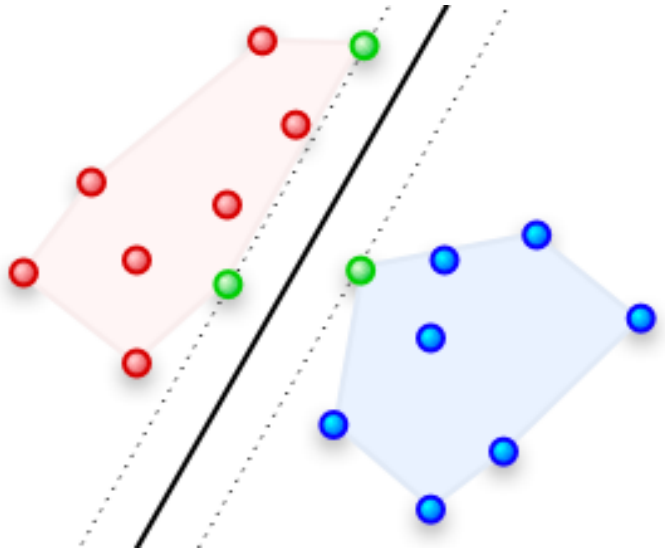
SVM Classification in μ-Arrays



„SVM classification and validation of cancer tissue samples using microarray expression data“

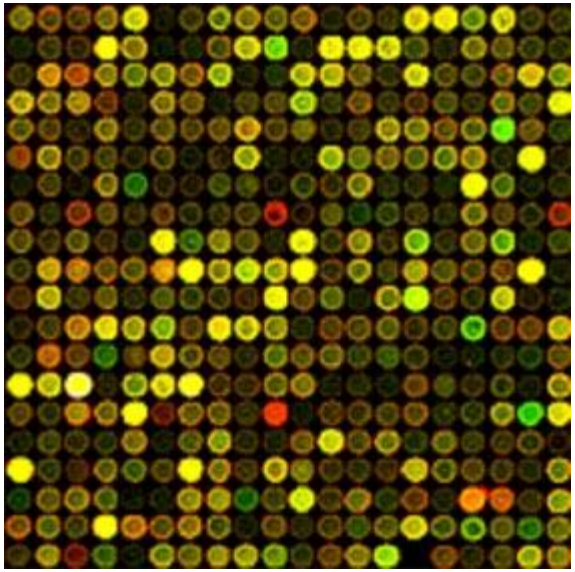
Furey et al, 2000

Special Topics in Bioinformatics, SS10

A. Regl, 7055213

What is it all about?

- ... its about classification of Micro-Array data

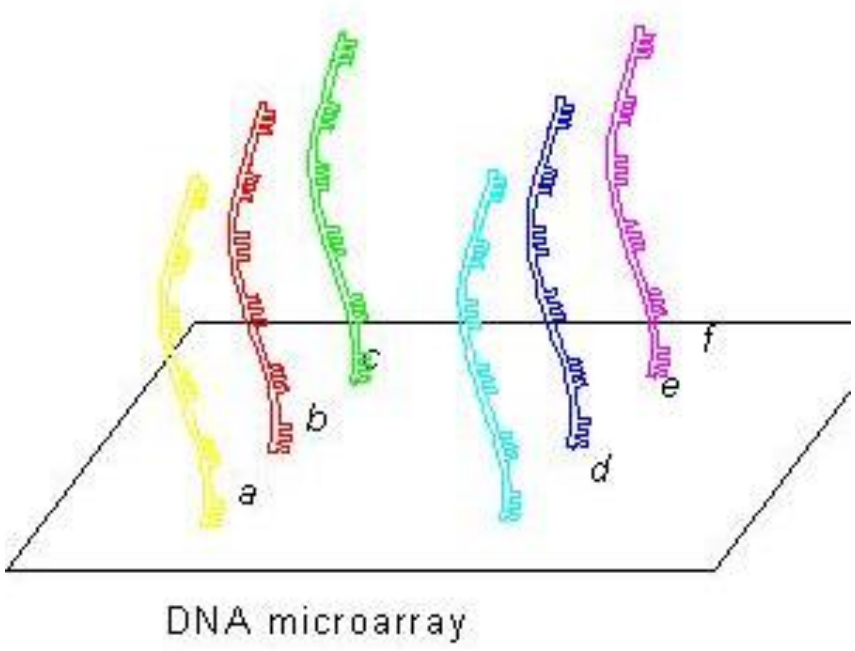


?

