MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY
and
CENTER FOR BIOLOGICAL AND COMPUTATIONAL LEARNING
DEPARTMENT OF BRAIN AND COGNITIVE SCIENCES

A.I. Memo No.1677
C.B.C.L Paper No.182

Support Vector Machine Classification of Microarray Data

S. Mukherjee, P. Tamayo, D. Slonim, A. Verri, T. Golub, J.P. Mesirov, and T. Poggio

This publication can be retrieved by anonymous ftp to publications.ai.mit.edu.
The pathname for this publication is: ai-publications/XXXXX

Abstract

An effective approach to cancer classification based upon gene expression monitoring using DNA microarrays was introduced by Golub et. al. [3]. The main problem they faced was accurately assigning leukemia samples the class labels acute myeloid leukemia (AML) or acute lymphoblastic leukemia (ALL). We used a Support Vector Machine (SVM) classifier to assign these labels. The motivation for the use of a SVM is that DNA microarray problems can be very high dimensional and have very few training data. This type of situation is particularly well suited for an SVM approach. We achieve slightly better performance on this (simple) classification task than Golub et. al.

This report describes research done within the Center for Biological and Computational Learning in the Department of Brain and Cognitive Sciences and at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. This research is sponsored by a grant from the Office of Naval Research under contract No. N00014-93-13085. Additional support is provided by Eastman Kodak Company, Daimler-Benz AG, Siemens Corporate Research Inc., AT&T, Digital Equipment and Central Research Institute of Electric Power Industry.

1 Introduction

The problem of cancer classification has clear implications on cancer treatment. Additionally, the advent of DNA microarrays introduces a wealth of genetic expression information for many diseases including cancer. An automated or generic approach for classification of cancer or other diseases based upon the microarray expression is an important problem.
A generic approach to classifying two types of acute leukemias was introduced in Golub et. al. [3]. They achieved good results on the problem of classifying acute myeloid leukemia (AML) versus acute lymphoblastic leukemia (ALL) using 50 gene expressions. They selected these 50 genes from 6817 genes in the DNA microarray based on the training set. Their approach to classification consisted of summing weighted votes for each gene on the test data, and looking at the sign of the sum.
We constructed Support Vector Machine (SVM) classifiers [7] for this problem. Our motivation was that SVMs have performed very well for a wide variety of classification problems [6] including microarray data [1]. We achieve better results than Golub et. al. without any feature selection step. We classify using all 7129 genes (the dataset contains 7129 genes including some control genes). The output of classical SVMs is a class designation ± 1. In this particular application it is important to be able to reject points for which the classifier is not confident enough. We introduced a confidence interval on the output of the SVM that allows us to reject points with low confidence values. It is also important in this application to infer which genes are important for the classification. In appendix A we describe preliminary work on a feature selection algorithm for SVM classifiers.

2 Classification Results

The data consisted of 38 training samples and 34 test samples. Each sample was a vector corresponding to 7129 genes. Each element in this vector is a $\log_{10}$ normalized expression value. This means that the expression level of each gene is normalized by the sample mean and variance of the training set and the logarithm is taken.
In Golub et. al. the top 50 "informative" genes were selected and used for the classification problem. We generated 4 data sets using 7129 genes, the top 999 genes, the top 99 genes, and the top 49 genes using the same criteria. Their criteria was the following. For each gene look at the statistic:

$$P(j) = \left| \frac{\mu_1(j) - \mu_{-1}(j)}{\sigma_1(j) + \sigma_{-1}(j)} \right|, \tag{1}$$

where j is the gene index, μ_1 is the mean of class 1 for gene j, μ_{-1} is the mean of class -1 for gene j, σ_1 is the standard deviation of class 1 for gene j, and σ_{-1} is the standard deviation of class -1 for gene j. The genes are then ranked in descending order according to $P(j)$ and the top values correspond to "informative" genes.
Golub et. al. classified 29 of the 34 test data correctly. The remaining 5 were rejects and of those 2 were errors.

