Support Vector Machine Classification of Cancer Using Gene Expression Data

Zhimin Chen
University of British Columnbia
Computer Science Department
zmchen@cs.ubc.ca

April 14, 2002

Abstract

DNA microarrays technology seems promising as a method to capture the characteristics of cancer tissues. Support vector machine is a robust classification method for high dimensional data. The combination of them is expected to produce a good method to classify cancer tissues. This report presents methods of applying support vector machine to classify cancer tissues with gene expression data under the context of binary classification problem and multi-class classification problem. The experiments of applying these method to analyze the AML/ALL dataset show that support vector machine performs well in classification of cancer with gene expression data.

1 Introduction

DNA microarrays can measure the activity levels of thousands of genes in tissue samples simultaneously and these expression data can provide valuable information to cancer diagnosis. Golub et al. [1] describe a case in which it is feasible to classify cancer solely by gene expression data. Furey et al. [2] present a method using support vector machines (SVM) to analyze the data. The first part of this project follows [2]'s approach and reproduces their result for the AML/ALL dataset. As additional work I build a multi-class classifier by combining multiple binary SVM classifiers. To test it I treat the B-cell ALL and the T-cell ALL as independent cancer classes so that there are three classes in Golubs AML/ALL dataset. The multi-class classifier can assign class labels to the samples in the test dataset with rather high accuracy in my experiment.

This report is organized as follows. Section 2 introduces gene expression data and Section 3 introduces SVM. Section 4 presents the SVM method and its experimental result to classify AML and ALL. Section 5 presents two multi-class SVM methods and their experimental results. Section 6 is about the future work and conclusion.

2 Gene Expression Data

DNA microarrays [3] allow the tracking of thousands of molecular reactions in parallel on a single chip. Though DNA microarrays can be used to detect the presence of a specific gene in a tissue sample or to sequence an unsequenced DNA string, their primary application is to measure the ratios of the amounts of the mRNAs of interest under two different conditions. DNA microarrays contain single-stranded DNAs representing thousands of different genes, each sitting at the assigned spot on the chip, and each spot has thousands of copies of a DNA strand for that particular gene. In the experiment, the chip gets hybridized with the mRNA molecules that are extracted from two tissue samples, one as the reference reflecting the background state and the other under a special condition set up in the experiment, and are labeled with two different fluorescent tags. Usually the DNA molecules complemtary to the mRNA molecules are used instead of hybridizing the chip directly with mRNA molecules, because the binding between DNAs is more stable than that between DNA and RNA. Then a scanner with a computer can produce a color readout from the array, and each number in the readout represents the ratio of the amount of a particular gene, or the expression level, under two conditions.

These gene expression data can be viewed as a feature vector of a tissue sample under a special condition. The goal of the cancer classifier is: given an expression vector of a tissue sample, determine accurately which class this sample belongs to.

3 Support Vector Machine

SVM [4][5][6] is a learning algorithm based on the principle of minimizing structural risk. In studies it shows a matching or significantly better error rate on test sets for a lot of learning tasks, such as object recognition, text categorization, etc. The intuition of SVM is to find an optimal hyperplane among the many that can separate the training example data. Let $d_+(d_-)$ denote the shortest distance from the hyperplane to the closest positive (negative) example. Then $d_+ + d_-$ denotes the margin of the hyperplane.The optimal hyperplane is the one that satisfies $d_+ = d_-$ and maximizes $d_+ + d_-$. (Figure 1)

Formally[4], let the training data be $\{< x_i, y_i >\}, i = 1, ..., m$, where each x_i is a feature vector of a sample and $y_i \in \{-1, 1\}$, and construct a hyperplane $w \cdot x + b = 0$ that separates the positive from the negative examples. The decision function is $y = sgn(w \cdot x + b)$. Scale w and b so that the closest positive examples are on the hyperplane $w \cdot x + b = 1$ and the closest negative examples are on the hyperplane $w \cdot x + b = -1$. (Figure 2) It follows that $y_i(w \cdot x + b) \geq 1, i = 1, ..., m$. Let x_1 be one of the closest positive data points and x_2 be one of the closest negative data points. The margin of the hyperplane is $\frac{w}{\|w\|} \cdot (x_1 - x_2)$. Notice that $w \cdot x_1 + b = 1$ and $w \cdot x_2 + b = -1$. The margin is:

$$\frac{w}{\|w\|} \cdot (x_1 - x_2) = \frac{w \cdot x_1 - w \cdot x_2}{\|w\|} = \frac{2}{\|w\|}$$

