Seminar on Data mining and data integration in Bioinformatics

GENE SELECTION FOR CANCER CLASSIFICATION USING SUPPORT VECTOR MACHINES

Authors:
Isabelle Guyon, Jeson Weston
Stephen Barnhill
Barnhill Bioinformatics, Savannah, Georgia
Vladimir Vapnik
AT&T Labs

Presented by: Nafisa Afrin Chowdhury

INTRODUCTION

- Micro Array devices produce huge amount of raw data
- Screen thousands of genes simultaneously using DNA micro-arrays .
- Determine whether those genes are active or silent in normal or cancerous tissue.
- An SVM classifier has been built using these DNA micro array as training data for genetic diagnosis as well as drug discovery.

What is DNA Micro Array?

- A **DNA microarray** is a multiplex technology used in molecular biology and in medicine. It consists of an arrayed series of thousands of microscopic spots of DNA oligonucleotides, called features, each containing picomoles (10^{-12} moles) of a specific DNA sequence.

What is DNA Micro Array?

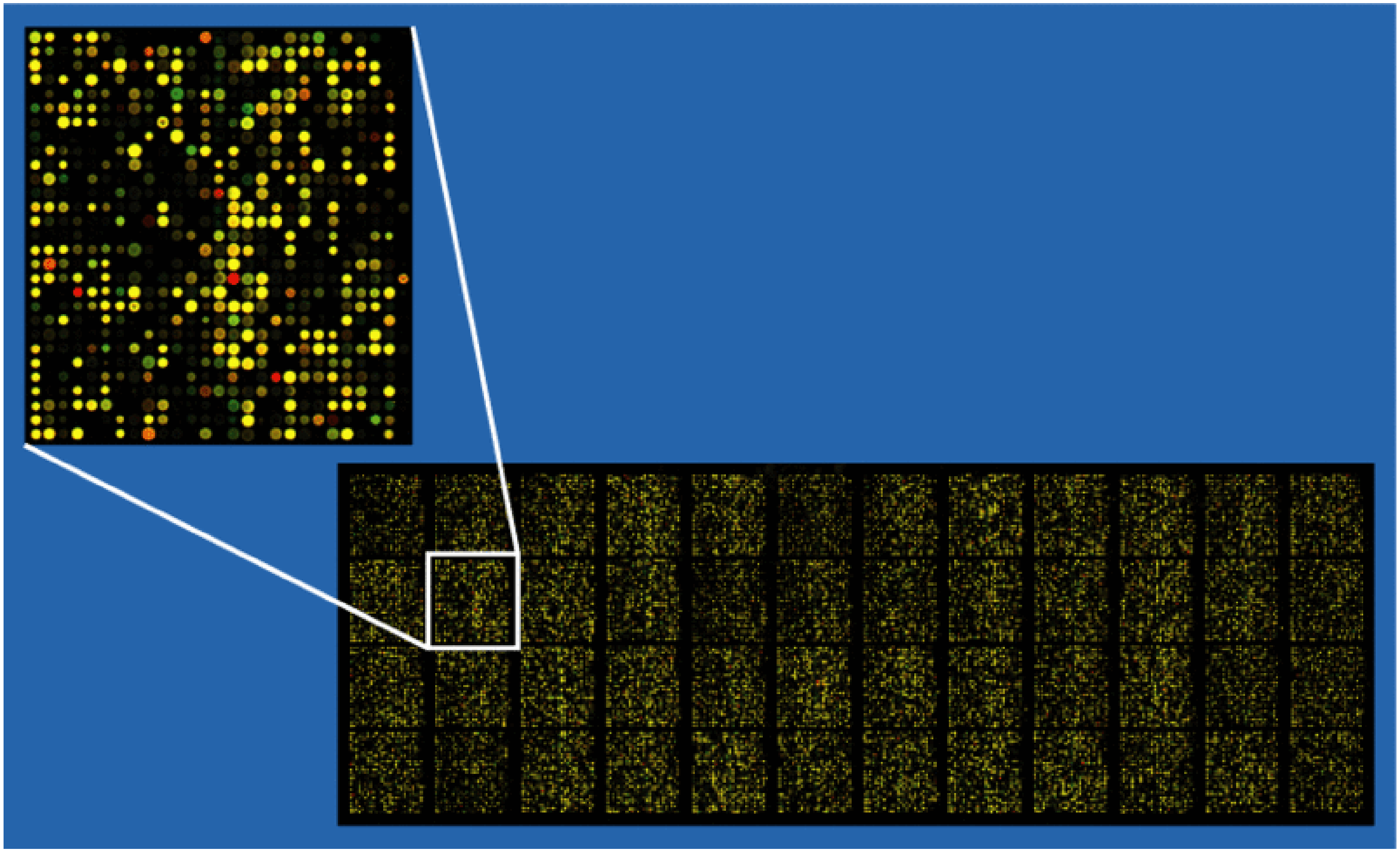

Fig 1: A DNA micro array sequence with 40,000 oligoneucliotides.

FINDING OUT THE SUSPECT GENES

- Previous works:
  - Unsupervised learning : clustering
  - Supervised learning : Classification of proteins
- Goal of this paper
  - Extracting a small subset of highly discriminant genes
  - Reducing overhead of medical diagnostic test of protein of serum

Problem Description

- Input -> vector of patterns (patients)
- Of n Components -> features (gene expression coefficients)
- Feature space F -> n dimensional
- Two-class classification problem
  - Positive (+)
  - Negative (-)
- A training set
  - set of number of patterns {x1, x2, x3…xk,….xl}
  - with known class labels {y1, y2 ,y3….yk, …yl}

PROBLEM DESCRIPTION CONT…

- A decision function D(x)
  - D(x) >0 => x∊ class(+)
  - D(x) <0 => x∊ class(-)
  - D(x) =0 => decision boundary.
- Decision functions are the simple weighted sums of the training patterns plus a bias are called linear discriminant functions:
- D(x) = w.x +b

SPACE DIMENSIONALITY REDUCTION

- The risk of “over fitting”
  - Number of features n is large (thousands of genes)
  - Number of training pattern are comparatively small (a few dozen patients)
  - Easy to find out a decision function D(x) that separates training data but performs poorly on test data.
- To overcome over fitting
  - Space dimensionality reduction
  - Feature selection like pruning
