligence Laboratory, Dept. of Electrical and Computer Engineering
University of Missouri – Rolla, Rolla, MO 65409-0249 USA

Abstract—**Gene expression profiles have become an important and promising way for cancer prognosis and treatment. In addition to their application in cancer class prediction and discovery, gene expression data can be used for the prediction of patient survival. Here, we use particle swarm optimization (PSO) to address one of the major challenges in gene expression data analysis, the curse of dimensionality, in order to discriminate high risk patients from low risk patients. A discrete binary version of PSO is used for gene selection and dimensionality reduction, and a probabilistic neural network (PNN) is implemented as the classifier. The experimental results on the diffuse large B-cell lymphoma data set demonstrate the effectiveness of PSO/PNN system in survival prediction.**

Keywords—**Gene expression data, Probabilistic neural networks, Particle swarm optimization, Cancer survival prediction.**

I. Introduction

Gene expression profiles have become an important and promising way for cancer prognosis and treatment. Particularly, their applications for cancer class prediction and discovery have already attracted numerous efforts from a wide variety of research communities [1-4]. On the other hand, prediction of patient survival, based on gene expression profiles, is also useful in choosing appropriate therapy [3-11]. This link provides an effective means to overcome the insufficiency of traditional methods in cancer research, which are largely dependent on the morphological appearance of tumors or the parameters derived from clinical observations. Several studies have been done for patient survival analysis based on either hierarchical clustering or statistical regression [3-11]. For example, Garber et al. used hierarchical clustering to divide adenocarcinoma tumors into three groups, and significant differences in patient survival rates is observed in these groups [4]. Bair and Tibshirani calculated the Cox score to find genes whose expression levels are correlated with patient survival and only performed clustering on the selected important high-scored genes [8]. However, machine learning and neural network technologies, which have achieved many appealing results in microarray data analysis, are also worth exploring for this problem.

Here, we propose a hybrid system, consisting of two important technologies in machine learning and neural networks, i.e., particle swarm optimization (PSO) and probabilistic neural networks (PNN), to perform survival analysis on a diffuse large B-cell lymphoma (DLBCL) data set [5]. PSO is an evolutionary computation technique for global optimization, which is based on the simulation of complex social behavior [12]. We use PSO to address one of the major challenges of microarray data analysis: the overwhelming number of measures of gene expression levels compared with the small number of samples. This is known as the curse of dimensionality [15]. Not all of these genes (features) are relevant to a specific tumor type, and the inclusion of the unrelated genes in the data analysis not only increases the computational complexity, but makes the results hard to interpret and prevents determining the appropriate therapy. Therefore, gene selection is critically important. By using PSO in gene selection, we consider the ability of PSO to balance global and local exploration, its effectiveness in achieving high-quality solutions, and the memory mechanism for keeping track of previous best solutions and therefore, avoiding the possible loss of previously learned knowledge. PNN was introduced as an implementation of nonparametric Pazen window estimation with feed-forward neural network architecture [14]. PNN has the advantage of fast training, only one user dependent parameters, and the capability to approximate arbitrarily complex decision boundaries, and has already shown appealing performance in cancer identification [2].

The paper is organized as follows. Section II describes the PSO/PNN system for survival analysis. The experimental results on DLBCL data set are presented and discussed in section III. Section IV concludes the paper.

II. Methods

PSO is motivated by the behavior of bird flocking or fish schooling and originally intended to explore optimal or near-optimal solutions in sophisticated continuous spaces [12]. A randomized velocity is associated with each potential solution, called a particle in the swarm. These particles change their positions in the search space until a stop condition is satisfied. The basic idea of PSO is to accelerate each particle towards two best locations at each time step, where one is the previous best solution for the particle, based on the calculated fitness value, and the other

0-7803-8740-6/05/$20.00 ©2005 IEEE.