So the problem of contructing of the optimal hyperplane turns into a constrained opti-

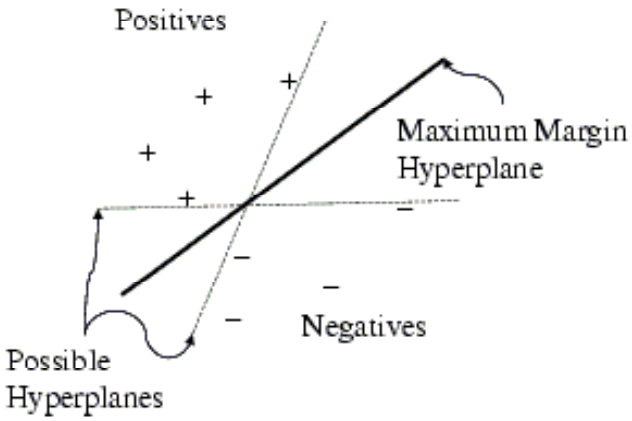

Figure 1: Optimal seperating hyperplane

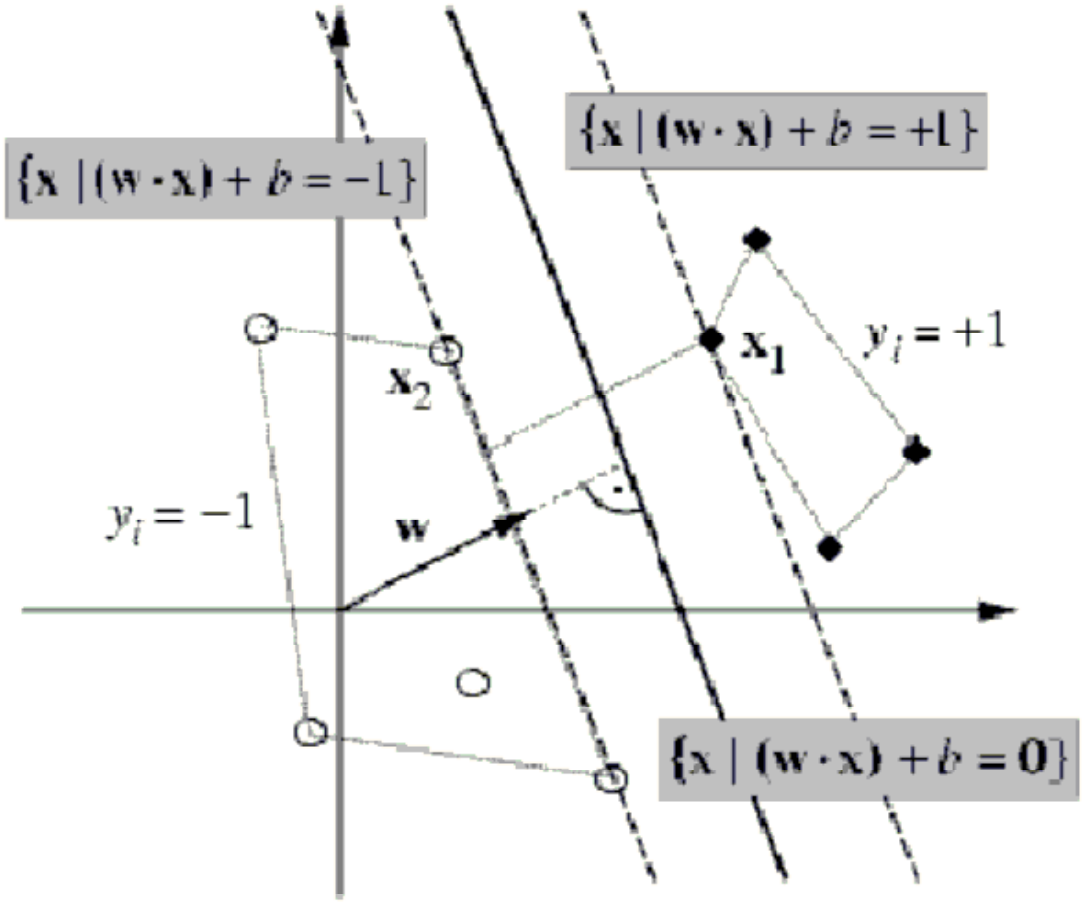

Figure 2: Margin of the optimal hyperplane

mization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2, \text{ subject to: } y_i(w \cdot x_i + b) - 1 \geq 0, i = 1, ..., m. \tag{1}$$

