SVM applications

Johan Suykens

KU Leuven, ESAT-STADIUS
Kasteelpark Arenberg 10
B-3001 Leuven (Heverlee), Belgium
Email: johan.suykens@esat.kuleuven.be
http://www.esat.kuleuven.be/stadius

Lecture 4

Contents

- Microarray data analysis
- Textmining applications
- Electricity load forecasting

Bioinformatics and Microarray Data Analysis

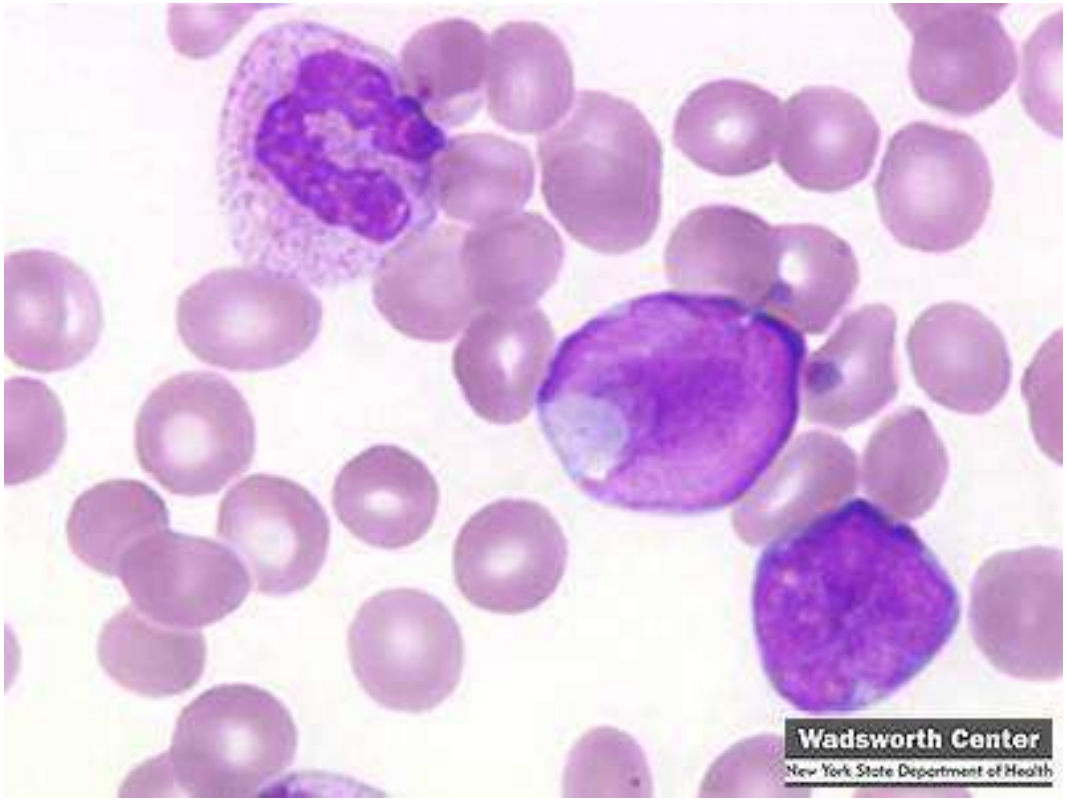
- Technological advances in **molecular biology** led to the development of **microarrays**, determining the expression levels of thousands of **genes** simultaneously. Disordered expression of genes lies at the origin of the behavior of tumors. Measurement of it can be very helpful to predict or to model the clinical behavior of malignant processes.

- Microarray data are represented by an expression matrix from which the rows and the columns respectively represent the gene expression profiles and the expression patterns of a patient.

- Data sets generated by microarrays consist of a large number of gene expression levels for each patient and a relatively small number of patients (different classes of tumors).

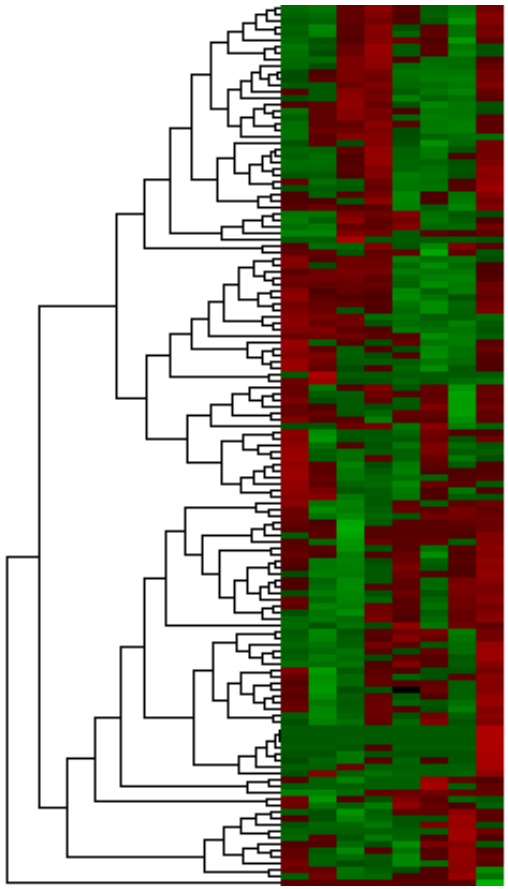
- SVMs are capable of learning and generalizing these microarray data well despite of the **high dimensionality**.

Microarray data (1)



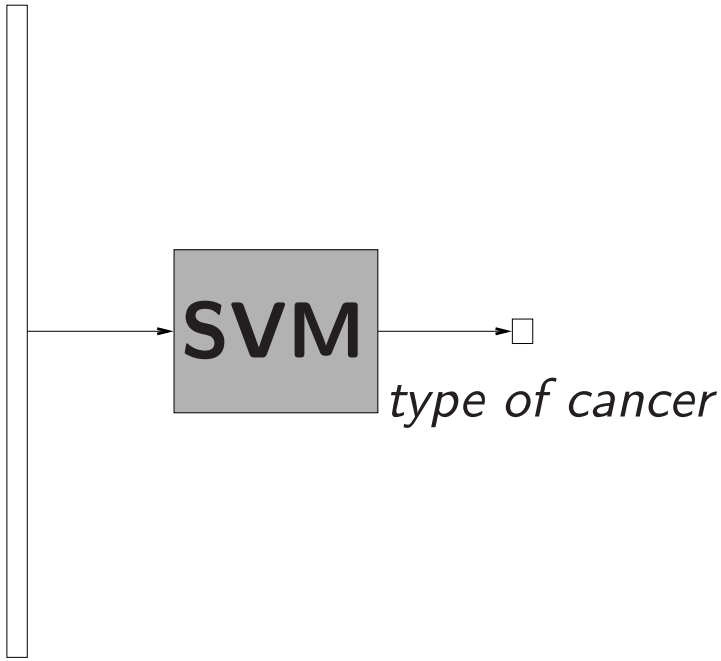
Some malignant lymphoblasts in the periferal blood of a patient suffering from acute lymphoblastic leukemia (ALL). These cells are typically selected for microarrays.

Microarray data (2)



Microarray data are represented by an expression matrix from which the rows and the columns respectively represent the gene expression profiles and the expression patterns of a patient.

Microarray data (3)



MIT Leukemia dataset (7,129 gene expressions):
for classical neural networks such as MLPs one first has to do a dimensionality reduction of the input space. SVMs on the other hand are able to learn and generalize on input vectors with 7000 genes.