- we want to extract relevant gene expression differences.
- The paper says „*yes, also SVMs can do the job*“.

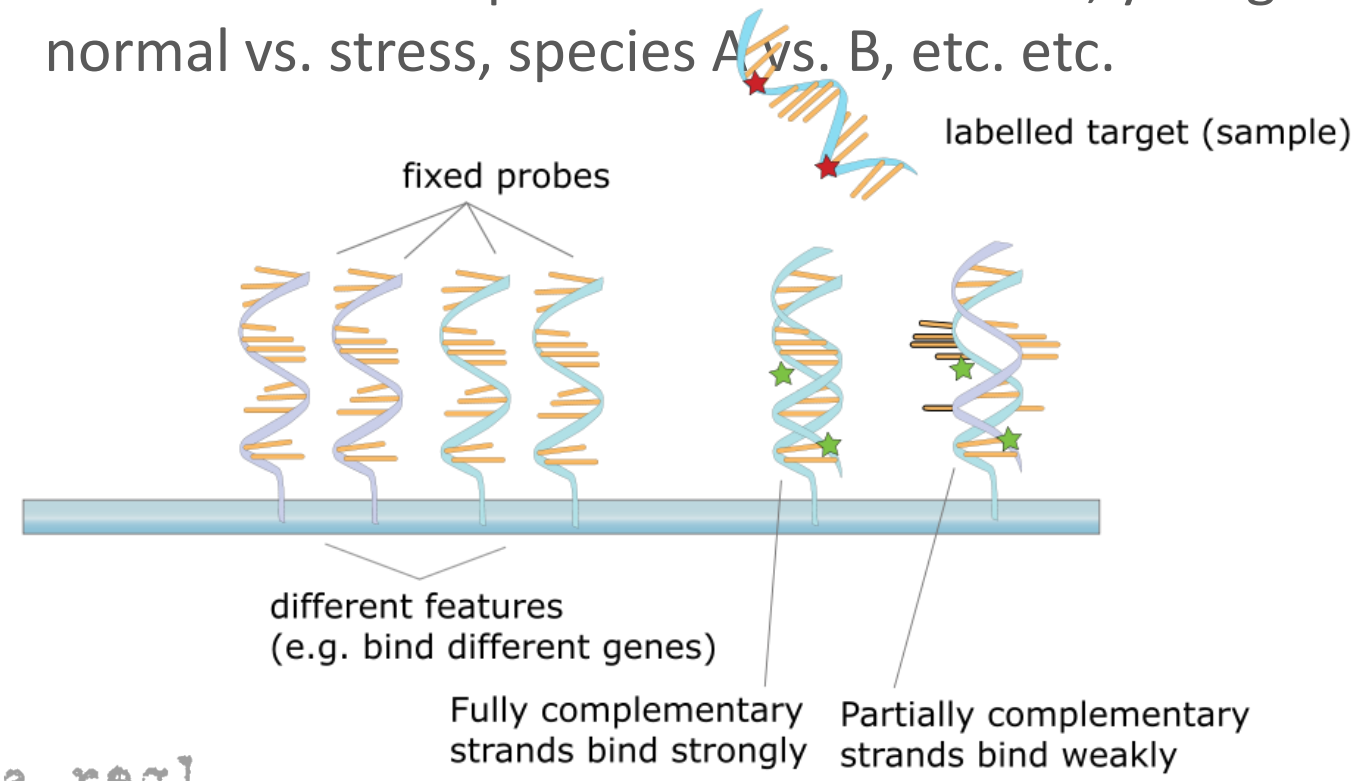
What are DNA Micro Arrays ? (1)

- „Array“ = array of short DNA strands (= probes)
- Sequences are taken from a genome, e.g. human
- genes are represented by ≥1 probes (= probe set)



What are DNA Microarrays ? (2)

- We take a cell, extract her mRNAs and transcribe into cDNA
- Then red/green markers are applied to the two samples
- Hybridizing with the probes will give us expression levels
- Now we can compare: normal vs. cancer, young vs. old, normal vs. stress, species A vs. B, etc. etc.