Introduce the vector of Lagrange multipliers $\alpha = (\alpha_1, ..., \alpha_m)$ and a Lagrangian:

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i (y_i(w \cdot x_i + b) - 1) \tag{2}$$

The primal problem (1) turns into the following dual problem:

$$\max_{w,b,\alpha} L(w, b, \alpha) \tag{3}$$

subject to:

$$\frac{\partial L(w, b, \alpha)}{\partial w} = 0, \tag{4}$$

$$\frac{\partial L(w, b, \alpha)}{\partial b} = 0, \tag{5}$$

$$\alpha_i \geq 0, i = 1, ..., m. \tag{6}$$

$\frac{\partial L}{\partial b} = 0$ leads to

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{7}$$

and $\frac{\partial L}{\partial w} = 0$ leads to

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i. \tag{8}$$

Substituting (7) back to $L(w, b, \alpha)$ in (2), we get

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i y_i (w \cdot x_i) + \sum_{i=1}^{m} \alpha_i. \tag{9}$$

Because $w = \sum_{i=1}^{m} \alpha_i y_i x_i$, so $\sum_{i=1}^{m} \alpha_i y_i (w \cdot x_i) = w \cdot (\sum_{i=1}^{m} \alpha_i y_i x_i) = w \cdot w = \|w\|^2$. Thus we have

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \|w\|^2 + \sum_{i=1}^{m} \alpha_i = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\|w\|^2. \tag{10}$$

Substituting (8) back to $L(w, b, \alpha)$ in (10), we get:

$$L(w, b, \alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}(\sum_{i=1}^{m} \alpha_i y_i x_i, \sum_{i=1}^{m} \alpha_i y_i x_i) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i y_i \alpha_j y_j (x_i, x_j). \tag{11}$$

Finally, the optimization problem becomes

$$\min_{\alpha}(-L(w,b,\alpha)) = \min_{\alpha}(\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i y_i \alpha_j y_j (x_i, x_j) - \sum_{i=1}^{m}\alpha_i) \tag{12}$$

$$\text{subject to: } \sum_{i=1}^{m}\alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, i = 1, ..., m. \tag{13}$$

The decision function is

$$y = sgn(w \cdot x + b) = sgn((\sum_{i=1}^{m} y_i \alpha_i x_i, x) + b) = sgn(\sum_{i=1}^{m} y_i \alpha_i (x_i, x) + b). \tag{14}$$

To compute this decision function we also need to know b. Going back to the primal problem, the w, b, and α should meet the Karush-Kuhn-Tucker complementary condictions

$$\alpha_i [y_i (w \cdot x_i + b) - 1] = 0, i = 1, ..., m. \tag{15}$$

Since from (14) we can compute m different bs, in the implementation one can just pick up the median of these bs to deal with the problem of numerical accuracy.

Armed with the quadprog function in MATLAB, it is trivial to implement the SVM. The $quadprog(H, f, A, b, Aeq, beq, lb, ub)$ solves the quadratic programming problem:

$$\min_{x} \frac{1}{2} x^T H x + f^T x \tag{16}$$

$$\text{such that } Ax \leq b \tag{17}$$

$$Aeq \cdot x = beq \tag{18}$$

$$lb \leq x \leq ub. \tag{19}$$

In the case of SVM, $H = diag(\alpha_1 y_1, ..., \alpha_m y_m) \times K \times diag(\alpha_1 y_1, ..., \alpha_m y_m)$, $f = (-1, -1, ..., -1)$, $Aeq = (y_1, y_2, ..., y_m)$, $beq = (0, 0, ..., 0)$, $lb = (0, 0, ..., 0)^T$, and $ub = (+\infty, +\infty, ..., +\infty)^T$), where $K = [k_{ij}]$ and $k_{ij} = (x_i, x_j) = x_i \cdot x_j$.