- Practical importance
  - Cost effectiveness
  - Easy to verify relevance of selected genes

FEATURE SELECTION

- Greedy algorithms -> feature ranking
  - A fixed number of top ranked features may be selected for further analysis or to design a classifier
  - Ranking to define a nested subsets of features $F1 \subset F2 \subset \cdot\cdot\cdot \subset F$, and select an optimum subset of features
- Several feature ranking algorithms
  - Feature ranking with correlation coefficients
  - Ranking criterion and classification
  - Feature ranking by sensitivity analysis
  - Recursive feature elimination

Feature ranking with correlation coefficients

- It is not possible to achieve an errorless separation with a single gene. Better results are obtained when increasing the number of genes.
- The coefficient used by previous paper (Golub 1999)
  - $w_i = (\mu_i (+) - \mu_i (-)) / (\sigma_i (+) + \sigma_i (-))$
    - μ_i = *mean*
    - σ_i = *Standard deviation of the gene expression values of gene I for all patients of class (+) and (-)*
  - Large negative w_i values indicate strong correlation with class (-)
- Each coefficient w_i is computed with information about a single feature (gene) and does not take into account mutual information between features.

RANKING CRITERION AND CLASSIFICATION

- The classification based on weighted voting
  - The features votes proportionally to their correlation coefficient.
  - $D(\mathbf{x}) = \mathbf{w} \cdot (\mathbf{x} - \boldsymbol{\mu})$
    - W is a vector of wi and $\mu = (\mu(+) + \mu(-))/2.$

FEATURE RANKING BY SENSITIVITY ANALYSIS

- For classification problems, the ideal objective function is the expected value of the error, that is the error rate computed on an infinite number of examples.
- For the purpose of training, this ideal objective is replaced by a cost function *J computed on training examples* only.
- Hence the idea to compute the change in cost function *DJ(i) caused by removing a given feature or, equivalently, by bringing its* weight to zero.
  - $DJ(i) = (1/2)\partial^2 J / \partial w_i^2 (Dw_i)2$
- The change in weight $Dwi = wi$ corresponds to removing feature i .
- To remove several features at a time
  - More computationally efficient
  - The method produces a feature subset ranking, as opposed to a feature ranking.
  - Feature subsets are nested $F1 \subset F2 \subset \cdots \subset F$.