2.1 Classification Without Rejections

Linear SVMs were constructed using vectors of 49, 99, 999 and 7129 gene expressions. The SVM was trained on the 38 points in the training set and tested on the 34 points in the test set. The

output of the SVM on the test set is a real number, d, that gives the distance from the optimal hyperplane. In standard SVMs, classification depends on the sign of d.
The training set was perfectly separable, meaning 100% accuracy in classifying the training data. A leave-one-out estimator on the training data also gave us 100% accuracy. The test set performances ranged from 0 to 2 errors for the data sets, see table (1). See figure (1) for the d values for the test data.

genes	errors
7129	1
999	0
99	0
49	2

Table 1: Number of errors in classification (without rejections) for various number of genes with the linear SVM.

Using nonlinear SVMs (polynomial kernels) did not improve performance. This would seem to indicate an additive linear model on the probability of gene expressions given a class.

2.2 Classification With Rejects

To reject points near the optimal hyperplane for which the classifier may not be very confident of the class label we introduced confidence levels based on the SVM output, d. These confidence levels are a function of d and are computed from the training data.
This allows us to reject samples below a certain value of $|d|$ because they do not fall within the confidence level. Introducing confidence levels resulted in 100% accuracy for all four cases and between 0 and 4 rejects, depending on the data set, table (2). Figure (2) plots the d values for the test data and the classification and rejection intervals.

genes	rejects	errors	confidence level	$\|d\|$
7129	3	0	$\sim 93\%$	.1
999	0	0	$\sim 95\%$	.08
99	2	0	$\sim 95\%$	.08
49	4	0	$\sim 93\%$	.165

Table 2: Number of errors, rejects, confidence level, and the $|d|$ corresponding to the confidence level for various number of genes with the linear SVM.

The computation of the confidence level is based on a Bayesian formulation and the following assumption for SVMs:

$$p(c|\mathbf{x}) \approx p(c|d).$$

We can rewrite $p(c|d)$ as

$$p(c|d) \propto p(d|c)p(c).$$

For our problem, we assume $p(1) = p(-1)$ and that $p(d|1) = p(-d|-1)$ this allows us to simply estimate $p(|d|\,|\{1, -1\})$. We make the previous assumptions so that we only have to estimate one confidence level based upon $|d|$ rather than two confidence levels, one for class 1 and one for class -1.

We use the leave-one-out estimator on the training data to get 38 $|d|$ values. We then estimate the distribution function, $\hat{F}(|d|)$ from the $|d|$ values. This was done using an automated non-parametric density estimation algorithm which has no free parameters [5]. The confidence level $C(|d|)$ is simply

$$C(|d|) = 1 - \hat{F}(|d|).$$

Figure (3) is a plot of the confidence level as a function of $|d|$ for the four cases. If we look at the d for the two classes separately we would get two confidence levels, figure (4).

2.3 Removal of Important Genes and Higher Order Information

We examined how well the SVM performed when the most important genes according to criteria (1) were removed. We also examined whether higher order interactions helped when important genes are removed.
Higher order statistics seem in fact to increase performance when the problem is artificially made more difficult by removing between 10 and 100 of the top features. Above this high order kernels hindered performance. This result is consistent with the concepts of generalization error which the SVM algorithm is based upon. When the data is less noisy the advantage of the flexibility of a more complicated model outweighs the disadvantage of the possibility of overfitting. When the data is noisy these aspects are reversed so a simpler model performs better. SVM performed well until 999 features were removed (see table (3)).

2.4 Treatment Success vs. Failure

Another problem addressed was prediction of treatment failure for a subset of the AML data. There were only 15 examples for this problem so we used the leave-one-out procedure to estimate the performance. We performed at chance level, 8 errors out of 15 points.