In the general cases the data may not be separable by a hyperplane. To tackle this problem, one needs a function ϕ to map the data into a higher dimensional feature space in which they can be separated by a hyperplane. The good news is that it is possible to compute the decision function without the explicit knowledge of ϕ. (Figure 3) Let $k(x, y) = (\phi(x), \phi(y))$. The problem of non-linear SVM turns into

$$\min_{\alpha} -L(w,b,\alpha) = \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i y_i \alpha_j y_j k(x_i, x_j) - \sum_{i=1}^{m}\alpha_i \tag{20}$$

$$\text{subject to: } \sum_{i=1}^{m}\alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, i = 1, ..., m. \tag{21}$$

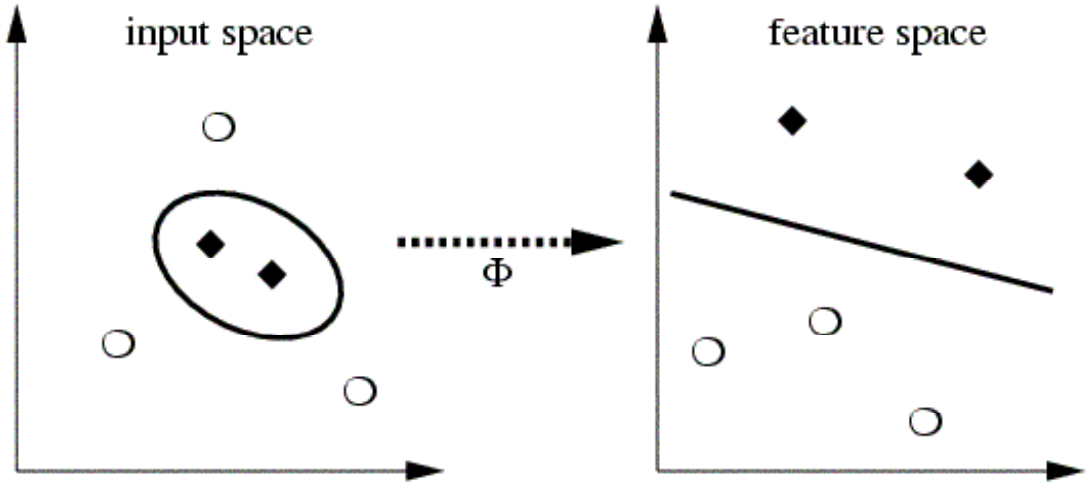

Figure 3: Non-linear SVM

and the decision function turns into

$$y = sgn(\sum_{i=1}^{m} y_i \alpha_i k(x_i, x) + b). \tag{22}$$

A legal kernel function $k(x, y) = (\phi(x), \phi(y))$ should meet the following Mercer's condition: $k(x, y)$ is a kernel function if and only if for any $g(x)$ such that $\int [g(x)]^2 dx$ is finite, then $\int k(x, y)g(x)g(y)dxdy \geq 0$. An example of kernel function is the Gaussian kernel function $k(x, y) = e^{-\|x-y\|^2/2\sigma^2}$.

4 Experimental Result

The SVM is applied to Golub's AML/ALL dataset. The scoring function to select features is the same as that used by Golub. Let the training dataset be $\{< x_1, y_1 >, ..., < x_m, y_m > \}, i = 1, ..., m$, where each x_i is a sample's gene expression data and $y_i \in \{-1, 1\}$, and Let x_{ij} is the expression level of the the j-th gene of sample x_i. Define $S(j)$, the scoring function for the j-th gene as follows.

$$\delta_+(i, j) = \begin{cases} 1 & \text{if } y_i = 1; \\ 0 & \text{if } y_i = -1. \end{cases}$$

$$\delta_-(i, j) = \begin{cases} 0 & \text{if } y_i = 1; \\ 1 & \text{if } y_i = -1. \end{cases}$$

$$\mu_+(j) = \frac{\sum_{i=1}^m \delta_+(i,j)x_{ij}}{\sum_{i=1}^m \delta_+(i,j)}.$$

$$\mu_-(j) = \frac{\sum_{i=1}^m \delta_-(i,j)x_{ij}}{\sum_{i=1}^m \delta_-(i,j)}.$$

$$\sigma_+(j) = \left(\frac{\sum_{i=1}^m \delta_+(i,j) \times (x_{ij} - \mu_+(j))^2}{\sum_{i=1}^m \delta_+(i,j)}\right)^{\frac{1}{2}}.$$

$$\sigma_-(j) = \left(\frac{\sum_{i=1}^m \delta_-(i,j) \times (x_{ij} - \mu_-(j))^2}{\sum_{i=1}^m \delta_+(i,j)}\right)^{\frac{1}{2}}.$$

$$S(j) = \frac{\mu_+(j) - \mu_-(j)}{\sigma_+(j) + \sigma_-(j)}.$$

Selecting n features is just selecting the n genes with largest score and the n genes with the smallest score.