Microarray data (4)

- **Leukemia data set - classification problem:** Bone marrow or peripheral blood samples are taken from 72 patients with either acute myeloid leukemia (AML) or acute lymphoblastic leukemia (ALL). The data is split into a training set consisting of 38 samples of which 27 are ALL and 11 are AML, and the test set of 24 samples, 20 ALL and 14 AML. The dataset contains expression levels for 7129 human genes produced by Affymetrix high-density oligonucleotide microarrays.

- **Pre-processing:** The scores in the dataset represent the intensity of gene expression after being re-scaled to make overall intensities for each chip equivalent. Following the methods in [Golub et al., 1999], normalization of these scores should be performed for each gene by subtracting the mean and dividing by the standard deviation of the expression values for that gene.

  [Golub T.R. *et al.*, Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, *Science* 1999; 286: 531-537].

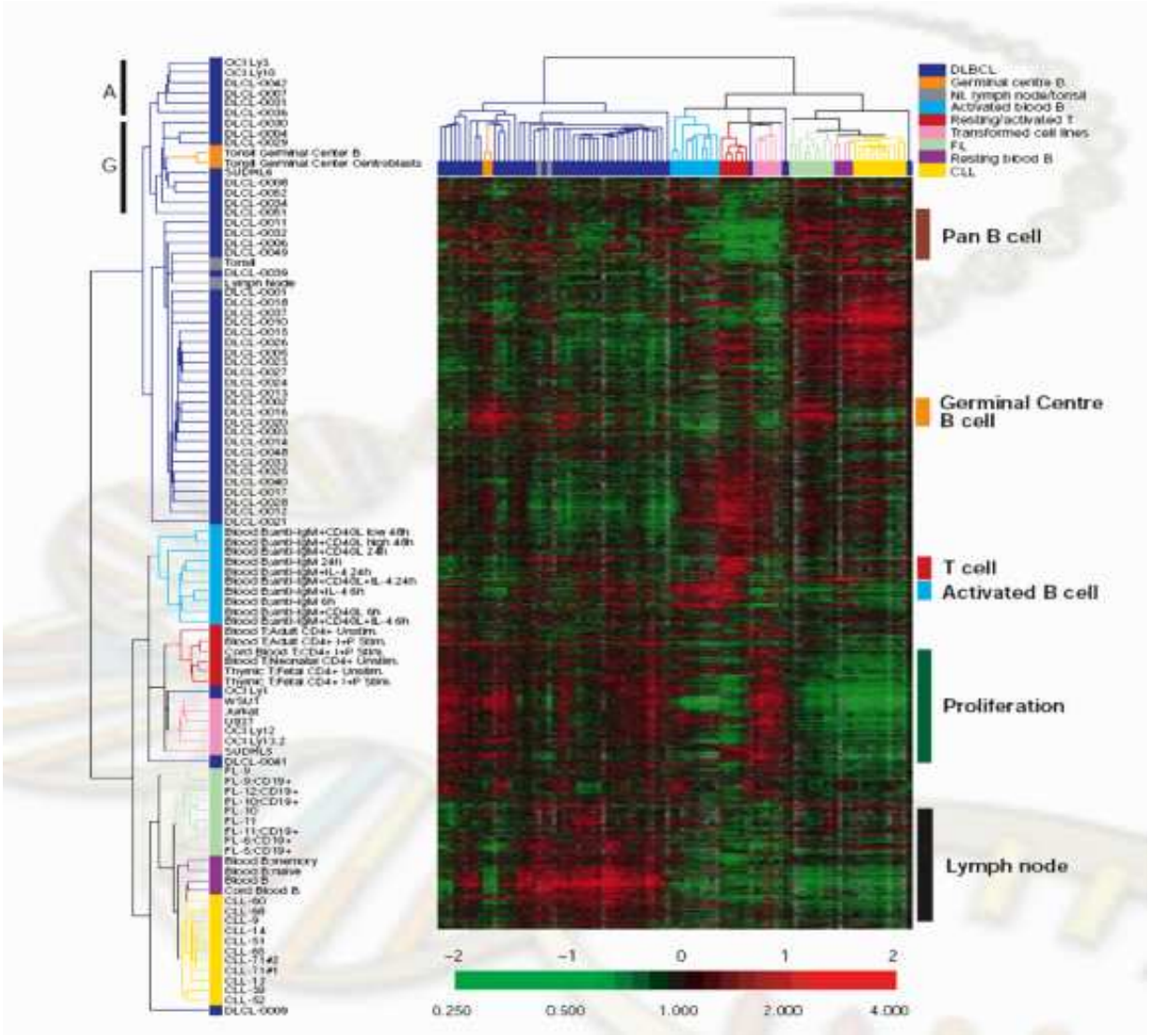
Microarray data (5)

- **Breast cancer data set:** A training data set containing 78 primary breast cancers was selected: 34 from patients who developed distant metastases within 5 years, 44 from patients who continued to be disease-free after a period of at least 5 years. All sporadic patients were lymph node negative, and under 55 years of age at diagnosis.

- From each patient, 5 μg total RNA was isolated from snap-frozen tumor material and used to derive complementary RNA (cRNA). A reference cRNA pool was made by pooling equal amounts of cRNA from each of the sporadic carcinomas. Two hybridizations were carried out for each tumor using a fluorescent dye reversal technique on microarrays containing 24481 human genes synthesized by inkjet technology.

  [van 't Veer L.J. *et al.*, Gene Expression Profiling Predicts Clinical Outcome of Breast Cancer, *Nature* 2002; 415: 530-536]

Microarray data (6)



Some genomes numbers

Group	Species	Genes	Genome (Mbase)
Phages	Bacteriophage MS2	4	0.003560
Viruses	HIV Type 2	9	0.009671
Bacteria	Haemophilus influenzae (1995)	1760	1.83
Archaea	Methanococcus jannaschii	1735	1.74
Fungi	Saccaromyces cerevisiae (yeast) (1996)	5800	12.1
Protoctista	Oxytricha similis	12000	600
Arthropoda	Drosophila melanogaster (fruit fly) (2000)	12000	165
Nematoda	Caenorhabdiis elegans (Round worm)(1998)	14000	100
Mollusca	Loligo Pealii	35000	2700
Plantae	Arabidopsis thaliana (Mustard cress)(2000)	25000	70-145
Chordata	Homo Sapiens	30000	3000

Estimated 265-350 genes are required for 'life'.

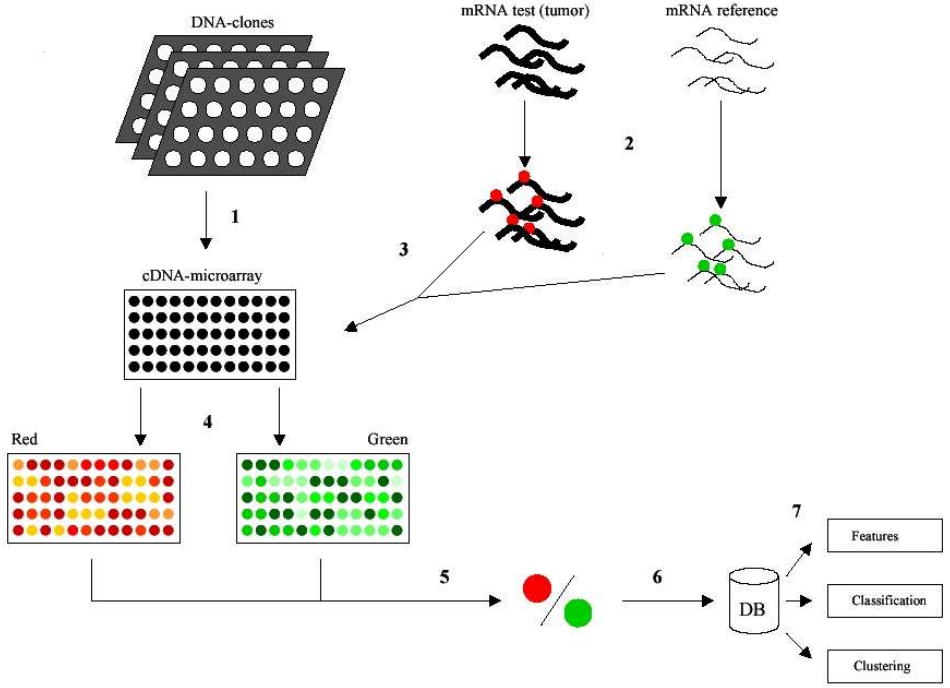
Types of microarrays (1)

- **cDNA-microarrays** (or spotted arrays) (**relative** measurements) consist of ten thousands of known cDNAs mechanically deposited onto modified glass slides by contact or ink jet printing.