How do we compare?

- We have a classifaction problem now:
- „Do we have a cancer tissue? Yes or no?“
  „Are certain genes co-expressed? Yes or no?“
  „Do we have a certain disease? Yes or no?
  ...

- Any classifaction algorithm will do:
- Clustering methods
  Self-organizing maps
  Correlation methods, and...
- ... Support Vector Machines

Properties of SVMs

- Supervised classification
  (Training set with known classes has to be provided)

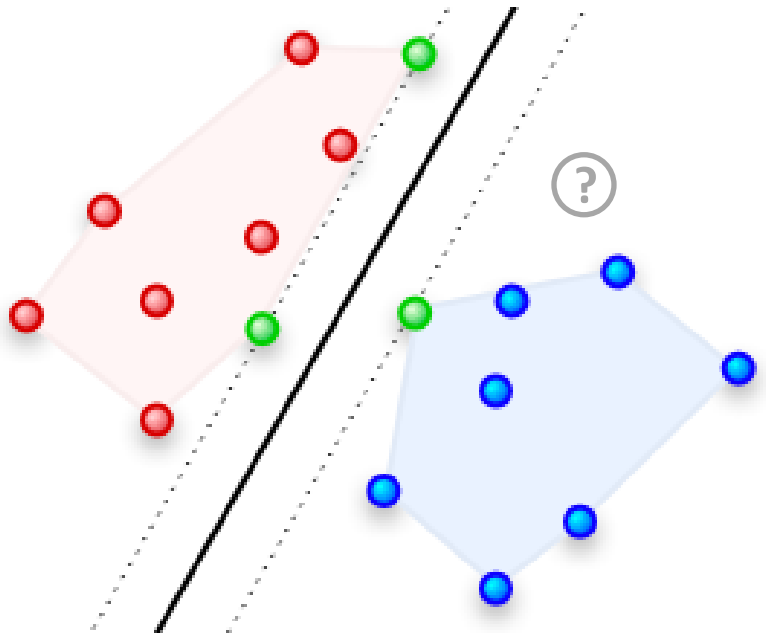
- Robust for large number of features
  (in contrast to other methods)

- Robust for noisy data
  (but: not generally!)

- Well defined for 2 classes only (called +1 and -1)
  (Extensions to n classes are avaible, but not straightforward)

What are SVMs?

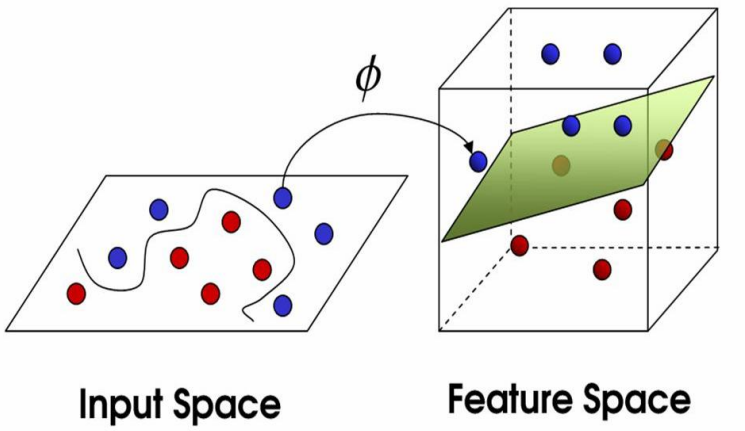
- What is given?
  A set of points in n-dimensional space, labelled with 2 classes
- What do we look for?
  Which (n-1)-dimensional hyperplane will result in maximal separation?