2.5 T vs. B cells in Acute Lymphoblastic Leukemia

There were two key subclasses of the ALL case, those that arose from T-cells and B-cells. We used a linear SVM to predict these two subclasses.
For this problem we used the same training set of 33 examples as Golub et. al. did. On leave-one-out estimates they classified 32 out of the 33 examples and rejected 1 example. This was the case whether they used the 50 or 200 most significant genes. For the same leave-one-out estimate we classify 32 or 33 examples correctly depending on whether we use all 7129 genes or the 999 most significant genes, table (4).
The results were the same when we performed a leave-one-out estimate on all 47 B-cell vs. T-cell cases. When all features are used 46 out of the 47 examples are classified and 1 is rejected. Using the top 999 features we classified 47 out of 47 examples.

3 Conclusion

A linear SVM classifier with a rejection level based upon confidence values performs well for both the AML vs. ALL and B vs. T cell classification problems. The prediction of failure vs. success of chemotherapy was at chance level. This performance was achieved without any gene

genes removed	1st order	2nd order	3rd order
10	2	1	1
20	3	2	1
30	3	3	2
40	3	3	2
50	3	2	2
100	3	3	2
200	3	3	3
300	3	4	4
400	4	4	4
500	4	4	4
600	4	5	5
700	3	3	3
800	3	3	3
900	3	4	7
1000	3	5	6
1100	4	6	6
1200	5	6	7
1300	7	8	8
1400	7	7	7
1500	7	7	8

Table 3: Number of errors as a function of the order of polynomial and the number of important genes removed.

genes	rejects	errors	confidence level
7129	1	0	$\sim 95\%$
999	0	0	$\sim 95\%$

Table 4: Number of errors, rejects, confidence level, for 7129 and 999 genes for the B vs. T cell problem with the linear SVM.

selection. It was also shown that the SVM classifier remained accurate even when the 1000 most significant genes were not used in the classifier. The fact that a linear SVM did as well as a polynomial classifier (when either all genes of the top genes are used) supports the assumption of Golub et. al. about the additive linearity of the genes in classification in this case. We expect that advantages of nonlinear SVM classifiers will be more obvious in more difficult problems in which interactions of several genes play a significant role.

A Classification With Gene Selection

Feature selection has two purposes in this problem: it can be used to improve generalization performance and to infer which genes are relevant in discriminating the two types of leukemias. In preliminary work we formulated a feature selection algorithm within the context of a SVM classifier. The basic principle is to rescale the input space such that the margin in feature space increases subject to the constraint that the volume feature space remains constant throughout feature rescaling steps.
The SVM classifier has the following form

$$d(\mathbf{x}) = \sum_{k=1}^{n} \alpha_k K(\mathbf{x}, \mathbf{x}_k) + b \tag{2}$$

where n is the number of support vectors and $K(\mathbf{x}, \mathbf{x}_k)$ is the kernel function. Feature selection uses the following iterative algorithm. First the standard SVM functional is minimized: given points $\{\mathbf{x}_1, ..., \mathbf{x}_\ell\}$ in $\mathbb{R}^h$ the following functional is minimized with respect to α

$$-\sum_{i=1}^{\ell} \alpha_i + \frac{1}{2}\sum_{i,j}^{\ell} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{3}$$

subject to

$$C \geq \alpha_i \geq 0\,, \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0.$$

Then the following functional is minimized with respect to the diagonal matrix $\mathbf{P}$ (with elements p_f)

$$\frac{1}{2}\sum_{i,j}^{\ell} y_i y_j \alpha_i \alpha_j K(\mathbf{P}\mathbf{x}_i, \mathbf{x}_j) \tag{4}$$

subject to

$$p_f \geq 0\,, \quad \sum_{f=1}^{h} g(p_f) = N,$$

where N can be interpreted as the number of expected features and imposes the constant volume constraint. The function $g(p_f) = p_f^2$ for linear and Gaussian kernels due to the properties of the mapping from input space to feature space [2], for polynomial kernels the function is more complicated but is analytic. Functional (4) is minimized using gradient descent with projection. Once the $\mathbf{P}$ is computed, the features corresponding to the top m elements are retained reducing the problem from $\mathbb{R}^h$ to $\mathbb{R}^m$. A SVM classifier is now constructed using the training data in $\mathbb{R}^m$. One can iteratively minimize functionals (3) and (4) to select features and maximize the margin. For details and extensions of this algorithm see ([4]).
We applied this algorithm to the leukemia data. With a linear SVM classifier we achieved 100% performance *with no rejects* on the test set using the top 40 genes selected. We were able to classify 32 of the 34 cases correctly using 5 genes.