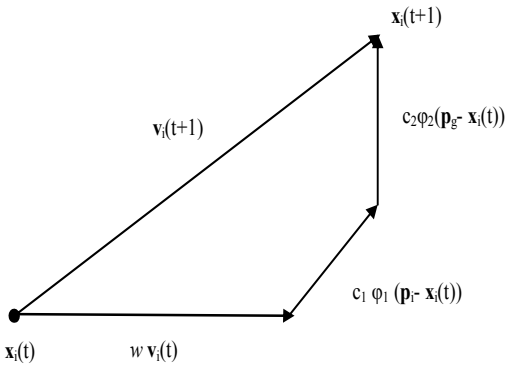

Fig. 1. . Concept of a swarm particle's position. $\mathbf{x}_i(t)$ and $\mathbf{v}_i(t)$ denote the particle's position and the associated velocity vector in the searching space at generation t, respectively. Vector $c_1\,\varphi_1\,(\mathbf{p}_i - \mathbf{x}_i(t))$ and $c_2\,\varphi_2\,(\mathbf{p}_g - \mathbf{x}_i(t))$ describe the particle's "cognitive" and "social" activities, respectively. The new velocity $\mathbf{v}_i(t+1)$ is determined by the momentum part, "cognitive" part, and "social" part. The particle's position at generation t+1 is updated with $\mathbf{x}_i(t)$ and $\mathbf{v}_i(t+1)$.

is the best overall value in the whole swarm. The basic concept of PSO is depicted in Fig. 1.

Since our goal is to select a subset of important genes (features) from a large gene pool and therefore reduce the dimensionality, we use a discrete binary version of PSO [13]. The major change of the binary PSO comes from the re-explanation of the meaning of the particle velocity. Given a set of particles $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)$, where N is the number of particles in the swarm, the velocity for the i^{th} particle $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{iD})$, where D is the number of dimensions in a particle, is represented as $\mathbf{v}_i = (v_{i1}, v_{i2}, ..., v_{iD})$. The possible values for each bit x_{id} $(1 \le i \le N,\ 1 \le d \le D)$ is either one or zero, indicating whether the corresponding genes are selected or not. Its velocity v_{id} is explained as the probability that x_{id} takes the value of one, and is squashed into the interval [0,1] through a logistic function $S(v_{id}) = 1/(1+\exp(-v_{id}))$. The basic procedure of binary PSO for gene selection is as follows:

i). Initialize a population of N particles with random positions and velocities. The dimensionality D of the problem space is dependent on the number of genes in the data set.

ii). Evaluate the classification performance of the classifier and calculate the optimization fitness function for each particle. Here, the design of fitness function aims to minimize the classification error and also favor the subset with fewer genes, which is defined as

$$f(\mathbf{x}_i) = Acc_{LOOCV} + 1/n, \quad (1)$$

where $Acc_{LOOCV} = \dfrac{\text{number of correctly classified patients}}{\text{total number of patients}} \times 100\%$ is the leave one out cross validation (LOOCV) [15] classification accuracy and n is the number of genes selected.

iii). Compare the fitness value of the i^{th} particle with its previous best position $\boldsymbol{p}_i$. If the current value is better than $\boldsymbol{p}_i$, reset both $\boldsymbol{p}_i$ and location to the current value and location.

iv). Compare the fitness value of each particle with the global best position $\boldsymbol{p}_g$. If current value is better than $\boldsymbol{p}_g$, reset $\boldsymbol{p}_g$ to the current particle's array index and value.

v). Update the velocity and position of the particle with the following equations.

$$v_{id} = w \times v_{id} + c_1 \times \varphi_1 \times (p_{id} - x_{id}) + c_2 \times \varphi_2 \times (p_{gd} - x_{id}) \quad (2)$$

$$x_{id} = \begin{cases} 1 & \text{if } \varphi_3 + \delta < S(v_{id}) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where x_{id} and v_{id} are the position and velocity of the d^{th} dimensionality of the i^{th} particle, respectively, w is the inertia weight, c_1 and c_2 are the acceleration constants, φ_1, φ_2, and φ_3 are uniform random functions in the range of [0, 1], δ is a parameter that limits the total number of genes selected to some certain range, and $S()$ is the sigmoid function. Compared with the original binary PSO in [13], we add the parameter δ in order to control the number of selected genes more flexibly.

vi). Return to step ii until the stop condition is satisfied, usually a maximum number of iterations or high-quality solutions.

The velocity update of a PSO particle in (2) comprises three parts. The first is the momentum part, which prevents abrupt velocity change. The second is the "cognitive" part, which represents learning achieved from its own search experience. The third is the "social" part which represents the cooperation among particles – learning from the group best's search experience. The inertia weight w controls the balance of global and local search ability. A large w facilitates the global search while a small w enhances local search. The velocity for each particle is restricted to a limit V_{max}. During the evolutionary procedure, the velocity is re-assigned to V_{max} if it exceeds V_{max}. For binary PSO, this limits the probability that a bit in a particle takes on the value of one. Usually, the smaller V_{max} is, the higher the mutation rate [13].