- Only a small subset of the points defines the plane! („Support Vectors“)
- Classification:
  On which side of the hyperplane is the unknown point?

Nonlinear hyperplanes

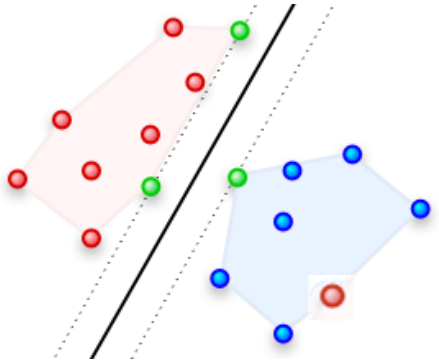
- What if the classes are not linearly separable?
- Try it in higher dimensions!
- Nonlinear mapping Φ from input space to feature space
- Linear separation plane in feature space corresponds to nonlinear separation plane in input space



- Φ = „kernel function“
- Kernel Matrix: $K_{ij} = <\Phi(\mathbf{x}^i),\Phi(\mathbf{x}^j)>$
- Generalized kernel functions: $\boldsymbol{K}_{ij} = \boldsymbol{K}(\mathbf{x}^i,\mathbf{x}^j)$
- Popular kernel functions:
  Dot Product: $<\mathbf{x}^i,\mathbf{x}^j>$
  Polynomial: $(<\mathbf{x}^i,\mathbf{x}^j>+1)^d$
  Gaussian: $\exp(-\|\mathbf{x}^i-\mathbf{x}^j\|/\sigma^2)$

Other intricacies

- Training errors are not tolerated
  (can lead to grossly false hyperplanes, see example)



- The answer: „soft-margin“ classifiers
- Or: modifiers for the kernel diagonal in the training phase
  $K \leftarrow K + \lambda \mathbf{1}$, ($\lambda$ to be tuned)
  or $K \leftarrow K + \lambda D$, with $D_{ii} = d^+$ or d^- (e.g. to reflect class size)
- Many more tweaks available, but not used in this paper.
- If you can‘t get enough: see „BI 2“.

Feature Selection

	Genes (j = 1 … n)	Y (class labels)
Expression vectors (i = 1 … m)	x_j^i	y_i
mean (+)	$\mu^+_1 \ldots \mu^+_j \ldots \mu^+_n$	
sd (+)	$\sigma^+_1 \ldots \sigma^+_j \ldots \sigma^+_n$	
mean (-)	$\mu^-_1 \ldots \mu^-_j \ldots \mu^-_n$	
sd (-)	$\sigma^-_1 \ldots \sigma^-_j \ldots \sigma^-_n$	
Feature quality	$F(x_j) = \lvert(\mu^+_j - \mu^-_j)/(\sigma^+_j + \sigma^-_j)\rvert$	

- Having many features can be nasty
- Idea: take relevant features only (to make life for the classifactor easier)
- In this paper: rank features according to relative expression level difference („Take only genes that show some action“)
- How many? The paper is very clear here: *„... some number of the top features are extracted ...“*

Data Sets

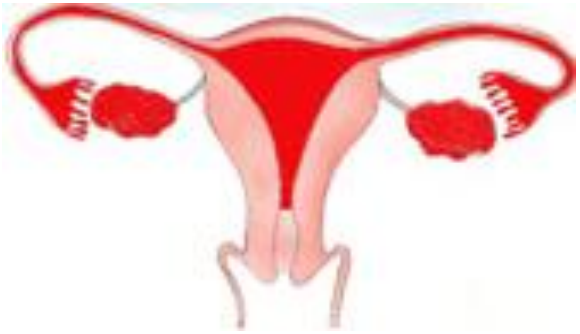
- Previously unpublished:
  Ovarian tissues

