International Journal of Computer and Communication Technology

Volume 3 | Issue 3 Article 5

July 2012

Microarray Analysis Using Statistical Approach

Smita Patnaik
Dept. of Computer Applications, Institute of Technical Education & Research Siksha 'O' Anusandhan University, Bhubaneswar, smita608@gmail.com

Tripti Swarnkar
Deptt. Of CA, SOA University Bhubaneswar, tripti_sarap@yahoo.com

Microarray Analysis Using Statistical Approach

Smita Patnaik [1] & Tripti Swarnkar [2]
[1] smita608@gmail.com,[2] tripti_sarap@yahoo.com
Dept. of Computer Applications, Institute of Technical Education & Research
Siksha 'O' Anusandhan University, Bhubaneswar

Abstract- **Over the past few decades rapid developments in genomic and other molecular research technologies and developments in information technologies have combined to produce tremendous amounts of information related to molecular biology. It is not possible to research on a large number of genes using traditional methods. DNA Microarray is one such technology which enables to monitor the expression levels for tens of thousands of genes in parallel . A common task with Microarray data is to determine which genes are differentially expressed between two samples obtained under two different conditions. To solve this problem several Statistical methods have been proposed. The Support Vector Machine is one of the most efficient & widely used statistical method for Microarray classification. In this paper we have classified leukemia dataset by using support vector machine under two conditions and also showed the performance of different type of kernels.**

KEYWORDS

DNA Microarray;Support Vector Machine;Kernels

INTRODUCTION

Over the last few years the routine use of DNA microarrays has made possible the creation of large data sets of molecular information characterizing complex biological systems. DNA Microarray techniques present a novel way for geneticists to monitor interactions among tens of thousands of genes simultaneously and have become standard lab routines in gene discovery,disease diagnosis and drug design. Advancing Statistical methods and Machine learning techniques have played important roles in analyzing microarray datasets for discovering patterns hidden it. The support vector machine is a statistical method for microarray classification. The support vector machine (SVM) is a promising classification technique proposed by Vapnik and his group at AT&T Bell Laboratories (Cortes & Vapnik, 1995). SVM is a good tool for the two classifications. It can separate the classes with a particular hyperplane which maximizes a quantity called the margin. The margin is the distance from a hyperplane separating the classes to the nearest point in the dataset. The advantage of maximum margin criterion is its robust characteristic against noise in data, and making a solution unique for linearly separable problems. In addition, it is important that the SVM with a theoretically strong support is based on the statistical learning theory framework. An important finding of the statistical learning theory is that the generalization error can be bound by the sum of the empirical error and term, which depends on the VC dimension that characterizes the complexity of the approximating function class (Pardo & Sberveglieri, 2005; Vapnik, 1998). SVM has been extensively used as a classification tool with a great deal of success in a variety of area from object recognition (Oren, Papageorgiou, Sinha, Osuna, & Poggio, 1997) to classification of cancer morphologies (Mukherjee et al., 1999). In addition, it has also been successfully applied to a number of real-world problems such as handwritten characters and digit recognition (Cortes & Vapnik, 1995; Scholkopf, 1997; Vapnik, 1995), face detection (Osuna et al., 1997) and speaker identification (Schmidt, 1996). Features are called attributes, properties, variables, or characteristics. Feature selection (the so-called variable selection) has become the focus of much research in the area of application for which datasets with tens or hundred of thousands of variables are available. Feature selection problems are found in many machine learning tasks including classification, regression, time series prediction, etc. An appropriate feature selection can enhance the effectiveness and domain interpretability of an inference model. Liu and Motoda (1998) indicated that the effect of
feature selection are [1]to improve performance (speed of learning, predictive accuracy, or simplicity of rules); [2]to visualize the data for model selection; and [3] to reduce dimensionality and remove noise. The universal algorithms of feature selection are often divided into three lines: filters, wrappers, and embedded (Guyon & Elisseeff, 2003; Kohavi & John, 1997). Both wrappers and filters have encountered