Recursive Feature Elimination

- The criteria *DJ(i) or (wi)2* become very sub-optimal when it comes to removing several features at a time, which is necessary to obtain a small feature subset.
- This problem can be overcome by using the following iterative procedure that we call Recursive Feature Elimination:

1. Train the classifier (optimize the weights *wi with respect to J)*.
2. Compute the ranking criterion for all features (*DJ(i) or (wi)2)*.
3. Remove the feature with smallest ranking criterion.

SUPPORT VECTOR MACHINE

- A state-of-art classification technique
- Although SVMs handle non-linear decision boundaries of arbitrary complexity, this paper limits, to linear SVMs because of the nature of the data sets under investigation.
- Linear SVMs are particular linear discriminant classifiers.
- If the training data set is linearly separable, a linear SVM is a maximum margin classifier .
- The decision boundary (a straight line in the case of a two-dimensional separation) is positioned to leave the largest possible margin on either side. A particularity of SVMs is that the weights *wi of the decision function D(**x**)* are a function only of a small subset of the training examples, called "support vectors".

ALGORITHM- SVM TRAIN

1. Inputs: Training examples {x1, *x2, . . . xk, . . . x} and class labels {y1, y2, . . . yk, . . . y}.*
2. Minimize over *ak :*
3. $J = (1/2)\sum_{hk} \mathrm{y_h}\, \mathrm{y_k}\, a_h\, a_k\, (\mathrm{x_h} \cdot \mathrm{x_k} + \lambda\delta_{\mathrm{hk}}) - \sum_k a_k$
4. subject to: $0 \leq ak \leq C$ *and* $\sum_k a_k y_k = 0$
5. Outputs: Parameters a_k .
   - δ_{hk} is the Kronecker symbol (δ_{hk}=1 if h =k and 0 otherwise), and λ and C are positive constants (soft margin parameters).
   - The soft margin parameters ensure convergence even when the problem is non-linearly separable or poorly conditioned.
6. The resulting decision function of an input vector x is:
   - $D(x) = w \cdot x + b$
   - With $\mathrm{w} = \sum_k a_k\, y_k\, \mathrm{x_k}$ and $\mathrm{b} = \mathrm{y_k} - \mathrm{w} \cdot \mathrm{x_k}$

ALGORITHM- SVM RFE

- *Inputs:*
  - Training examples
  - X0 = [x1, *x2, . . . xk, . . . x]T*
- Class labels
  - y = [y1, *y2, . . . yk, . . . y]T*
- Initialize:
  - Subset of surviving features
    - s = [1, *2, . . . n]*
- Feature ranked list *r = [ ]*
- Repeat until s = []
- Restrict training examples to good feature indices
  - $X = X_0(:, s)$

ALGORITHM- SVM RFE CONT…

- Train the classifier
  - a = *SVM-train(X,* ***y)***
- Compute the weight vector of dimension length(**s)**
  - w **=**$\sum_k a_k\, y_k\, \mathrm{x_k}$
- Compute the ranking criteria
  - $\mathrm{c_i} = (w_i)^2$, *for all i*
- Find the feature with smallest ranking criterion
  - f = *argmin(**c)***
- Update feature ranked list
  - r = [s*(f), r]*
- Eliminate the feature with smallest ranking criterion
  - s = s*(1: f − 1, f + 1: length(s))*
- Output: Feature ranked list **r.**