- **Oligonucleotide microarrays** (or DNA chips, Affymetrix) (**absolute** measurements) are produced by the synthesis of oligonucleotides on silicium chips.

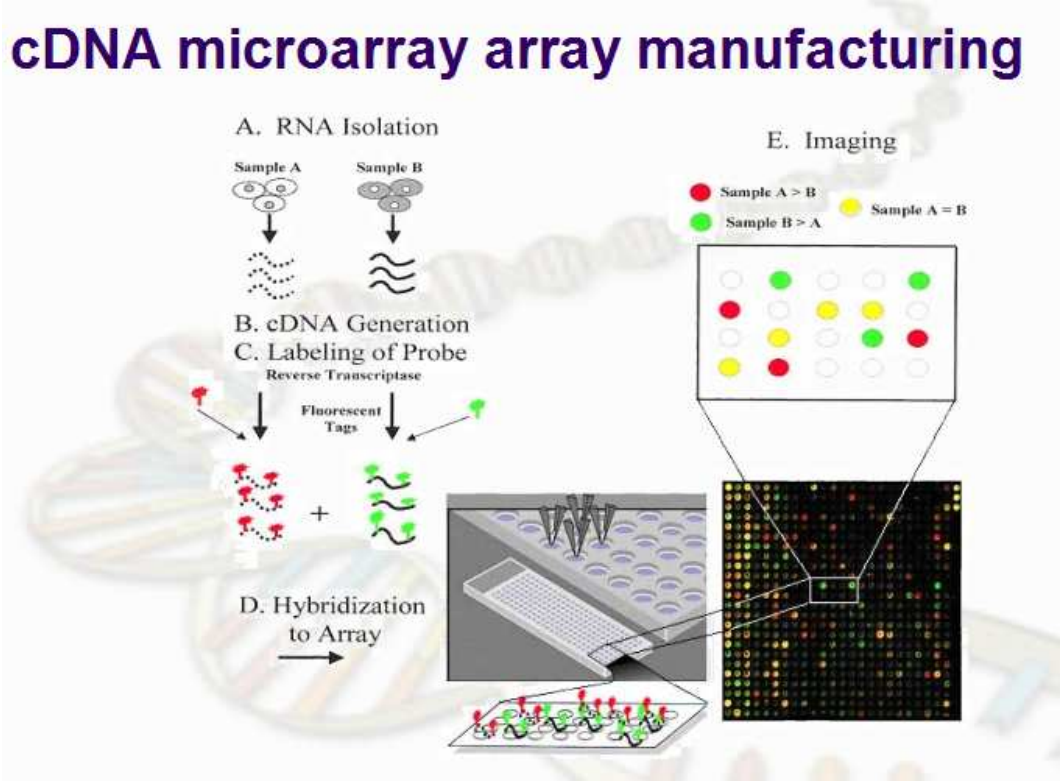
- Both techniques have their own characteristics, which have to be taken into consideration when **pre-processing** the data.

Types of microarrays (2)



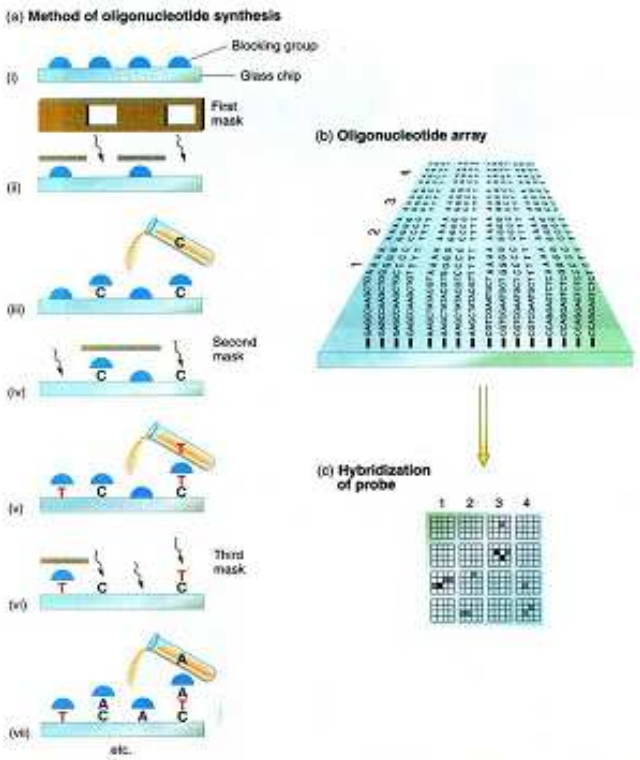
Schematic overview of cDNA-microarray: (1) Spotting of the pre-synthesized DNA-probes on the glass slide; (2) Labeling of the total mRNA of the test sample (tumor - red) and reference sample (green); (3) Pooling of the two samples and hybridization; (4) Read-out of the red and green intensities; (5) Calculation of the relative expression levels (intensity in the red channel / intensity in the green channel); (6) Storage in a database; (7) Data mining. [De Smet et al., 2001]

Types of microarrays (3)



cDNA-microarrays (spotted arrays) (relative measurements, differential hybridization): glass slides on which cDNA is deposited.

Types of microarrays (4)