Experiments were performed on the dataset by selecting 25, 50 and 500 features, respectively. The SVM perfectly classifies all the testing samples in the cases of using 50 and 500 features, and misclassifies 2 samples in the case of using 25 features. The SVM is also tested on the dataset with the whole 7129 features and misclassifies 1 sample.

The preceding experiments were repeated on the SVM with a Gaussian kernel. One problem with the Gaussian is that $k(x,y)$ approaches 0 exponentially with $\|x-y\|^2$ and soon goes beyond the boundary of accuracy in MATLAB. To counteract this effect I choose σ(the parameter in Gaussian function) as 5000, 10000, 50000 and 100000 in the cases of using 25, 50, 500 and all features, respectively. The accuracy of the SVM with Gaussian kernel is comparable to that of linear SVM. It perfectly classifies all the testing samples in the case of 500 features, but misclassifies 1 sample in the case of using 25, 50 and all features, respectively.

The results of these experiments are summarised in Table 1.

5 Multiclass Classification

Basically, SVM is a binary classifier. [7] provides a general framework to reduce the multiclass problem to multiple binary classfiers. The idea is to train multiple binary

Table 1: Summary of Experimental Results

	25 Features		50 Features		500 Features		All Features	
	ALL	AML	ALL	AML	ALL	AML	ALL	AML
Linear SVM	20/20	12/14	20/20	14/14	20/20	14/14	19/20	14/14
Gaussian SVM	20/20	13/14	20/20	13/14	20/20	14/14	19/20	14/14

classfiers $f_1(x)$,...,$f_k(x)$, and to define a k-ary function f to map the output of the binary classifiers to a class label, i.e., $y = f(f_1(x), ..., f_k(x))$.

One way is the ECOC(Error Correcting Output Code) method. In ECOC the output of each $f_i(x)$ is +1 or -1. There is a coding matrix where each row is a code as a vector of +1/-1 for a class. The decision function f is to choose the row closest to $(f_1(x), ..., f_k(x))$ according to some distance(for instance, Hamming distance), and return the class label associated with that row.

The algorithm tested here is the standard method to build a N-class SVM[8]. First construct N SVMs for N classes. The ith SVM is trained with all the samples in the ith class with label +1 and all the other samples with label -1. Let the decision function of the ith SVM be $f_i(x) = \frac{1}{\|w_i\|}(w_i \cdot x + b)$, i.e., the normalized distance from x to the ith optimal hyperplane. The dicision function is:

$$y = \arg \max_{i=1,...,k}(f_i(x)).$$

The ALL class has two subclasses, namely B-cell ALL and T-Cell ALL, in Golub's dataset. There are 8 samples of T-ALL in the training set and 1 sample of T-ALL in the test set. Since there are too few T-ALL samples in the test set, the dataset is rearranged as follows. The first 3 T-ALL samples (sample 2,3 and 6) in the training set are put into the test set and the one (sample 55)in the test set is put into the training set. As a result, there are 6 T-ALL samples in the new training set and 3 T-ALL samples in the new test set.

The scoring function for feature selection is also extended as follows. For a pair of classes i and j, the scoring function for this pair is: for the k-th gene x_k in the dataset, $S_{ij}(x_k) = \frac{\mu_k^i - \mu_k^j}{\sigma_k^i + \sigma_k^j}$, where $\mu_k^i(\mu_k^j)$ is the mean of all the samples labeled as class i(class j) and $\sigma_k^i(\sigma_k^j)$ is the standard deviation of all the samples labeled with class i(class j). Selecting n features is selecting the n genes with largest score S_{ij} and the n genes with the smallest score S_{ij} for each pair of classes (i, j).

Experiments were performed on the dataset by selecting 50, 100 and 500 features respectively. The SVM perfectly classifies all the testing samples in the case of 500 features, and misclassifies 1 sample in the cases of 50 and 100 features. The SVM is also tested on the dataset with the whole 7129 features and misclassifies 2 samples. The results of these experiments are summarised in Table 2.