some success with induction tasks, but they can be very computationally expensive for tasks with a larger number of variables. Although the embedded method has lower computational cost compared with the above, the exhaustive search is not good for dealing with the large features. Simply stated, these three methods may suffer from a block of wasting computational cost when variables are too large. Although the approach of the SVM with kernel function is useful for classification, its performance must be improved, especially for complex data. This is particularly important for people who want to obtain a high level of accuracy in advanced areas such as precision engineering and medical diagnosis. In addition to accuracy, feature selection is another substantial issue for classification. Although feature selection offers many advantages, it may face the risks of accuracy decreasing or over-fitting. Thus, how to achieve/keep the expected classification performance and avoid the risks in feature selection is an important consideration. In this study SVM with linear kernel, quadratic kernel, polynomial kernel and Gaussian Radius Base Function kernel (the so-called RBF kernel) has experimented respectively.

1.1 MATHEMATICAL BACKGROUND OF SVMS

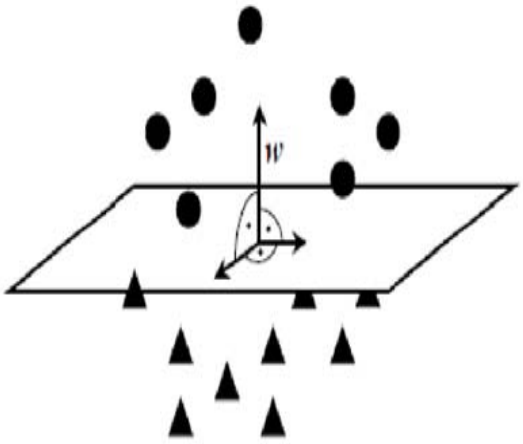
We start with a geometric intuition of SVMs and then give a more general mathematical formulation.

1.1.1 A GEOMETRICAL INTERPRETATION

A geometric interpretation of the SVM illustrates how this idea of smoothness or stability gives rise to a geometric quantity called the margin which is a measure of how well separated the two classes can be. We start by assuming that the classification function is linear[8].

$$f(x) = w \cdot x = \sum_{i=1}^{n} w_i x_i$$

where xi and wi are the i^{th} elements of the vectors x and w, respectively. The operation w · x is called a dot product. The label of a new point xnew is the sign of the above function, ynew = sign [f(xnew)]. The classification boundary, all values of x for which f(x) = 0, is a hyperplane[1] defined by its normal vector w. see figure (a)



Figure(a)

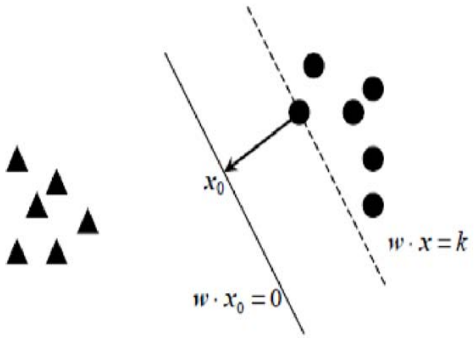
Assume we have points from two classes that can be separated by a hyperplane and x is the closest data point to the hyperplane, define x0 to be the closest point on the hyperplane to x. This is the closest point to x that satisfiesw · x = 0 (see Figure b). We then have the following two equations: w · x = k for some k, and w · x = 0.

Subtracting these two equations, we obtain w · (x −x0) = k. Dividing by the norm of w (the norm of w is the length of the vector w), we obtain:

$$\frac{w}{\|w\|} \cdot (x - x_0) = \frac{k}{\|w\|}$$

Where

$\|w\| = \sqrt{\sum_{i=1}^{n} w_i^2}$. Noting that w / ||w|| is a unit vector (a vector of length 1), and the vector x −x0 is parallel to w, we conclude that $\|x - x_0\| = \frac{|k|}{\|w\|}$.