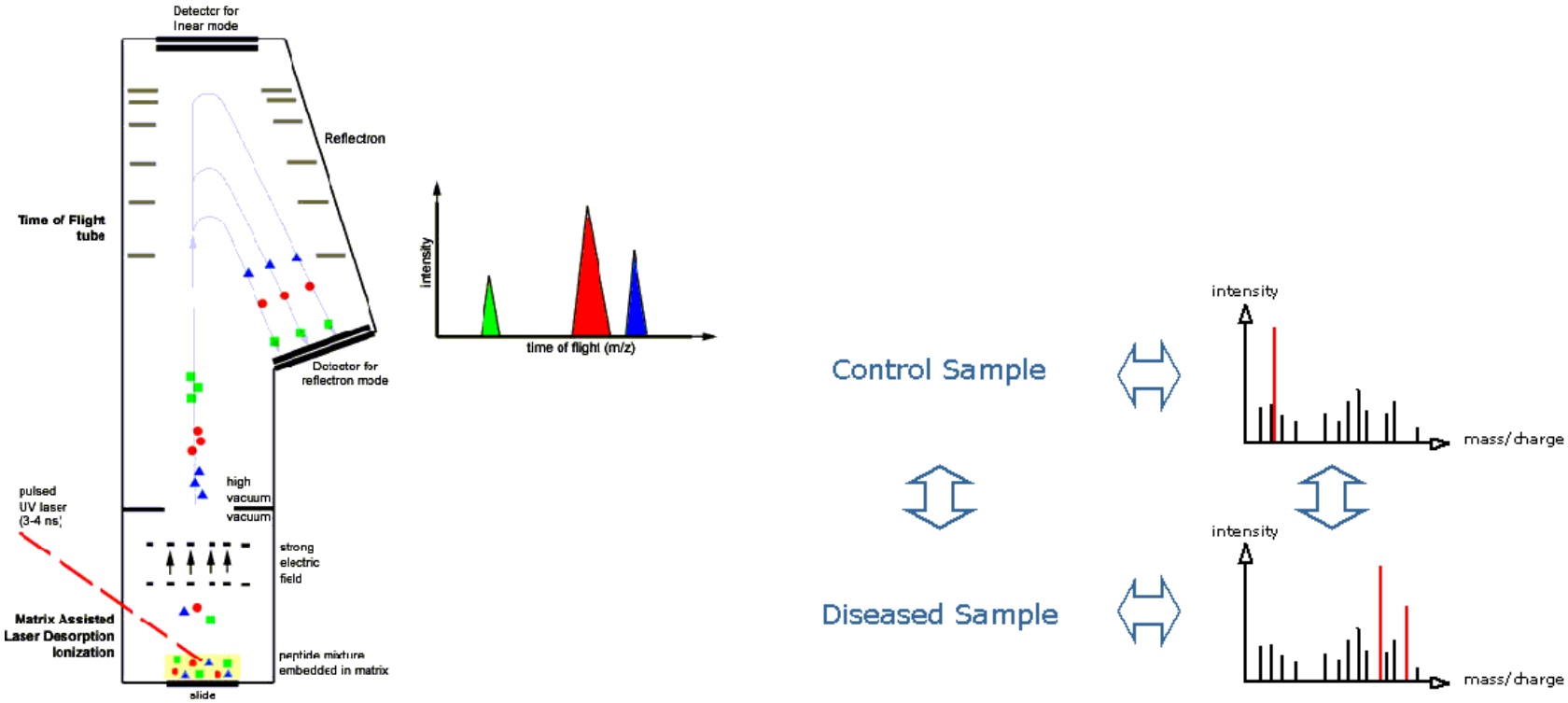
- clones are covalently bonded on chip
- DNA frames vertically on chip each representative of one gene
- mRNA of test (red tag) and control (green tag) samples
- samples are hybridized to array
- hybridized array scanned by red/green fluorescency

http://www.bio.davidson.edu/courses/genomics/chip/chip.html

Oligonucleotide microarrays (DNA chips, Affymetrix) (absolute measurements): produced by the synthesis of oligonucleotides on silicium chips.

Proteomics

MALDI-TOF mass spectrometer



Mass Spectrometry: measure the molecular masses of molecules or molecule fragments: mass analysis of complex organic mixtures, identification of proteins and peptides.

Structural proteomics: X-ray crystallography and NMR spectroscopy (high-throughput determination of protein structures)

Case study 1: Performance of clinical predictions (1)

- Main goal: Make **predictions** about the clinical information (e.g. diagnosis, prognosis, therapy response, . . .) of individual patients based on microarray measurements (possibly supplemented with other clinical data)

- Secondary goal: Finding most relevant genes for the classification (new insights for **drug design**)

- Data: few samples (20-30), large dimensions (5,000-100,000)

- Training set: Collection of patients for which is known to which class they belong. Those are the patients for whom for example the stage determination, histopathological diagnosis, prognosis, therapy response, etc. is known.

Case study 1: Performance of clinical predictions (2)

- Inputs (= genes) are ranked based on (Golub et al., 1999), using the scores from the known samples only. A number of the top features are extracted. These are used to train the SVM and classify the unknown sample. Linear kernel with leave-one-out crossvalidation performs well.

- Examples that have been consistently misclassified in all tests are identified. These examples can then be investigated by a biologist, and if it is determined that the original label is incorrect, a correction is made, and the process is repeated. Making clinical predictions for individual patients from microarray data seems to be possible, but larger-scale systematic experiments must be conducted.

  [S. Mukherjee, P. Tamayo, J.P. Mesirov, D. Slonim, A. Verri, T. Poggio. Support vector machine classification of microarray data, A.I. Memo 1677, MIT Artificial Intelligence Laboratory, 1998.] [T.S. Furey, N. Duffy, N. Cristianini, D. Bednarski, M. Schummer, D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics, 16(10): 906-914, 2000].

Case study 2: Performance of biological predictions (1)

- Goal: **gene function prediction**: Making predictions about the function of **uncharacterized genes** and the contribution of genes to the carcinogenesis.

- Discriminate between the **members and non-members** of a given **functional class**:

  [Brown et al., 2000] begins with defining a set of genes that have a common function, genes coding for ribosomal proteins or genes coding for components of the proteasome. In addition, a separate set of genes that are known not to be members of the functional class is specified. Using the training set, an SVM learns to discriminate between the members and non-members of a given functional class based on expression data.

- **Unbalanced** problem: each class contains few genes compared to the total number of genes (many negative examples).

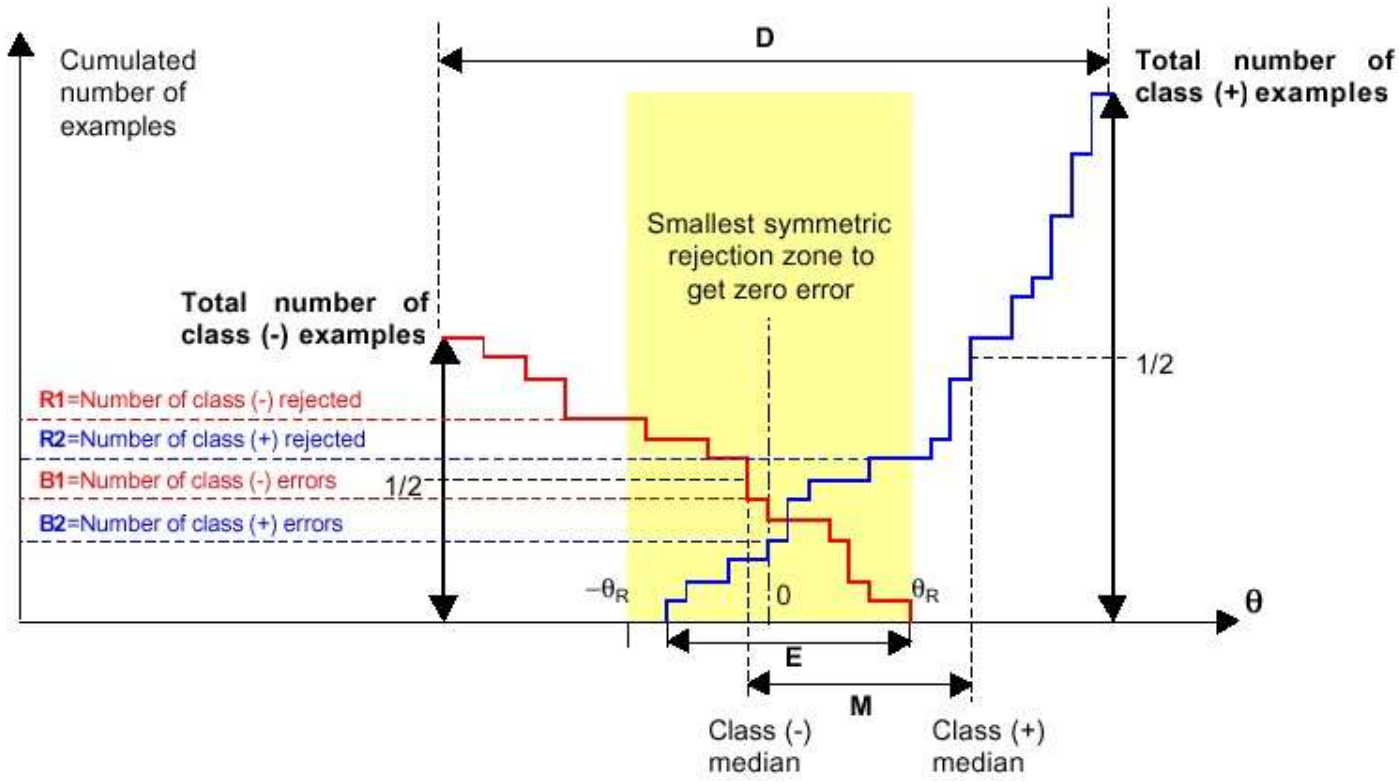
Case study 2: Performance of biological predictions (2)