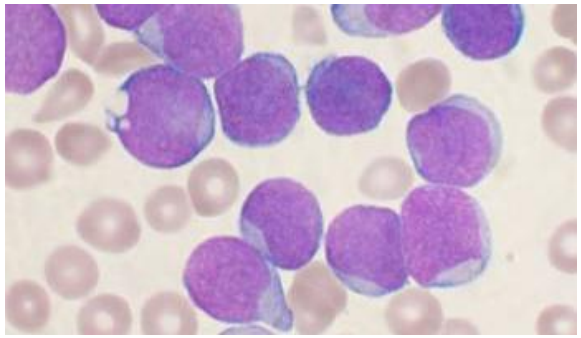
- Previously published:

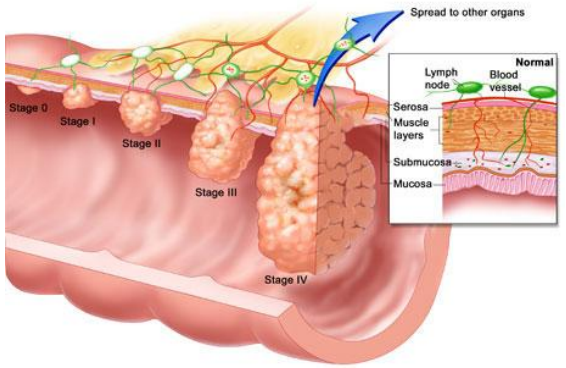
  Blood samples

  Colon

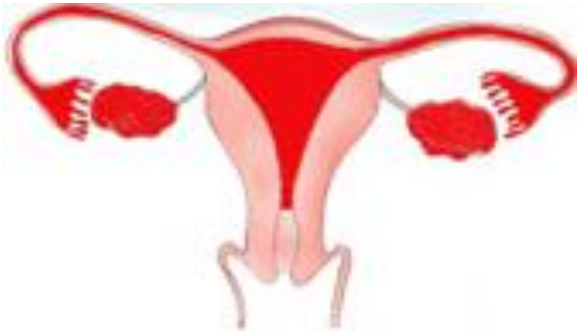
- Common question:
  „Cancer - yes or no?“







Ovarian Dataset



- 98000 DNA clones
  31 tissue samples
  2 classes (cancer or not)
- Leave-one-out cross validation
- Experimenting with parametes:
  Diagonal factor (λ): 0, 2, 5, 10
  Feature selection: 25, 50, 100, 500, 1000, 98000
  Kernels: dot-product, polynomial and RBF

- One misclassification detected
  one „outlier“ removed.

Results for Ovarian Dataset

λ	nF	FP	FN	TP	TN	FP+FN	TP+TN	
0	25	5	4	10	12	9	22	71%
2	25	5	2	12	12	7	24	77%
5	25	4	2	12	13	6	25	81%
10	25	4	2	12	13	6	25	81%
0	50	4	2	12	13	6	25	81%
2	50	3	2	12	14	5	26	84%
5	50	3	2	12	14	5	26	84%
10	50	3	2	12	14	5	26	84%
0	100	4	3	11	13	7	24	77%
2	100	5	3	11	12	8	23	74%
5	100	5	3	11	12	8	23	74%
10	100	5	3	11	12	8	23	74%
0	98000	17	0	14	0	17	14	45%
2	98000	9	2	12	8	11	20	65%
5	98000	7	3	11	10	10	21	68%
10	98000	5	3	11	12	8	23	74%

some λ and low nr. of features give good results

avoid using all features available

results are somehow disappointing

a.regl

Survivors of the Feature Selection Process

- Remember: DNA sequences could be genes - or not.
- Lets look at the 10 top-ranked sequences.
  Are they biologically significant genes?