References

[1] M. Brown, W. Grundy, D. Lin, N. Christianini, C. Sugnet, M. Ares Jr., and D. Haussler. Support vector machine classification of microarray gene expression data. UCSC-CRL 99-09, Department of Computer Science, University California Santa Cruz, Santa Cruz, CA, June, 1999 1999.

[2] C.J.C Burges. *Geometry and Invariance in Kernel Based Methods.* M.I.T. Press, Cambridge, MA, 1999.

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

[4] S. Mukherjee. A feature selection algorithm for support vector machines. Technical report. In progress.

[5] S. Mukherjee and V. Vapnik. Multivariate density estimation: An svm approach. AI Memo 1653, Massachusetts Institute of Technology, 1999.

[6] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *International Conference on Computer Vision*, pages 555–562, Bombay, India, 1998.

[7] V. N. Vapnik. *Statistical learning theory.* J. Wiley, 1998.

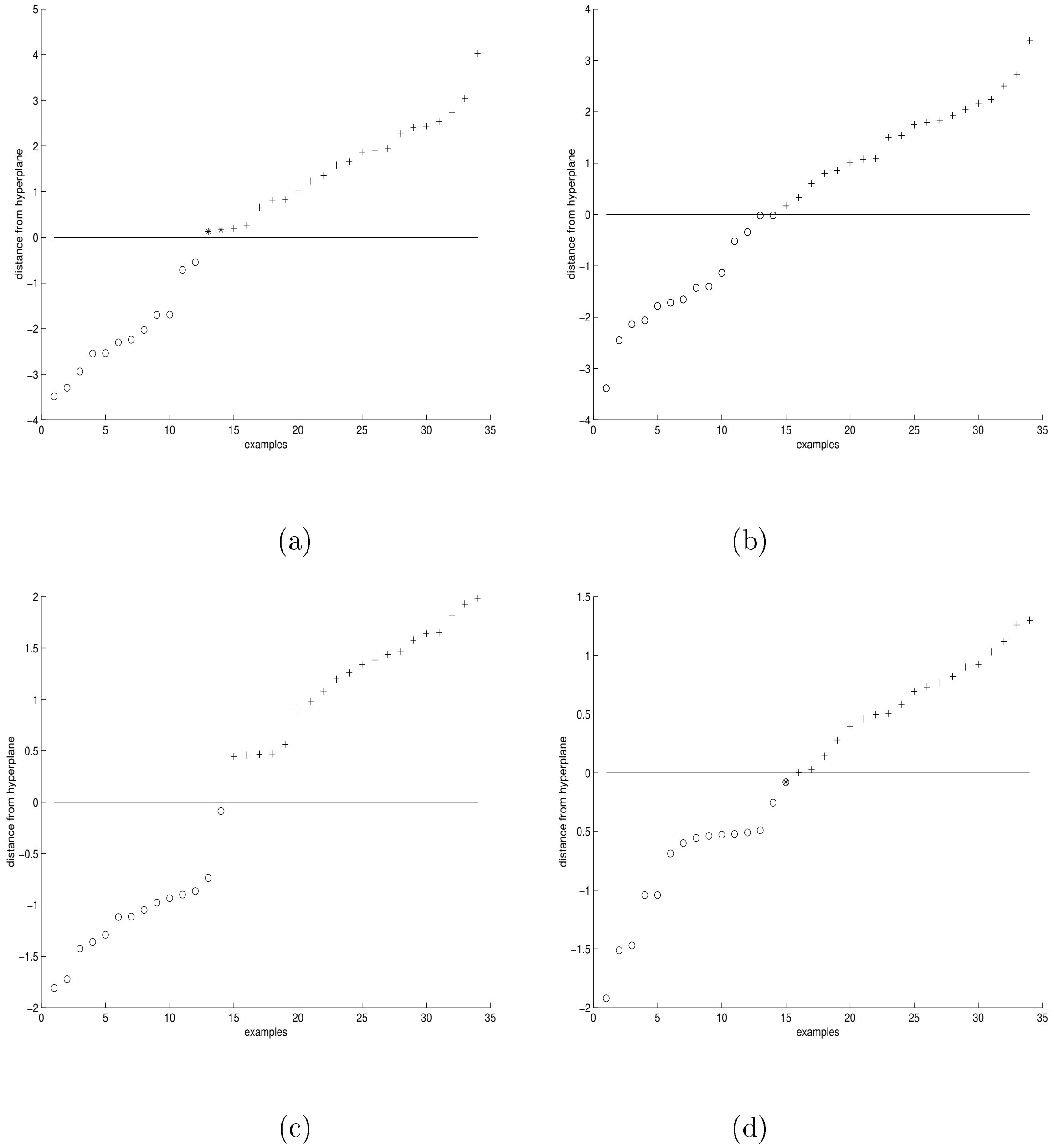


Figure 1: Plots of the distance from the hyperplane for test points for (a) feature vector of 49 (b) feature vector of 99 (c) feature vector of 999 (d) feature vector of 7129. The + are for class ALL, the o for class AML, the $*$ are mistakes, and the line indicates the decision boundary.