DATA SET DESCRIPTION

- Two different datasets
  - Cancer patients with two different types of leukemia: Acute Lymphoid Leukemia (ALL) and Acute Myeloid Leukemia (AML).
    - Training set contains 38 samples (27 ALL and 11 AML) from bone marrow specimen
    - Test set contains 34 samples (20 ALL and 11 AML)
    - All samples have 7129 features
  - Cancerous or normal colon tissues.
    - Total 62 sample tissues : 22 normal and 40 colon cancerious
    - Each have 2000 gene expression values (features)
    - Among all half of the samples used in training and the rest is in test

EXPERIMENT

- Data Preprocessing
  - From each gene expression value the mean has been subtracted and divided by its SD
- Feature Elimination
  - Recursive Feature Elimination
  - Obtain nested subsets of genes of increasing informative density.
- Designing classifier
  - A linear SVM classifier
  - Baseline method
- Compare results

RESULTS

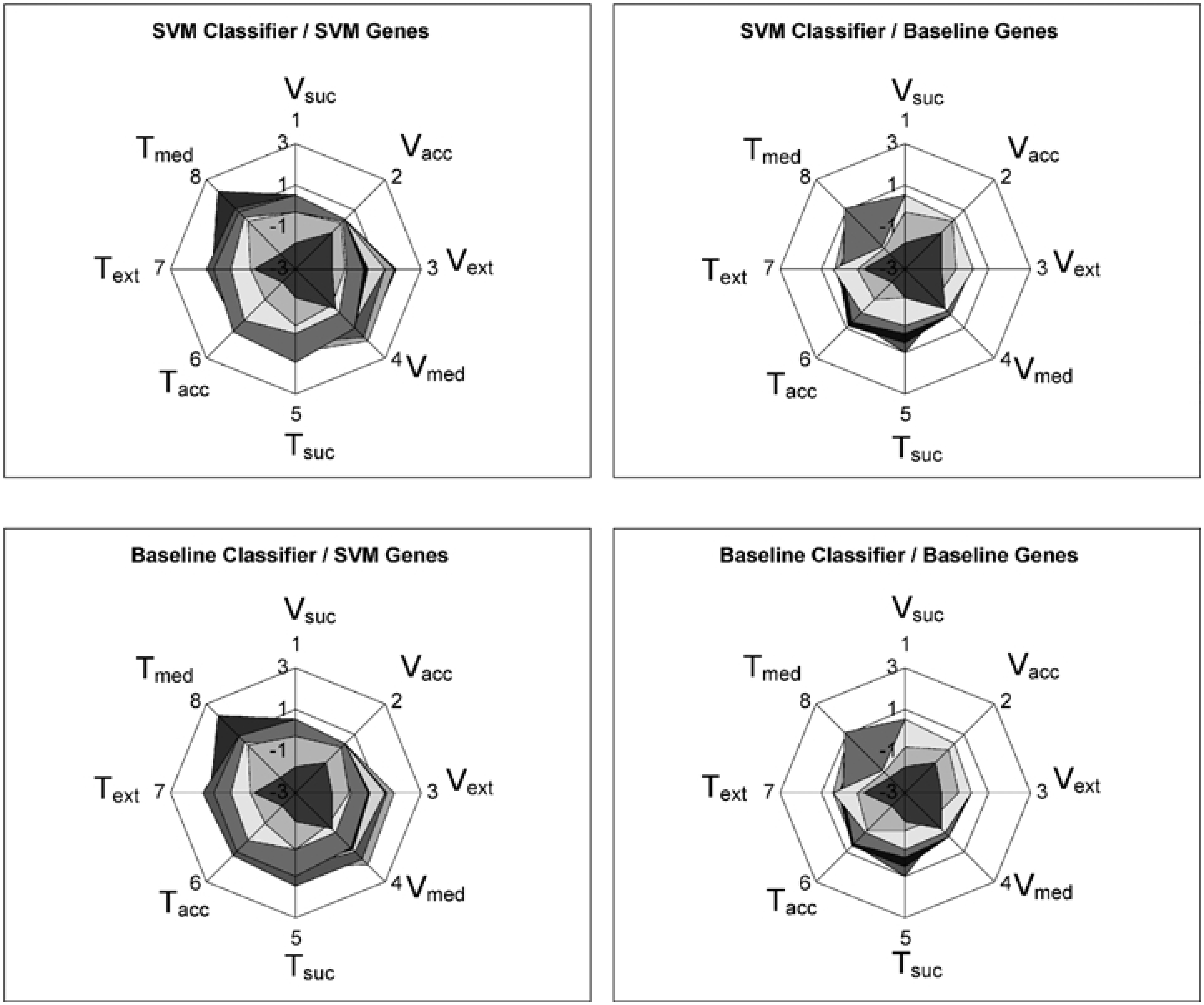