- SVMs with different kernel functions (linear, polynomial and Gaussian kernels) are tried out. A comparison with other methods like Parzen windows, Fisher linear discriminant and decision trees, is conducted.

- The results show that for all classes **SVM with Gaussian kernels** performs best. It is possible to improve classification by using only a carefully selected part of the genes.

  [M.P.S. Brown, W.N. Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, Jr.M. Ares, D. Haussler, Knowledge-based analysis of microarray gene expression data by using support vector machines, *Proc. Natl. Acad. Sci. USA*, 97: 262-267, 2000].

Case study 2: Performance of biological predictions (3)



The red and blue curves represent example distributions of two classes (class (-) and class (+)). Red: Number of examples of class (-) whose decision function value is larger than or equal to θ. Blue: Number of examples of class (+) whose decision function is smaller than θ [Guyon et al., 2002].

SVM classifier for unbalanced data

Introduce a different weight per class in the SVM classifier objective function:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{N} v_i \xi_i$$

where

$$\begin{cases} v_i = \dfrac{N}{2N_+} & \text{if } y_i = 1; \\ v_i = \dfrac{N}{2N_-} & \text{if } y_i = -1. \end{cases}$$

In this way problems with unbalanced data can also be solved.

Case study 3: Discovery of relevant groups of genes (1)

- Searching for **relations between gene expressions and class labels**: not all gene expressions are correlated to the different diagnostic or biological classes. The intention is to find gene expressions or groups of gene expressions that are correlated to the different diagnostic classes.

- It allows to gain insight in the molecular biology underlying carcinogenesis. The selected genes could open up new perspectives for finding **drug targets** and for **finding tumor markers**.

- By using the score introduced by [Golub et al., 1999] for **ranking and selecting the genes**, there seems to be only **few biological relevance**.

Case study 3: Discovery of relevant groups of genes (2)

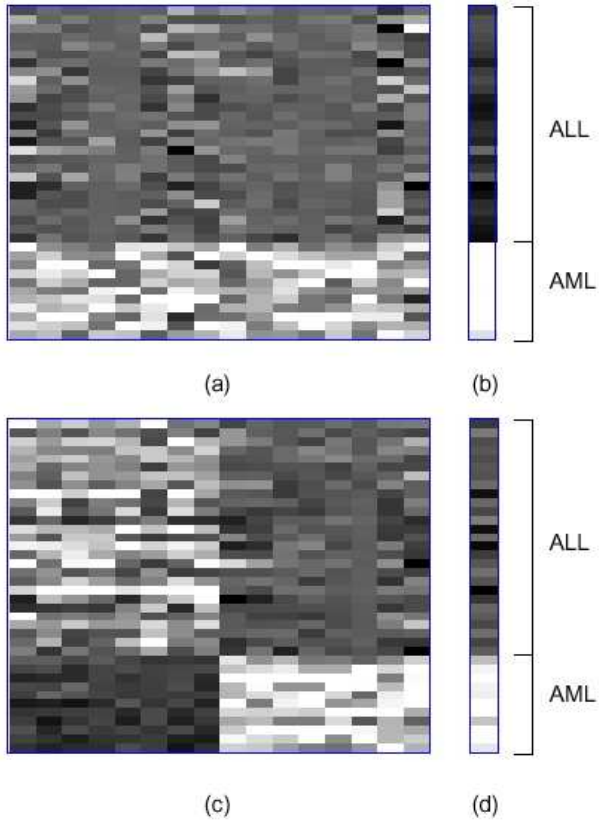
- Another gene selection procedure is proposed by [Guyon et al., 2002]: the SVM method of **Recursive Feature Elimination** (RFE). Genes are ranked based on their weight learned by an SVM. Genes are removed one by one (or by chunks), and one re-runs the SVM at each iteration. The top ranked genes found by SVMs all have a **plausible relation to cancer**.

- Feature ranking methods do not dictate the **optimal number of features** to be selected. An auxiliary **model selection criterion** must be used for that purpose.

- It should be noted that the proper way to conduct **leave-one-out cross-validation** for feature selection is to avoid using a fixed set of features selected with the whole training data set, because this induces a bias in the results.

  [I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, *Machine Learning,* 46(1/3): 389-422, 2002]

Case study 3: Discovery of relevant groups of genes (3)



Best sets of 16 genes (Leukemia data). In matrices (a) and (c), the columns represent different genes and the lines different patients from the training set. The 27 top lines are ALL patients and the 11 bottom lines are AML patients. The gray shading indicates gene expression: the lighter the stronger. (a)(b) [Golub, 1999]; (c)(d) [Guyon *et al.*, 2002].

Data fusion

- In order to further optimize clinical and biological predictions, it is possible to **integrate heterogeneous data**, like for example expert knowledge, anamnestical data, clinical data, histopathological data, image data (e.g. echos, scans), . . .

- **Different methods:**
  (i) The data vectors of the microarray expression data and the external data can be concatenated into a single data vector;
  (ii) **early fusion:** form a **combined kernel** by adding a microarray kernel and an external data kernel;
  (iii) **late fusion:** Also different SVMs can be trained and the models can then be added together (e.g. committee networks).

Data fusion - learning the kernel matrix (1)

- Consider $K = \sum_{i=1}^{p} \mu_i K_i \ (\mu_i \geq 0)$. Learn an optimal combination of μ_i together with SVM classifier by solving a single **convex problem** [Lanckriet et al., JMLR 2004]
- QP problem of SVM:

$$\max_{\alpha} \ 2\alpha^T 1 - \alpha^T \mathrm{diag}(y) K \mathrm{diag}(y) \alpha \ \text{ s.t. } \ 0 \leq \alpha \leq C, \alpha^T y = 0$$

is replaced by

$$\min_{\mu_i} \max_{\alpha} \ 2\alpha^T 1 - \alpha^T \mathrm{diag}(y) (\sum_{i=1}^{p} \mu_i K_i) \mathrm{diag}(y) \alpha$$

$$\text{s.t. } \ 0 \leq \alpha \leq C, \quad \alpha^T y = 0, \quad \mathrm{trace}(\sum_{i=1}^{p} \mu_i K_i) = c, \quad \sum_{i=1}^{p} \mu_i K_i \succeq 0.$$

Can be solved as semidefinite program (SDP problem) [Boyd & Vandenberghe, 2004] (LMI constraint for positive definite kernel)

Data fusion - learning the kernel matrix (2)