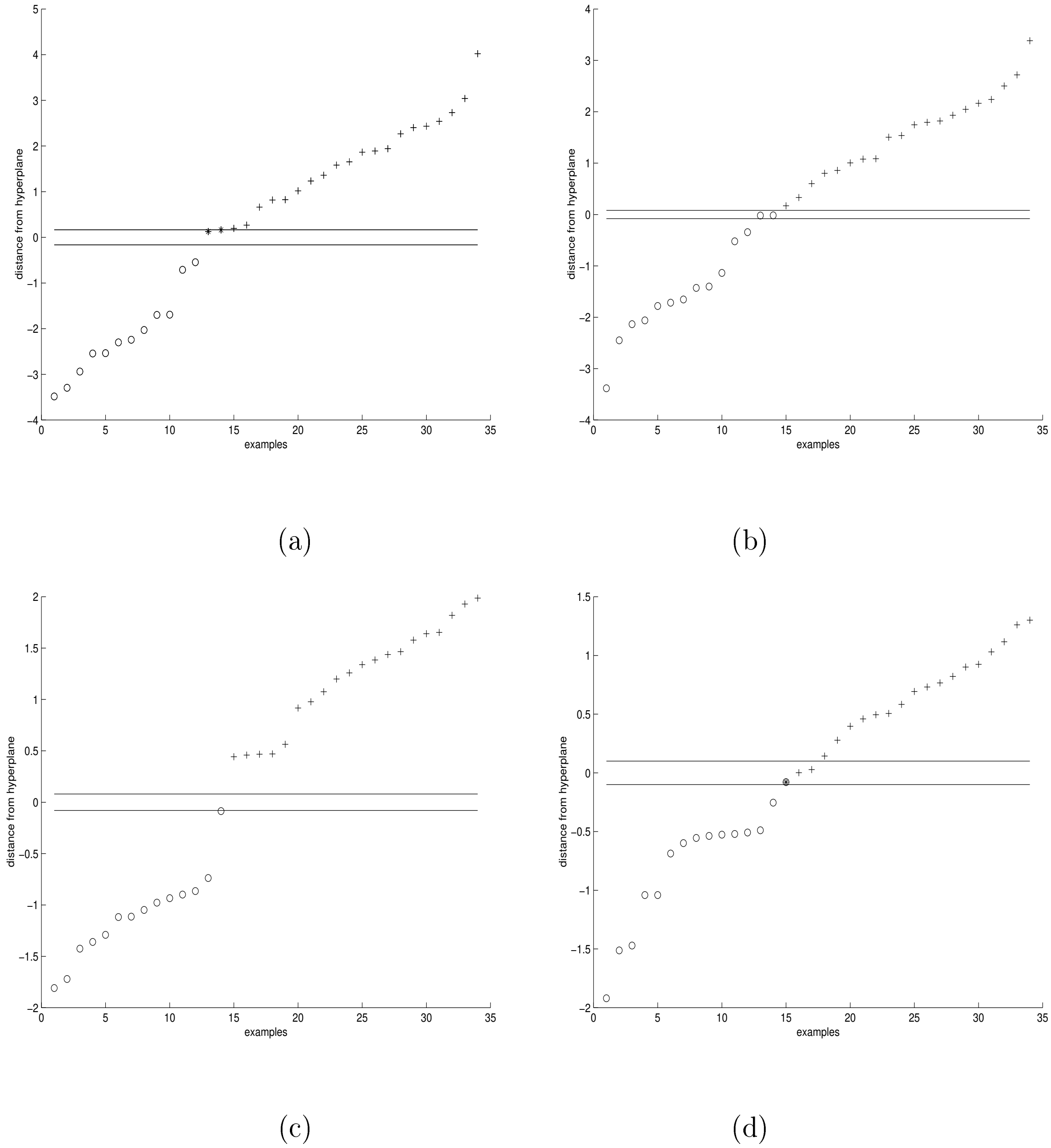