Fig 2: The features selected matter more than the classifier used

RESULTS

- Whether SVM or baseline classifier, SVM genes are better with 84.1% confidence based on test error rate and 99.2% based on the test rejection rate.
- SVMs select relevant genes

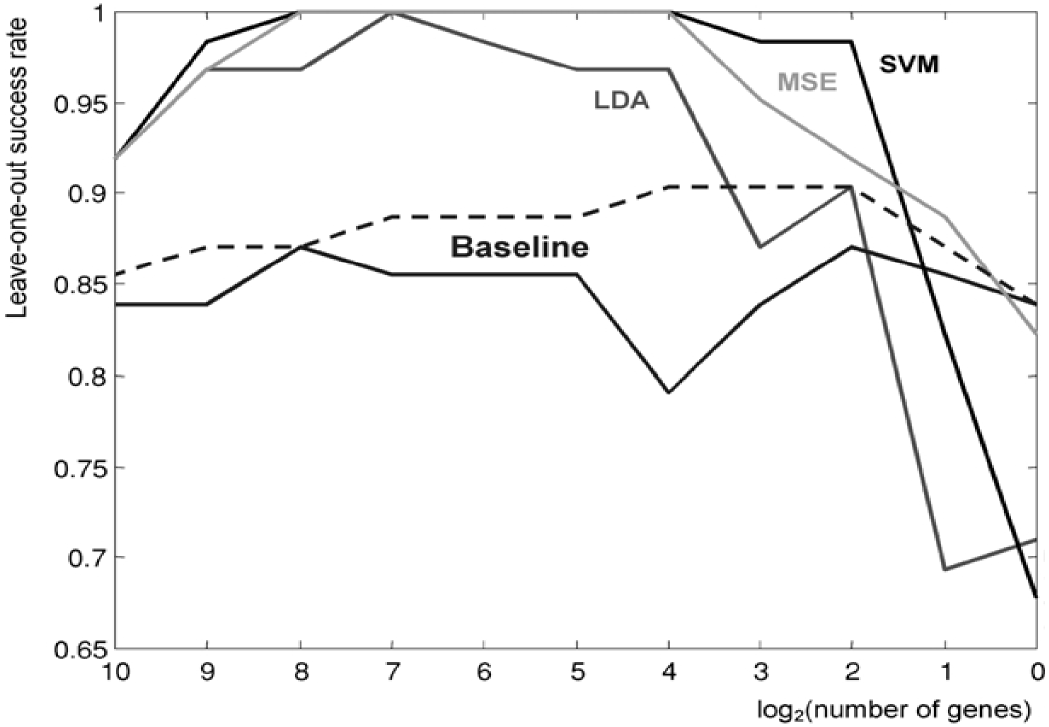

Fig 3. Comparison of feature (gene) selection methods (Colon cancer data).

RESULTS

- SVM RFE shows better performance than all the other methods; selects down the 4 genes mostly suspected for cancer.
- The first gene that is related to tissue composition and mentions "smooth muscle" in its description ranks 5 for the baseline method, 4 for LDA, 1 for MSE and only 41 for SVM.
- In patients with leukemia our method discovered 2 genes that yield zero leave one-out error, while 64 genes are necessary for the baseline method to get the best result (one leave-one-out error).
- In the colon cancer database, using only 4 genes our method is 98% accurate, while the baseline method is only 86% accurate.