Figure(b)

Our objective is to maximize the distance between the hyperplane and the closest point, with the constraint that the points from the two classes fall on opposite sides of the hyperplane. The following optimization problem satisfies the objective:

$$\max_{w} \min_{x_i} \frac{y_i(w \cdot x_i)}{\|w\|} \quad \text{subject to} \quad y_i(w \cdot x) > 0 \text{ for all } x_i$$

Note that y(w w x) = |k| when the point x is the circle closest to the hyperplane in Figure(b). For technical reasons, the optimization problem stated above is not easy to solve. One difficulty is that if we find a solution w, then c w for any positive constant c is also a solution. This is because we have not fixed a scale or unit to the problem. So without any loss of generality, we will require that for the point xi closest to the hyperplane k = 1. This fixes a scale and unit to the problem and results in a guarantee that yi(wwxi) ≥1 for all xi. All other points are measured with respect to the closest point, which is distance 1 from the optimal hyperplane. Therefore, we may equivalently solve the problem

$$\max_{w} \min_{x_i} \frac{y_i(w \cdot x_i)}{\|w\|} \quad \text{subject to} \quad y_i(w \cdot x_i) \geq 1$$

An equivalent, but simpler problem (Vapnik, 1998) is

$$\min_{w} \tfrac{1}{2}\|w\|^2 \text{ subject to } y_i(w \cdot x_i) \geq 1$$

Note that so far, we have considered only hyperplanes that pass through the origin. In many applications, this restriction is unnecessary, and the standard separable (i.e. the hyperplane can separate the two classes) SVM problem is written as

$$\min_{w,b} \tfrac{1}{2}\|w\|^2 \quad \text{subject to} \quad y_i(w \cdot x_i + b) \geq 1$$

where b is a free threshold parameter that translates the optimal hyperplane relative to the origin. The

distance from the hyperplane to the closest points of the two classes is called the margin and is 1/||w|| . SVMs find the hyperplane that maximize the margin. Figure c & d illustrates the advantage of a large margin

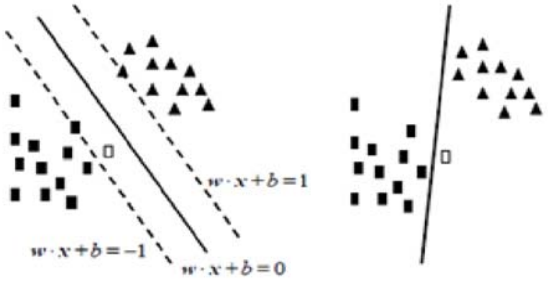
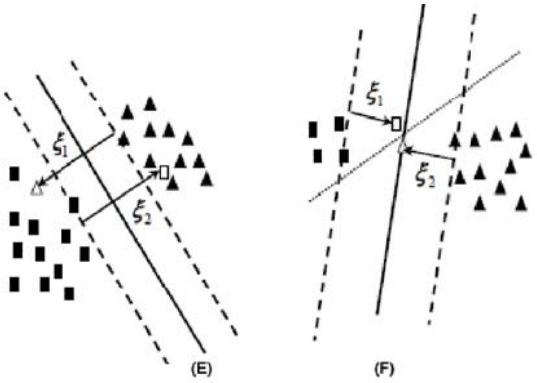


Figure (c) Figure (d)

In practice, data sets are often not linearly separable. To deal with this situation, we add slack variables that allow us to violate our original distance constraints. The problem becomes now:

$$\min_{w,b,\xi} \tfrac{1}{2}\|w\|^2 + C\sum_i \xi_i \quad \text{subject to} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i$$

where ξi ≥ 0 for all i. This new formulation trades off the two goals of finding a hyperplane with large margin (minimizing ||w||), and finding a hyperplane that separates the data well (minimizing the ξi). The parameter C controls this trade-off. This formulation is called the soft margin SVM. The parameter C controls this trade-off. It is no longer simple to interpret the final solution of the SVM problem geometrically SVM.



2

SVMs can also be used to construct nonlinear separating surfaces. The basic idea here is to nonlinearly map the data to a feature space of high or possibly infinite dimensions, x →φ(x). We then apply the linear SVM algorithm in this feature space. A linear separating hyperplane in the feature space corresponds to a nonlinear surface in the original

space. Using the data points mapped into the feature space, and we obtain Equation