A disadvantage of the standard N-class SVM is that there is no bound for the generalization error (the error rate on the test set). [8] proposed a DAG algorithm and proved a upper bound of the generalization error.

Table 2: Experimental Results for the standard N-class SVM

	T-ALL (3 samples)	B-ALL (19 samples)	AML (14 samples)
50 Features	3/3	18/19	14/14
100 Features	3/3	19/19	13/14
500 Features	3/3	19/19	14/14
All Features	3/3	17/19	14/14

Table 3: Experimental Results for the DAG SVM

	T-ALL (3 samples)	B-ALL (19 samples)	AML (14 samples)
50 Features	3/3	18/19	14/14
100 Features	3/3	19/19	13/14
500 Features	3/3	19/19	13/14
All Features	3/3	17/19	14/14

The DAG algorithm is as follows.

```
DAGSVM(x)
{
  for each pair of class i and class j{
    train a classifier for i and j only on the subset of training points labelled by i and j;
  }
  classList=[1..N];
  while (classList has more than 1 element){
    cFirst=first element of classList;
    cLast=last element of classList;
    test x with the binary SVM for pair (cFirst, cLast);
    if (x belongs to cFirst class)
      remove the last element of classList;
    else
      remove the first element of classList;
  }
  return the only element in classList as the class label;
}
```

The DAG algorithm is tested as the standard N-class SVM algorithm. The SVM misclassifies 1 sample in the cases of using 50, 100 and 500 features, respectively, and misclassifies 2 samples in the case of using all features. The results of these experiments are summarised in Table 3.

6 Future Work and Conclusion

Solving the problem of multiclass classification by using multiple binary classifiers cannot capture the correlation among multiple(≥ 3) classes. [9] proposed a method to construct a N-class SVM directly, without the use of multiple binary SVMs. It is worthwhile to see how their algorithm works in the application of classification of gene expression data.

A significant disadvantage of the SVM approach in this project is that it cannot provide a confidence level for the prediction. The output function $f(x) = w_i \cdot x + b$ can be viewed as a measure of confidence. More work is needed to determine how to convert it to a confidence percentage.

In addition to choosing the optimal hyperplane, SVM can use slack variables $\xi_i \geq 0$ to avoid overfitting by relaxing the constraints in (1) as $y_i(w \cdot x + b) \geq 1 - \xi_i$. This is equivalent to using a soft margin instead of a hard one and can reduce the prediction error. If the dataset is large, it may be necessary to adopt this approach.

In conclusion, SVM gives prediction with comparable accuracy to Golub's approach for the AML/ALL dataset. The N-class SVMs, the standard algorithm and the DAG algorithm, also give rather accurate predition for the multiclass classfication problem. Because the AML/ALL dataset has only 38 training samples and 34 test samples, more data are needed for further experiments.

References

[1] 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. Bloomfied, E. S. Lander.(1999) Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science 286:531-537, 1999.

[2] Terrence S. Furey, Nello Cristianini, Nigel Duffy, David Bednarski, Michel Schummer, David Haussler. (2000) Support Vector Machine Classification and Validation of Cancer Tissue Samples Using microarrays Expression Data. Bioinformatics 16:906-914, 2000.

[3] Stephen H. Friend, Roland B. Stoughton.(2002) The Magic of Arrays. Scientific American, Feb. 2002

[4] Berhard Scholkopf.(2000) Statistical Learning and Kernel Methods. MSR-TR 2000-23, Microsoft Research, 2000.

[5] Christopher J.C. Burges.(1998) A Tutorial on Support Vector Machine for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2):955-974, 1998.

[6] Vladimir N. Vapnik.(1995) The Nature of Statistical Learning Theory. Springer, New York, 1995.

[7] Erin Allwein, Robert Schapire and Yoram Singer.(2000) Reducing Multiclass to Binary: A unifying Approach for Margin Classifiers. Proc. 17th International Conf. on Machine Learning. Morgan Kaufmann, San Francisco, CA, 2000.

[8] John C. Platt, Nello Cristianini, John Shawe-Taylor.(2000) Large Margin DAGs For Multiclass Classfication. Advances in Neural Information Processing Systems, 12 ed. S.A. Solla, T.K. Leen and K.-R. Muller, MIT Press, 2000.

[9] Koby Crammer, Yoram Singer.(2001) On the Algorithmic Implementation of Multiclass Kernel-based Vector Machinces. Journal of Machine Learning Research, 2001.