DRAWBACKS WITH SVM COMPUTATION

- The fastest methods of feature selection are correlation methods: for the data sets under study, several thousands of genes can be ranked in about one second by the baseline method (Golub, 1999) with a Pentium processor.
- Training algorithms such as SVMs or Pseudo-inverse/MSE require first the computation of the (l,l) matrix H of all the scalar products between the training patterns. The computation of H increases linearly with the number of features (genes) and quadratically with the number of training patterns.
- The training time is of the order of the time required to invert matrix *H*.
- Matlab implementation of SVM RFE on a Pentium processor returns a gene ranking in about 15 minutes for the entire Colon dataset (2000 genes, 62 patients) and 3 hours on the Leukemia dataset (7129 genes, 72patients).

Then Why do we need computationally expensive SVM?

- A simple geometric interpretation of the feature ranking criterion based on the magnitude of the weights: for slopes larger than 45 degrees, the preferred feature is x1, otherwise it is x2.

- Feature *x1 separates perfectly all examples but has a* higher variance. We think of feature *x1 as the relevant feature (a cancer-related gene) and as* feature *x2 as the irrelevant feature (a tissue composition related gene): most examples are* very well separated according to tissue composition, but one valuable outlier contradicts this general trend.

- The baseline classifier (Golub, 1999) prefers feature *x2. But the SVM* prefers feature *x1*.

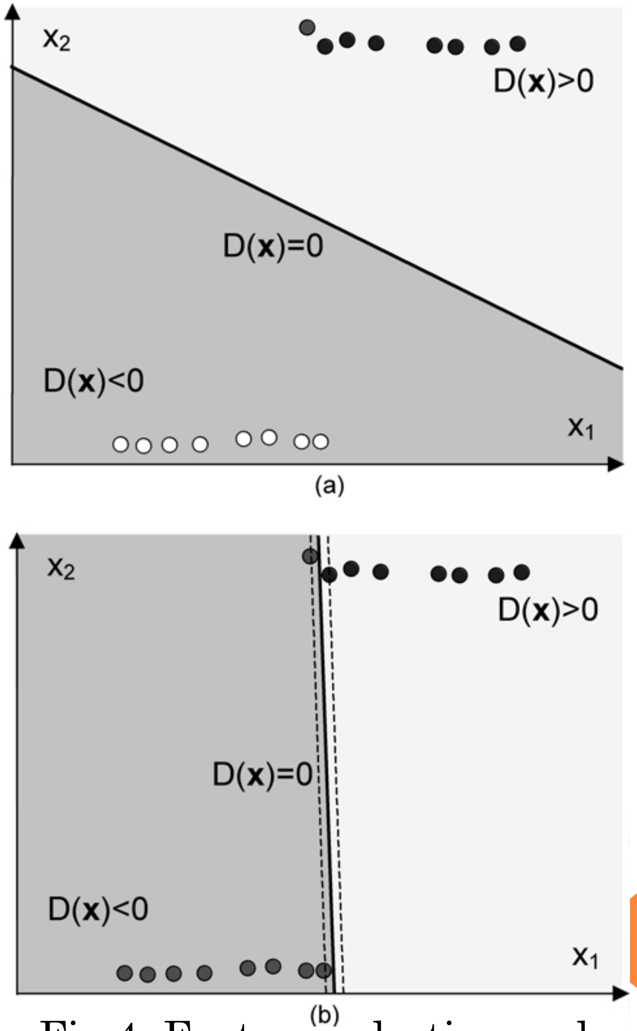

Fig 4: Feature selection and support vectors.

CONCLUSION

- SVM can easily deal with a large number of features (thousands of genes) and a small number of training patterns (dozens of patients). They integrate pattern selection and feature selection in a single consistent framework.
- The top ranked genes found by SVM all have a plausible relation to cancer.
- So SVM has both qualitatively and quantitatively advantage in comparison with other gene selection methods.

I LIKED THE PAPER

- Although it was a long one but I liked it because
  - Well organized
  - Contains explanation for every formula
  - Shows proper reason of choosing methods
  - Gather more applicable knowledge about SVM known from the ML class

QUESTIONS
?