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$\xi \geq 0$ for all i, where the vector w has the same dimensionality as the feature space and can be thought of as the

$$f(x) = w \cdot \phi(x) + b = \sum_{i=1}^{l} c_i \phi(x_i) \cdot \phi(x) + b$$

since the normal to the hyperplane can be written as a linear combination of the training points in the feature space,

$$w = \sum_{i=1}^{l} c_i \phi(x_i).$$

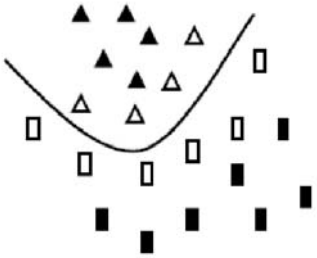
For both the optimization problem and the solution the dot product of two points in the feature spaces needs to be computed. This dot product can be computed without explicitly mapping the points into feature space by a kernel *function*, which can be defined as the dot product for two points in the feature space:

$$K(x_i, x_j) \equiv \phi(x_i) \cdot \phi(x_j)$$

So our solution to the optimization problem has now the form:

$$f(x) = \sum_{i=1}^{l} c_i K(x_i, x_j) + b$$

Most of the coefficients ci will be zero; only the coefficients of the points closest to the maximum margin hyperplane in the feature space will have nonzero coefficients. These points are called the support vectors. Figure (g)illustrates a nonlinear decision boundary and the idea of support vectors.



(g)

The following example illustrates the connection between the mapping into a feature space and the kernel function.Assume that we measure the expression levels of two genes, TrkC and SonicHedghog (SH). For each sample, we

$$\phi : x \to \left\{ x_{SH}^2, x_{TrkC}^2, \sqrt{2} x_{SH} x_{TrkC}, x_{SH}, x_{TrkC}, 1 \right\}$$

$$\begin{aligned} K(x,z) \equiv \phi(x) \cdot \phi(z) &= (x \cdot z + 1)^2 \\ &= x_{SH}^2 z_{SH}^2 + x_{TrkC}^2 z_{TrkC}^2 + 2 x_{SH} x_{TrkC} z_{SH} z_{TrkC} + \\ &\quad x_{SH} z_{SH} + x_{TrkC} z_{TrkC} + 1 \end{aligned}$$

which is called a second order polynomial kernel. Note, that this kernel uses information about both expression levels of individual genes and also expression levels of pairs of genes. This can be interpreted as a model that incorporates co-regulation information. Assume that we measure the expression levels of 7,000 genes.

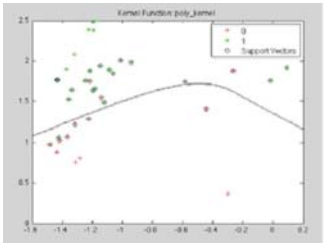
The feature space that the second order polynomial would map into would have approximately 50 million elements, so it is advantageous that one does not have to explicitly construct this map.

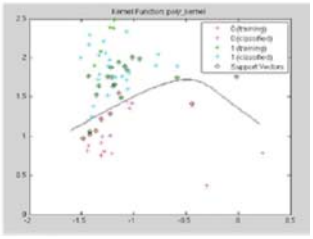
$$K(x, z) = (x^T z + 1)^p \text{ and } K(x, z) = \exp(-\|x - z\| / 2\sigma^2).$$

2.EXPERIMENTAL ACTIVITIES AND RESULT

Step 1 : We have taken the leukemia data set leu_data_label , which consits of a 72 x 3571 matrix of gene expression values (72 rows of samples and 3571 columns of expression values) and a 72 x 1 array of 72 class labels . This label array will have values 1 and 2 corresponding to the 72 samples. There are 2 class of samples, class 1 denotes acute lymphoblastic leukemia (abbreviated as ALL) , class 2 denotes acute myeloid leukemia (abbreviated as AML). there are 47 'class 1' samples and 25 'class 2' samples which have been experimentally determined.

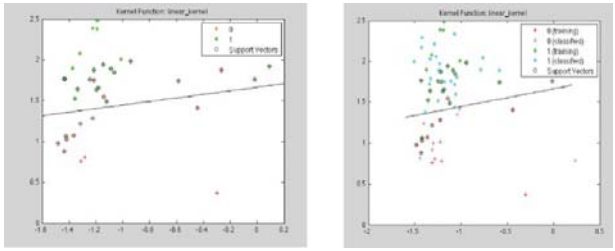
Step 2 : From the expression matrix (72 x 2) and randomly choose a part of the matrix as training data (say it may be 37 x 2, it has 37 samples) and has applied svmtrain to train the svm classsifier. We get the plot after training.