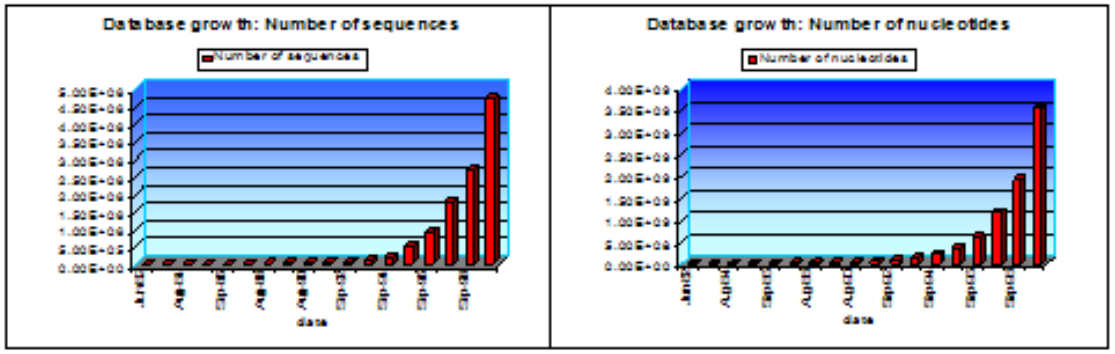
- Consider a combination of kernel matrices $K = \sum_{i=1}^{m} \mu_i K_i \ (\mu_i \geq 0)$ with

Kernel	Data	Similarity measure
K_1	protein sequences	Smith -Waterman
K_2	protein sequences	BLAST
K_3	protein sequences	Pfam HMM
K_4	hydropathy profile	FFT
K_5	protein interactions	linear kernel
K_6	protein interactions	diffusion kernel
K_7	gene expression	RBF kernel
K_8	random numbers	linear kernel

- Improved results by combining kernels [Lanckriet et al., 2004]

Growth of databases

Database category	Data content	Examples
1. Literature database	Bibliographic citations	MEDLINE (1971)
	On-line journals	
2. Factual database	Nucleic acid sequences	GenBank (1982), EMBL (1982), DDBJ (1984)
	Amino acid sequences	PIR (1968), PRF (1979), SWISS-PROT (1986)
	3D molecular structures	PDB (1971), CSD (1965)
3. Knowledge base	Motif libraries	PROSITE (1988)
	Molecular classifications	SCOP (1994)
	Biochemical pathways	KEGG (1995)



Hence: computational complexity important (e.g. exploit sparse matrices)

Data fusion with kernels (1)

- **Multiple kernel learning** in SVM:
  Comparison of different norms in combining data sources:

$$\begin{aligned} \min_{\alpha} \max_{\theta} \quad & \alpha^T (\sum_{j=1}^{p} \theta_j Q_j) \alpha \\ \text{subject to} \quad & Q_j \succeq 0, j = 1, ..., p \\ & \alpha \in \mathcal{C} \\ & \theta_j \geq 0, j = 1, ..., p \\ & \|\theta\|_m = 1 \end{aligned}$$

  including the cases $m = 1, 2, \infty$.

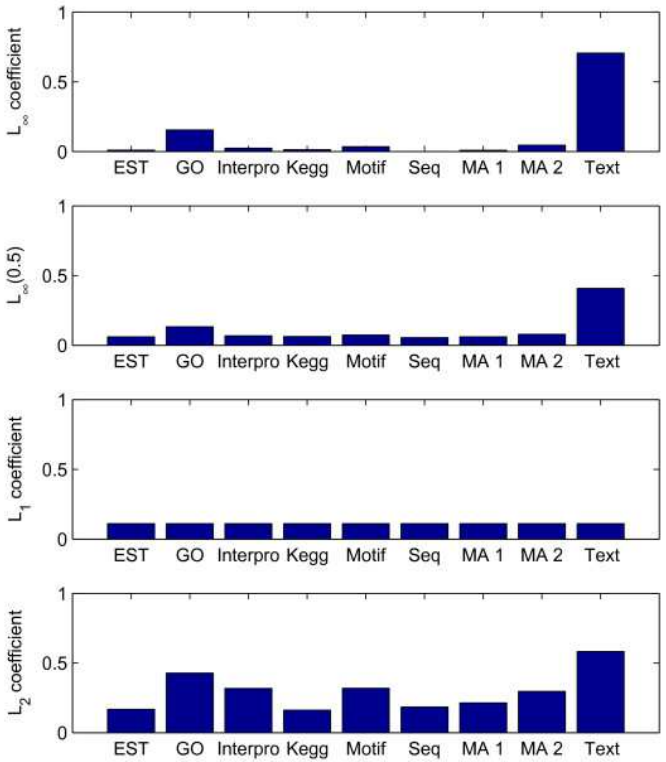
- **Non-sparse** 2-norm combination typically yields better results

[Yu S., Falck T., Daemen A., Tranchevent L., Suykens J.., De Moor B., Moreau Y., BMC Bioinformatics, 2010]

Data fusion with kernels (2)

- Experiment: disease gene prioritization
- Combined kernels from 9 heterogeneous genomic sources
- Average coefficients of 20 repetitions
- 3 most important data-sources ranked by L_∞: Text, GO, Motif
- L_2 method shows the same ranking on the 3 best data sources
- L_2 method gives a more refined ranking

(notation: L_n with $n = \frac{m}{m-1}$)



[Yu S., Falck T., Daemen A., Tranchevent L., Suykens J.., De Moor B., Moreau Y., BMC Bioinformatics, 2010]

Textmining applications

- **Text categorization:** Classification of natural text (or hypertext) documents into a fixed number of predefined categories based on their content. This problem arises in a number of different areas including **email filtering, web searching, office automation, sorting documents by topic, and classification of new agency stories**.

- **Kernels** somehow incorporate a **similarity measure** between instances, and it is reasonable to assume that experts working in the specific application domain have already identified valid similarity measures, particular in areas such as **information retrieval** (IR).

- Transform documents, which typically are strings of characters, into a **representation suitable for the learning algorithm** and the classification task.

Text categorization

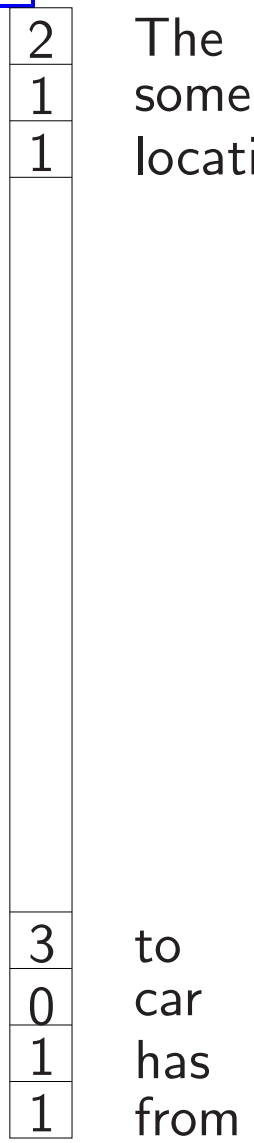
The snow continues to pile up across Colorado

with some locations reporting accumulations of

more than fifty inches up to twelve feet.

The Denver Metro area has picked up from

one to two feet with additional inches likely



The
some
locations

to
car
has
from

Textmining: vector space model (1)

- Encodes the document into a vector where each component represents a corresponding word (called vector space model or **bag-of-words** model)

- Each distinct word w_i corresponds to a component of the feature vector, with the number of times the word w_i occurs within the document d as its value.

- To avoid unnecessarily large feature vectors, typically words are considered if they occur in the training data at least 3 times ("stop–words" like "and","or", etc. are not considered).