A typical PNN architecture is illustrated in Fig. 2, which consists of three layers: input layer, pattern layer and category layer [14-15]. The input layer works as a distribution mechanism and receives input components from the data set. Therefore, the number of nodes in this layer is equal to the dimension of the input vector. All of these nodes are fully connected with the nodes in the pattern layer, which is considered as the key of PNN. PNN requires m pattern nodes if the total number of training patterns is m, so that each pattern node can correspond to a training pattern. In contrast to the link between input and pattern

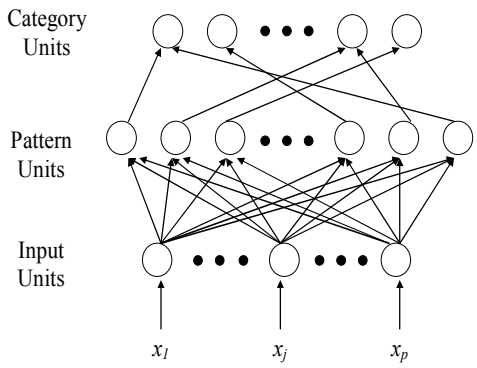

Fig. 2. PNN architecture. Each pattern node represents a pattern in the training set. The Bayesian posterior probability for each category is obtained as the output of the corresponding category node.

layers, the nodes of pattern and category layers are sparsely connected. Each pattern node is only connected to the category node that correctly indicates its associated class. PNN calculates the Bayesian posterior probability for each category. During the training phase, the weights connecting the input and pattern layer are simply set as the copy of input vectors, i.e. $\mathbf{w}_i = \mathbf{x}_i$, for $i = 1,...,n$. This process is one of the fastest known training strategies. During the test phase, each pattern node performs a dot product operation with a new pattern vector $\mathbf{x}$ and a weight vector $\mathbf{w}_i$, expressed as $P_i = \mathbf{x} \bullet \mathbf{w}_i$. The final output pattern layer is obtained via a nonlinear transformation. Usually a Gaussian activation function $\exp\left((P_i - 1)/\sigma^2\right)$ is used. Here, σ is the smoothing parameter of the Gaussian kernel and is also the only user-dependent parameter. Note that if both the training patterns and the new patterns are normalized to unit length, the output of pattern layer can be represented as

$$\begin{aligned} &\exp\left((P_i - 1)/\sigma^2\right) \\ &= \exp\left(-(\mathbf{x}^T\mathbf{x} + \mathbf{w}_i{}^T\mathbf{w}_i - 2\mathbf{x}^T\mathbf{w}_i)/2\sigma^2\right), \\ &= \exp\left(-(\mathbf{x} - \mathbf{w}_i)^T(\mathbf{x} - \mathbf{w}_i)/2\sigma^2\right) \end{aligned} \qquad (4)$$

which is identical to the Parzen window function. In this sense, each pattern node provides the corresponding category node with the class conditional probability given the training pattern. These values are then summed up in the category layer for each category as the estimated probability for the new pattern. The label of the pattern can be predicted by just choosing the maximum probability.

III. Results

We performed the survival analysis on the DLBCL data set of Resenwald et al., which consists of measurements of 7,399 genes from 240 patients [5]. The data set is divided into a training set with 160 patients and a test set with 80 patients. The survival time for the patients ranges from 0 to 21.8 years. Any patient who lived longer than the median survival time (2.8 years) is placed into the low-risk group, otherwise, into the high-risk group. For those censored patients who left the follow-up before the median, we estimated the probability of their survival according to the Kaplan-Meier survival curve and then assigned its label.

During the training phase, we used leave one out cross validation as the error estimation for gene selection. Each time, a different single sample was left out as the test point and the other samples were used for training. Then we evaluated the prediction performance of the classifier on the independent test set.