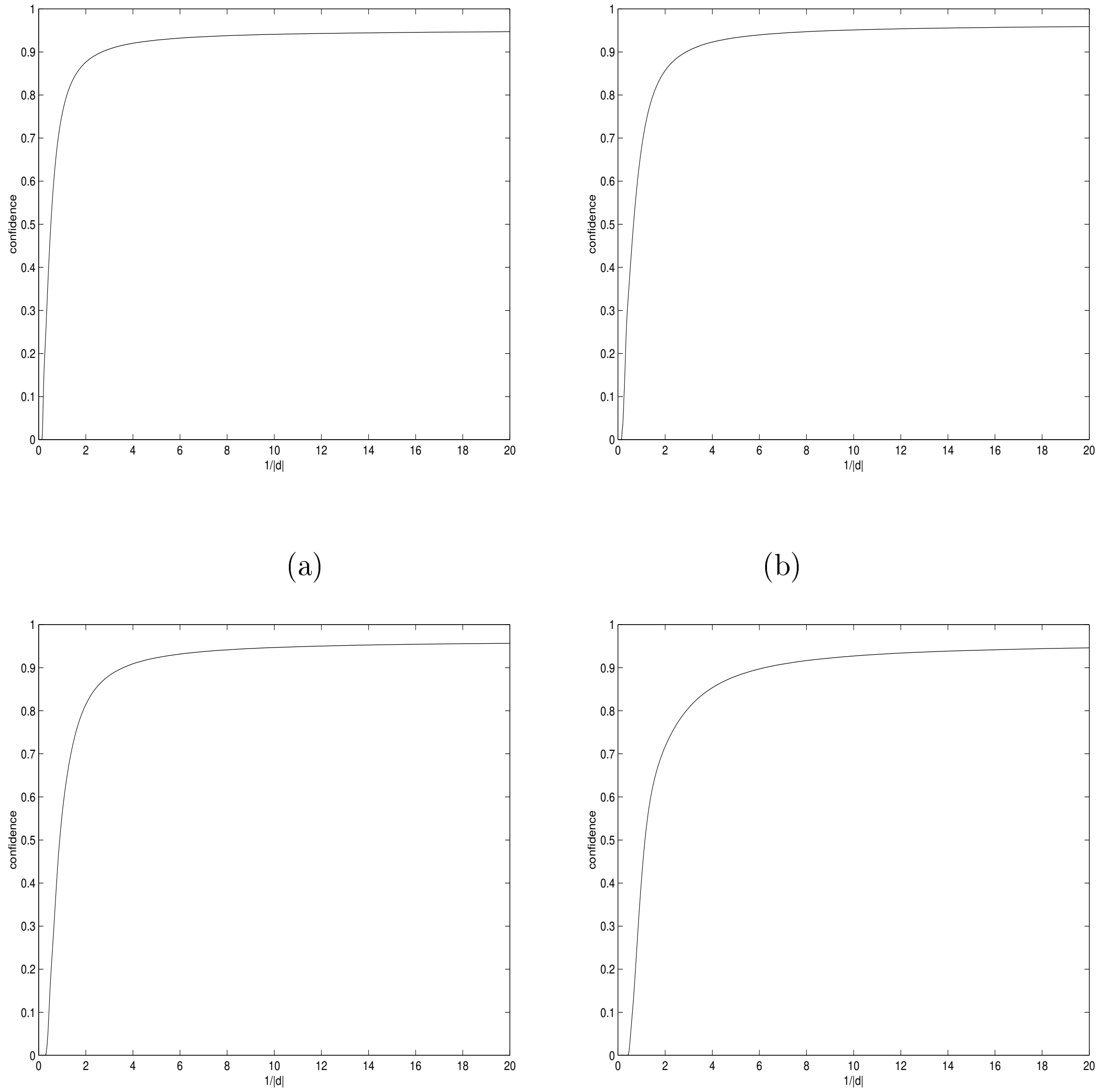