- There is one feature vector per document d.

Textmining: vector space model (2)

- **TF (Term Frequency)**: The i-th component of the feature vector is the number of times that a word appears in the document.

- **TF-IDF** uses the above TF multiplied with the IDF (inverse document frequency). The document frequency (DF) is the number of times that the word occurs in all the documents (excluding words that occur in less than three documents). The **inverse document frequency** (IDF) is defined as:

$$IDF\left(w_i\right) = \log\left(\frac{|D|}{DF(w_i)}\right)$$

  where $|D|$ is the number of documents. Typically, the feature vector of the TF-IDF entries is normalized to unit length.

- These representation schemes leads to very **high–dimensional feature spaces** containing 10,000 dimensions and more. Note that this approach **neglects the grammatical structure** of the text.

Text categorization: characteristics

- **High dimensional** input spaces
- Few irrelevant features
- Document vectors are sparse, i.e. many elements of the vectors are zero. This fact can be computationally exploited.
- Many text categorization problems are linearly separable. Hence **linear SVMs** will usually perform well.
- Construction of the kernel:

$$K(x, z) := \varphi(x)^T \varphi(z)$$

with **explicit feature map** $\varphi(\cdot)$ defined according to the vector space model (bag-of-words model).

String (sequence) kernels (1)

- **Bag-of-Words model:** Relative position of words or phrases has little importance in most Information Retrieval (IR) tasks. Documents are just represented by word frequencies (and possibly additional weighting and normalization), and the loss of the information regarding the word positions is more than compensated for by the use of powerful algorithms working in vector space.

  Some methods inject positional information using phrases or multi-word units or local co-occurrence statistics. The underlying techniques are still based on a vector space representation.

- **String (sequence) kernels:** String kernels are similarity measures between documents seen as **sequences of symbols** (e.g., possible characters) over an **alphabet**.

  In general, similarity is assessed by the number of matching subsequences shared by two sequences. The **more subsequences** they have **in common**, the **more similar** they are.

String (sequence) kernels (2)

- **Some definitions:**
  Let Σ be a finite alphabet. A string is a finite sequence of characters from Σ. For strings s, t, $|s|$ denotes the length of the string $s = s_1 \dots s_{|s|}$, and st is the string obtained by concatenating s and t. The string $s[i:j]$ is the substring $s_i \dots s_j$ of s. We say that u is a **subsequence** of s, if there exist indices $\mathbf{i} = (i_1, \dots, i_{|u|})$, with $1 \leq i_1 < \dots < i_{|u|} \leq |s|$, such that $u = s[\mathbf{i}]$ for short.

  Example: If s is the sequence $CART$ and $\mathbf{i} = [2;4]$, then $s[\mathbf{i}]$ is the subsequence AT.

- **Definition of feature map:** For a string s and $u \in \Sigma^n$:

$$\phi_u(s) = \sum_{\mathbf{i}: u = s[\mathbf{i}]} \lambda^{l(\mathbf{i})}$$

  for some $\lambda \leq 1$. These features measure the number of occurrences of subsequences in the string s weighting them according to their lengths.

String (sequence) kernels (3)

Definition of feature map and kernel function - Example:
Consider as documents the words

cat, *car*, *bat*, *bar*

Consider e.g. $k = 2$. One has then a 8-dimensional feature space:

	c-a	c-t	a-t	b-a	b-t	c-r	a-r	b-r
ϕ(cat)	λ^2	λ^3	λ^2	0	0	0	0	0
ϕ(car)	λ^2	0	0	0	0	λ^3	λ^2	0
ϕ(bat)	0	0	λ^2	λ^2	λ^3	0	0	0
ϕ(bar)	0	0	0	λ^2	0	0	λ^2	λ^3

Unnormalized kernel: $K(\text{car,cat}) = \phi(\text{car})^T \phi(\text{cat}) = \lambda^4$.
Normalized kernel: $K(\text{car,car}) = K(\text{cat,cat}) = 2\lambda^4 + \lambda^6$,
$K(\text{car,cat}) = \lambda^4/(2\lambda^4 + \lambda^6) = 1/(2 + \lambda^2)$

String (sequence) kernels (4)

- **String subsequence kernel:**

$$\begin{aligned} K_n(s,t) &= \sum_{u \in \Sigma^n} \phi_u(s)^T \phi_u(t) \\ &= \sum_{u \in \Sigma^n} \sum_{\mathbf{i}: u = s[\mathbf{i}]} \lambda^{l(\mathbf{i})} \sum_{\mathbf{j}: u = t[\mathbf{j}]} \lambda^{l(\mathbf{j})} = \sum_{u \in \Sigma^n} \sum_{\mathbf{i}: u = s[\mathbf{i}]} \sum_{\mathbf{j}: u = t[\mathbf{j}]} \lambda^{l(\mathbf{i})+l(\mathbf{j})} \end{aligned}$$

- A direct computation of these features would involve $O(|\sum|^n)$ time and space. In order to derive an effective procedure for computing such kernel, a **recursive computation** is done for this kernel.

- One prefers to work with a **normalized** kernel: $\tilde{K}(s,t) = \frac{K(s,t)}{\sqrt{K(s,s)K(t,t)}}$

- Related problems for sequence analysis in bioinformatics.

Textmining: one-class problems

In many applications, like search engines, we don't have access to two classes. For training such a task, one only has **one class** of subjects at disposal.

Example: websites one is interested in.

One-class SVM

- One-class SVM formulation [Scholkopf et al., 2000]:

$$\begin{array}{ll} \min\limits_{w,\xi,\rho} & \frac{1}{2}w^T w + \frac{1}{\nu N}\sum\limits_{i=1}^{N}\xi_i - \rho \\ \text{subject to} & w^T\phi(x_i) \geq \rho - \xi_i,\ \forall i \\ & \xi_i \geq 0,\ \forall i \end{array}$$

  with decision function

$$f(x) = \text{sign}[w^T\phi(x_i) - \rho]$$

- Applications to document classification [Manevitz & Yousef, 2002].

Textmining: references

M.W. Berry, Z. Drmac, E.R. Jessup. "Matrices, Vector Spaces, and Information Retrieval," *SIAM Review*, vol 41, No 2, pp 335-362, 1999.

N. Cancedda, E. Gaussier, C. Goutte, J.M. Renders. "Word-Sequence Kernels," *Journal of Machines Learning Research* 3, pp 1059-1082, 2003.

H. Drucker, D. Wu, V. Vapnik. "Support Vector Machines for Spam Categorization," In *Advances in Neural Information Processing Systems* 9, pp 155-161, Cambridge, MA, 1997.

T. Joachims. *Learning to Classify Text using Support Vector Machines*, Kluwer, 2002.

H. Lohdi, C. Saunders, J. Shawe-Taylor, and C. Watkins. "Text classification using string kernels," *Journal of Machines Learning Research* 2, pp 419-444, 2002.

L.M. Manevitz, M. Yousef. "One-Class SVMs for Document Classification," *Journal of Machines Learning Research* 2, pp 139-154, 2002.

S. Tong, D. Koller. "Support Vector Machine Active Learning with Applications to Text Classification," In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

Electricity Load Forecasting (1)

- **Problem:** Electricity load forecasting is an important issue in the energy context. Linear models, ARMAX based models, neural networks, prediction of different hourly profiles, etc. using historical load information and some external explanatory variables (like weather indicators) are among the main techniques applied so far.

- **Reference:**
  M.-W. Chang, B.-J. Chen, C.-J. Lin, "EUNITE Network Competition: Electricity Load Forecasting," 2001.