We set the parameters for PSO as follows: w=0.7, $c_1 = c_2 = 2$, and V_{max}=2. We adjusted the value of δ in order to control the total number of selected genes in the subsets. Each time, the evolution is processed for 200 generations with 30 particles included in the swarm. The smoothing parameter of the Gaussian kernel is set to 2. Unlike other algorithms, we only have four parameters that are user-dependent. Their values can be easily determined and the performance is not sensitive to their change [2]. Fig.3 shows the Kaplan-Meier curves for the estimation of patient survival in the training set and the test set, respectively. We

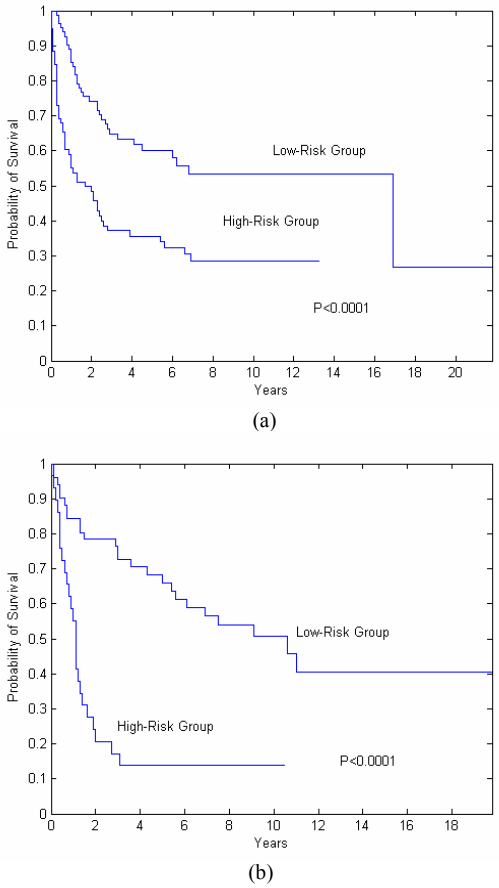

Fig. 3. Kaplan-Meier curves show significant differences in survival of the high-risk and low-risk group, based on the performance of PSO/PNN on the training set (a) and the independent test set (b).

set δ as 0.49, which leads to the selection about 92-185 genes. We chose the subset consisting of 113 genes that can achieve 80% classification accuracy on the test set and drew the curves. We also increased or decreased the value of δ to check the effect of gene subsets on the classification rate. We find in both ways, the performance is deteriorated. For example, the best result for the subsets including 5-30 genes is 73%. When all genes are used, the classification accuracy is only 52%. These results reflect the importance of gene selection in the prediction of clinical phenotypes, as its application in tumor classification [1-2]. We applied the log-rank test to test the difference between the low-risk and high-risk group, each of which is associated with a survival curve, as depicted in Fig. 3. The *p*-values for both the training set and the test set are less than 0.0001, which indicates the significant difference between the two risk groups, divided by the PSO/PNN method. There are 11 patients with less than 2.8 years survival time that are misclassified into the low-risk group in the test set. For the 5 misclassifications in the high-risk group, four of them are censored and the other is a patient died at 3.1 years. Furthermore, we calculated the frequency of each gene appearing in the selected 210 subsets. We find that the selection frequencies of 5 genes are more than 13% and those of 34 genes are more than 10%. This shows that PSO tends to choose a subset of genes associated with the phenotype in spite of the different initial conditions. However, the challenge to find the relation of genes with the survival still remains open due to the existing uncertain factors [16].

IV. Conclusion

We have proposed a PSO/PNN system for the analysis of patient survival through their gene expression profiling, which can be a significant factor for pharmaceutical selection and treatment. PSO is an effective computational technique for global optimization, in this case, for the selection of a subset of genes relevant to the phenotype. This process, which is also can be regarded as dimensionality reduction, is critically important in this type of analyses since cancer data sets usually consist of overwhelming number of measurements of gene expression levels compared with a very small set of samples. The experiment results on the DLBCT data set demonstrate that the methods can be very useful in connecting the clinical observance with gene expression profiling. Further research includes the simulation study on more survival data sets, the further investigation of the selected genes, and the application of a hybrid of PSO and evolutionary algorithm (EA) for gene selection. This design considers the complementary properties of PSO and EA and is more powerful for optimization.

Acknowledgment

Partial support for this research from the National Science Foundation, and from the M.K. Finley Missouri endowment, is gratefully acknowledged.

References

[1] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, pp. 531-537, 1999.