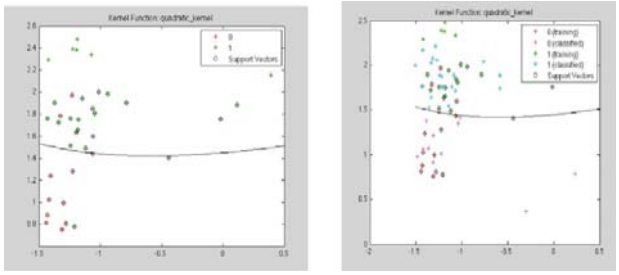


Step 3 : We have taken the remaining part as test data (35x 2 , it has 35 samples , so 37 + 35 = 72) and apply svmclassify with the trained classifier to test if the classification worked fine . We get the plot after classification. Now We have all the data, the training and test data on the same plot.

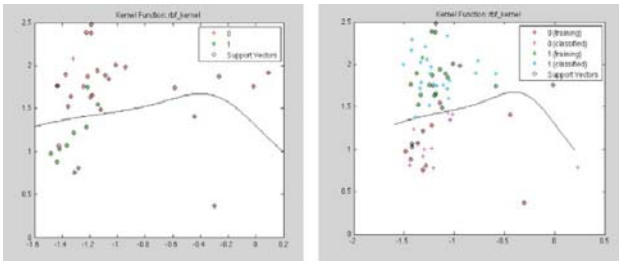
Step 4 : We calculate the performance of the classifier after it returns the classified classes .



After Training (fig i) After Classification (fig ii)



After Training (fig iii) After Classification (fig iv) After Training (fig v) After Classification(fig vi)



After Training (fig vii) After Classification (fig viii)

Kernel	Classification Performance (2 sets of gene expression values 72x2)	Classification Performance (full dataset, 72x3571)
Linear	0.80	0.9714
Quadratic	0.8571	0.8286
Polynomial(3= cubic)	0.80	0.3429
RBF	0.8286	0.6573

Table 1

3. CONCLUSION:

SVMs are considered a supervised computer learning method because they exploit prior knowledge of gene function to identity unknown genes of similar function from expression data.SVMs avoid several problems associated with unsupervised clustering methods. It has demonstrated that support vector machines can accurately classify genes into functional categories based upon expression data from DNA microarray hybridization experiments. Among the 4 types of kernels that it examined , SVM that uses a quadratic kernel function provides the best performance .

4. REFERENCE:

[1] Allwein E.L., Schapire R.E., Singer Y. (2000). Reducing multiclass to binary: a unifying approach for margin classifiers. Journal of Machine Learning Research 1:113-141.

[2] Amaldi, E., & Kann, V. (1998). On the approximation of minimizing non zero variables or unsatisfied relations in linear systems. Theoretical Computer Science, 209, 237–260. 2.Aronszajn, N. (1950). Theory of reproducing kernels. Transactions of the American Mathematical Society, 68(3), 337–404.

[3] Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases. Dept. Infor. Comput. Sci., Univ. California, Irvine, CA.

[4] Bhattacharjee A., Richards W.G, Staunton J., Li C., Monti S., Vasa P., Ladd C., Beheshti J., Bueno R., Gillette M., Loda M., Weber G., Mark E.F., Lander E.S., Wong W., Johnson B.E., Golub T.R., Sugarbaker D.J., Meyerson M. (2001). Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. Proc. Natl. Acad. Sci. USA 98:13790–13795.

[5] Bousquet O. and Elisseeff A. (2002). Stability and Generalization. Journal of Machine Learning Research 2, 499-526.

[6] Dietterich T.G. and Bakiri G. (1991). Error-correcting output codes: A general method for improving multiclass inductive learning programs. Proc. of the Ninth National Conference on Artificial Intelligence, AAAI Press, 572-577.

[7] Evgeniou T., Pontil M., Poggio T. (2000). Regularization Networks and Support Vector Machines. Advances in Computational Mathematics 13:1-50.