  1, 2, 3: not readable
  4,5: poly-A-tailssequence
  6: no relation to cancer
  7: ferritin-H
  8, 9: homologs to cancer-library ESTs
  10: related to white blood cells in cancer tissues

directly related to cancer

- A look at some „bottom-rankers“:
  there are cancer related genes also
- *„... additional effort is needed to develop ways of identifiying meaningful features ...“*

indirectly related to cancer

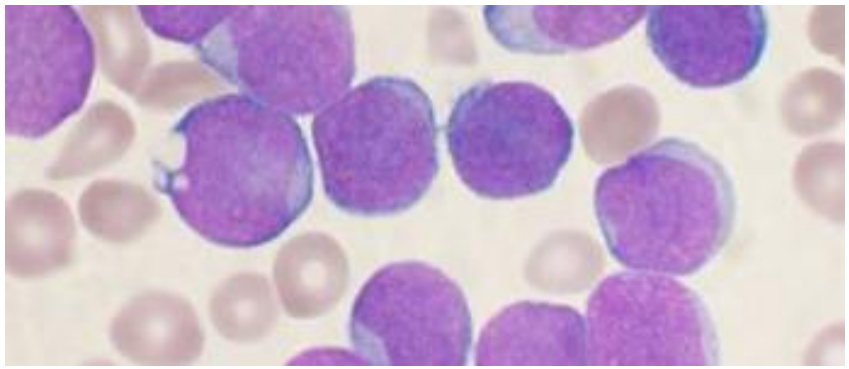
Results for Leukemia and Colon tumor Dataset

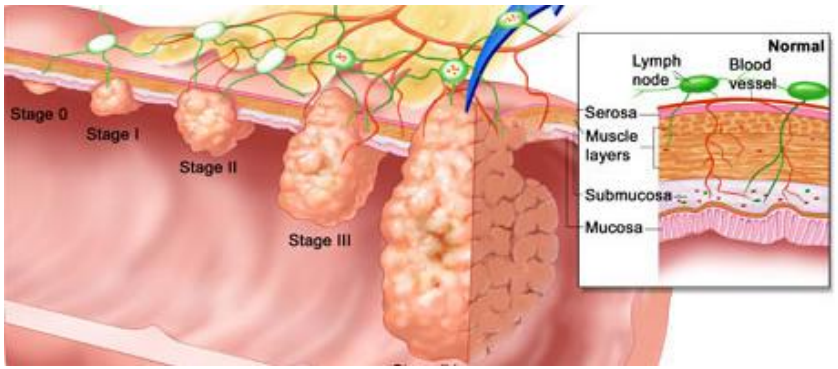
Leukemia

- 7192 genes from 72 patients
- normalized Affy scores
- Original (SOM):
  29 OK, 5 „dont know“
- SVM:
  30-32 OK (including the 29)
  (slightly better in special cases)

Colon tumor

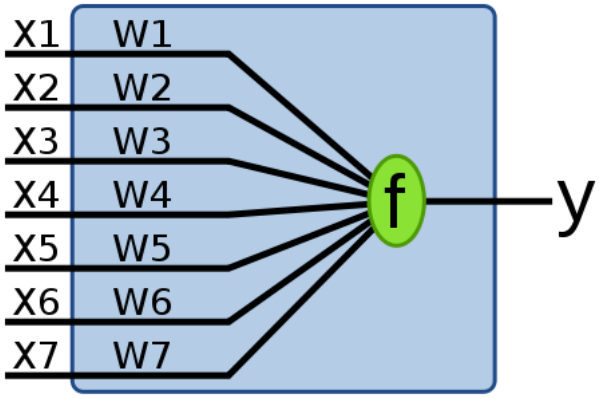
- 6500 genes from 40+22 pat.
- no normalization
- Original (clustering method):
  (35OK+3F) + (19OK+5F)
- SVM:
  (56OK+6F)





And what about Perceptron-like Classifiers?

- Perceptron by Rosenblatt (1958!)
- Simple algorithm, updates its weight vector with each „mistake“ ($\mathbf{w}^{i+1} = \mathbf{w}^i + y^i\mathbf{x}^i$)
- Modification required for non-perfect linear separation



- Results for our data sets are ...
- ... comparable to SVM!

Conclusions

- SVM does the job, but not really superior to other methods
- Even simple perceptrons are equally good

- BUT: datasets contain too few examples to draw a hard conclusion.
- With more examples, more complex kernels could be necessary, and then SVMs could outperform other methods.

- And: the paper dates from 2000, only a short time after SVMs and Microarrays had been available

Any Questions?