[2] R. Xu, and D. Wunsch, "Probabilistic neural networks for multi-class tissue classification with gene expression data," *Proceedings of International Joint Conference on Neural Networks 03*, vol. 3, pp. 1696-1701, 2003.

[3] D. Beer, S. Kardia, C. Huang, T. Giordano, A. Levin, D. Misek, L. Lin, G. Chen, T. Gharib, D. Thomas, M. lizyness, R. kuick, S. Hayasaka, J. Taylor, M. Iannettoni, M. Orringer, and S. Hanash, "Gene-expression profiles predict survival of patients with lung adenocarcinoma," *Nature Medicine*, vol. 8, no. 8, pp. 816-824, 2002.

[4] M. Garber, O. Troyanskaya, K. Schluens, S. Petersen, Z. Thaesler, M. Pacyna-Gengelbach, M. Rijn, G. Rosen, C. Perou, R. Whyte, R. Altman, P. Brown, D. Botstein, and I. Petersen, "Diversity of gene expression in adenocarcinoma of the lung," *Proc. of Natl. Acad. Sci.*, vol. 98, no. 24., pp. 13784-13789, 2001.

[5] A. Rosenwald, G. Wright, W. Chan, J. Connors, C. Campo, R. Fisher, R. Gascoyne, H. Muller-Hermelink, E. Smeland, and L. Staudt, "The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma," *The New England Journal of Medicine*, vol. 346, no. 25, pp. 1937-1947, 2002.

[6] M. Vijver, Y. He, L. Veer, H. Dai, A. Hart, D. Voskuil, G. Schreiber, J. Peterse, C. Roberts, M. Marton, M. Parrish, D. Atsma, A. Witteveen, A. Glas, L. Delahaye, T. Velde, H. Bartelink, S. Rodenhuis, E. Rutgers, S. Friend, and R. Bernards, "A gene-expression signature as a predictor of survival in breast cancer," *The New England Journal of Medicine*, vol. 347, no. 25, pp. 1999-2009, 2002.

[7] G. Glinsky, A. Glinskii, A. Stephenson, R. Hoffman, and W. Gerald, "Gene expression profiling predicts clinical outcome of prostate cancer," *J. Clin. Invest*. vol. 113, pp. 913-923, 2004.

[8] E. Bair and R. Tibshirani, "Semi-supervised methods to predict patient survival from gene expression data," *PLoS Biology*, vol. 2, no. 4, pp. 511-522, 2004.

[9] M. Shipp, K. Ross, P. Tamayo, A. Weng, J. Kutok, R. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. Pinkus, T. Ray, M. Koval, K. Last, A. Norton, T. Lister, J. Mesirov, D. Neuberg, E. Lander, J. Aster, T. Golub, "Diffuse large B-cell lymphoma outcome prediction by gene expression profiling and supervise machine learning," *Nature Medicine*, vol. 8, no. 1, pp. 68-74, 2002.

[10] T. Sørlie, C. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, M. Eisen, M. Rijn, S. Jeffrey, T. Thorsen, H. Quist, J. Matese, P. Brown, D. Botstein, P. Lønning, and A. Børresen-Dale, "Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications," *Proc. of Natl. Acad. Sci.,* vol. 98, no. 19, pp. 10869-10874, 2001.

[11] L. Li and H. Li, "Dimension reduction methods for microarrays with application to censored survival data," *Bioinformatics*, vol. 18, pp. 3406-3412, 2004.

[12] J. Kennedy, R. Eberhart, Y. Shi, "Swarm intelligence," Morgan Kaufmann Publishers, 2001.

[13] J. Kennedy and R. Eberhart, "A discrete binary version of the particle swarm optimization," *Proc. of Conf. on System, Man, and Cybernetics*, pp. 4104-4108, 1997.

[14] D. Specht, "Probabilistic Neural Networks," *Neural Networks*, vol. 3, pp. 109-118, 1990.

[15] R. Duda, P. Hart, and D. Stork, *Pattern classification*, 2nd Ed.. Wiley & Sons, New York, 2001.

[16] L. Ein-Dor, I. Kela, G. Getz, D. Givol, and E. Domany, "Outcome signature genes in breast cancer: is there a unique set?," *Bioinformatics*, vol. 21, pp. 171-178, 2005.