Figure 2: Plots of the distance from the hyperplane for test points (a) feature vector of 49 (b) feature vector of 99 (c) feature vector of 999 (d) feature vector of 7129. The + are for class ALL, the o for class AML, the $*$ are mistakes, and the line indicates the decision boundary.



(a) (b) (c) (d)

Figure 3: Plots of the confidence levels as a function of $1/|d|$ estimated from a leave-one-out procedure on the training data for (a) feature vector of 49 (b) feature vector of 99 (c) feature vector of 999 (d) feature vector of 7129.

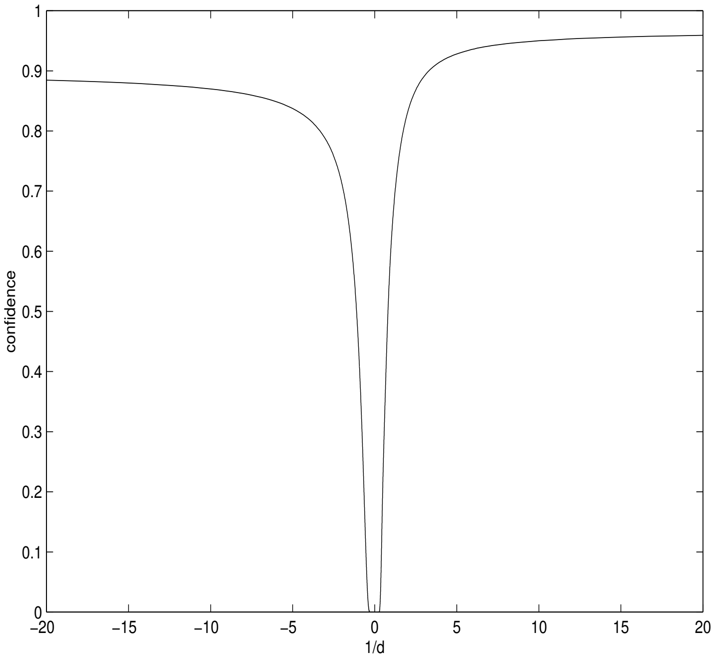


Figure 4: Plot of the confidence levels as a function of $1/d$ estimated from a leave-one-out procedure on the training data for a feature vector of 999.