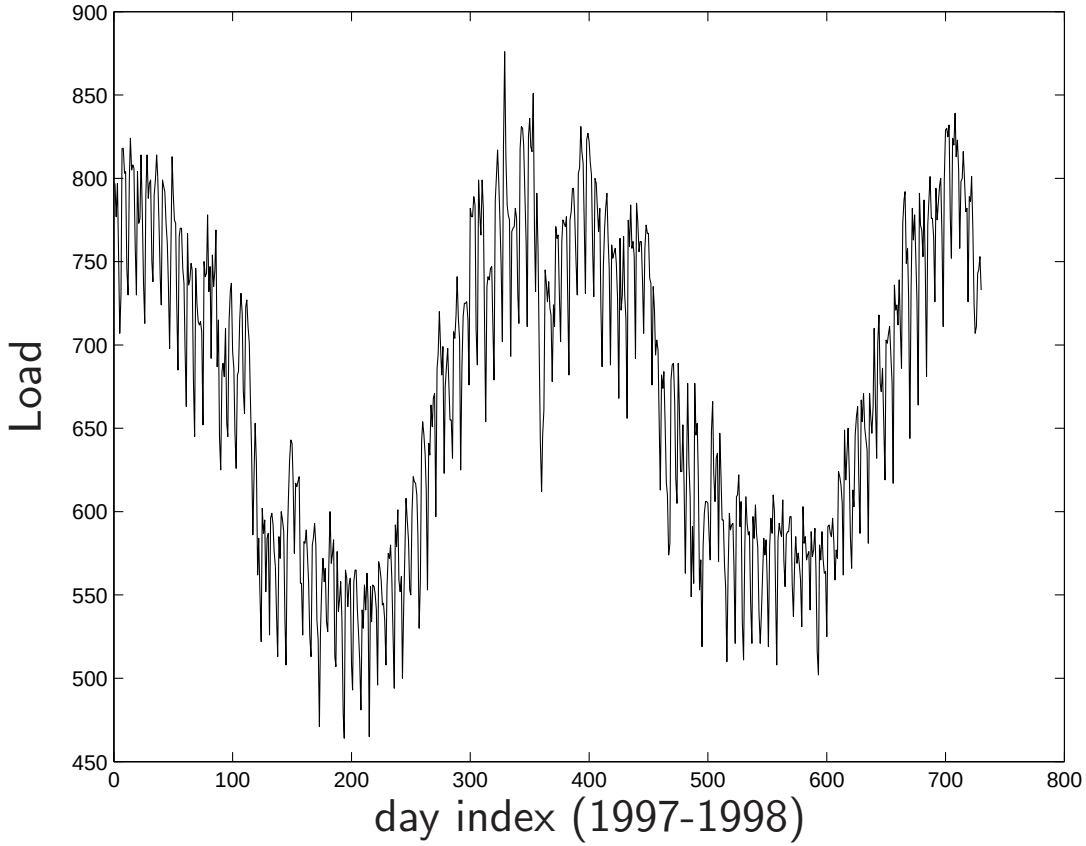
- **Data:**
  In 2001 EUNITE (European Network on Intelligent Technologies for Smart Adaptive Systems) organized a competition based on the problem of short term load forecasting.

Electricity Load Forecasting (2)

- The organizers provided the following data:
  - Electricity Load demand recorded every half an hour, for the years 1997 and 1998.
  - Average daily temperature, from 1995 to 1998.
  - Dates of Holidays, 1997 to 1999.

- **Goal:**
  To predict the maximum daily values of electrical load for each one of the 31 days in January 1999.

Electricity Load Forecasting (3)



This figure shows the available load data for the 2-years training period that were available for the competition participants. It is clear that there is a strong seasonal behaviour of the series.

Electricity Load Forecasting (4)

- **Winning Result:** SVM model had lowest MAPE (mean absolute prediction error) for out-of-sample iterative prediction (Jan 1999).

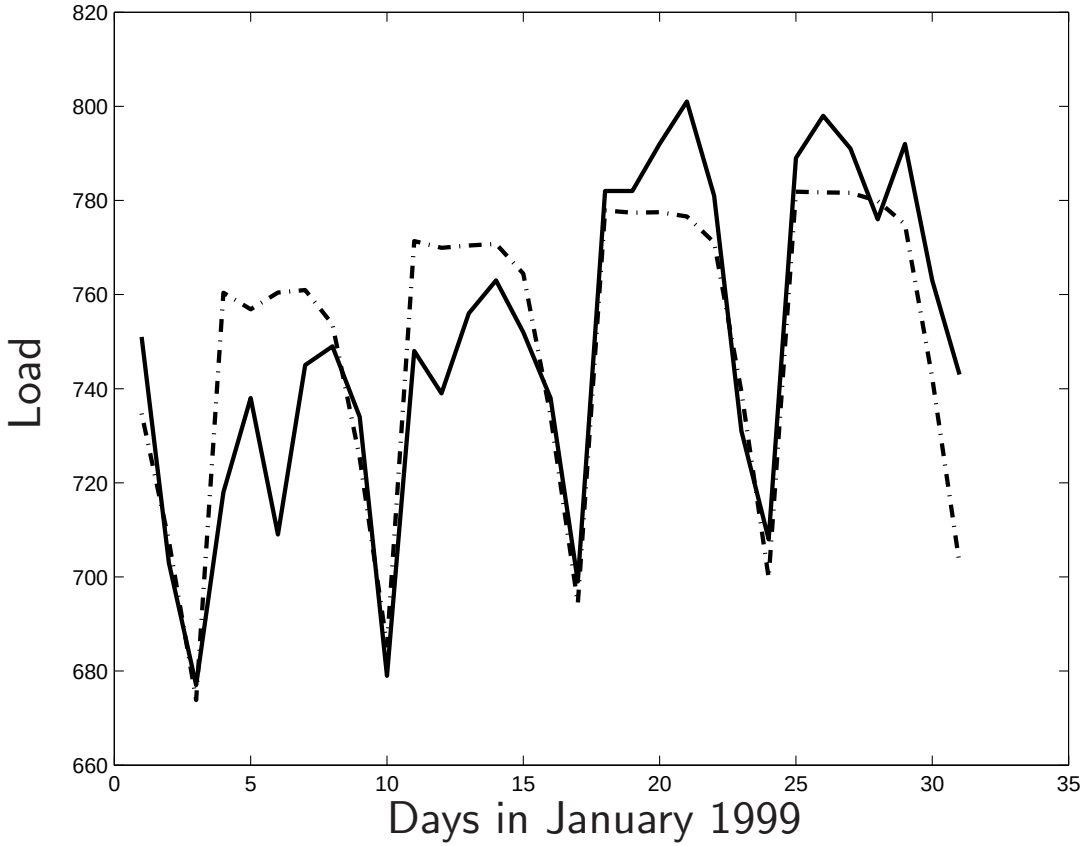
- **Methodology and Implementation:** The winning model was trained using the maximal load as the target, and the following attributes:

  – Seven attributes for maximal loads of the past seven days
  – Seven binary attributes, indicating the day of the week
  – One binary attribute indicating if the particular day is a holiday
  – One attribute for the daily average temperature

  This last attribute was later removed. Only keeping the loads and calendar information was better for short term forecasting.

- A test set was defined as the data for January 1998, and the SVM was trained on the remaining data. Summer data were discarded. Only winter data were used for training of the final model. An RBF kernel was used.

Electricity Load Forecasting (5)



True (full line) and predicted ('-.' line) values for the winning SVM model.