Support Vector Machines (SVM), and applications in bioinformatics

Jean-Philippe Vert

Bioinformatics Center, Kyoto University, Japan
Jean-Philippe.Vert@mines.org

Graduate School of Mathematics, Kyushu University, Japan, June 28, 2002.

Outline

1. Overview of statistical learning theory
2. Linear Support vector machines
3. Non-linear SVM and kernels
4. Applications in bioinformatics

Part 1

An overview of Statistical Learning Theory (SLT)

The pattern recognition problem

- Observation : $x \in \mathcal{X}$
- Class : $y \in \{-1, 1\}$
- Goal : predict the unknown class of an observation

Examples

- Character recognition (OCR): x is an image, y is a letter
- Face recognition: x is an image, y indicates the presence of a face in the picture
- Text classification: x is a text, y is a category (topic, spam / non spam...)
- Medical diagnosis: x is a set of features (age, sex, blood type, genome...), y indicates the risk.

Probabilistic setting

- (X, Y) assumed to be a $\mathcal{X} \times \{-1, 1\}$-valued random pair
- Unknown joint distribution P
- A classifier is a mapping $f : \mathcal{X} \to \{-1, 1\}$
- Risk of a classifier: $R(f) = P(f(X) \neq Y)$

Learning algorithm

- $S = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ P-i.i.d. is observed
- A learning algorithm is a mapping $S \to f_n \in \mathcal{F}$, where $\mathcal{F}$ is a set of classifiers
- Goal: $R(f_n)$ as small as possible

Empirical risk minimization

- Empirical risk of a classifier:

$$R_{emp}(f) = \frac{1}{n} \sum_{i=1}^{n} 1_{\{f(X_i) \neq Y_i\}}$$

- ERM induction principle: choose

$$\hat{f} = \arg\min_{f \in \mathcal{F}} R_{emp}(f)$$

- Question: under which conditions is the algorithm consistent, i.e., $R(\hat{f}_n) \to \inf_{f \in \mathcal{F}} R(f)$ when $n \to \infty$?

Uniform law of large numbers

Let f^* s.t. $R(f^*) = \inf_{f \in \mathcal{F}} R(f)$. Then we have:

$$0 \leq R(\hat{f}) - R(f^*) \leq R(\hat{f}) - R_{emp}(\hat{f}) + R_{emp}(f^*) - R(f^*)$$
$$\leq \sup_{f \in \mathcal{F}} \left(R(\hat{f}) - R_{emp}(\hat{f}) \right) + R_{emp}(f^*) - R(f^*)$$

The term $|R_{emp}(f^*) - R(f^*)|$ converges in probability to 0, by the law of large numbers.

Equivalence between ERM and ULLN

Theorem 1. [Vapnik and Chervonenkis] *One-sided uniform converge in probability,*

$$\lim_{n \to +\infty} P \left\{ \sup_{f \in \mathcal{F}} (R(f) - R_{emp}(f)) > \epsilon \right\} = 0,$$

for all $\epsilon > 0$ *is a necessary and sufficient condition for nontrivial consistency of empirical risk minimization.*

(remains consistent even after the "best" functions have been removed).

When does one-sided ULLN hold?

For a given sample $S_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ let $\mathcal{N}(\mathcal{F}, S_n)$ be the cardinality of $\mathcal{F}$ when restricted to $(x_1, \ldots, x_n)$. Then Vapnik showed:

$$P\left\{\sup_{f\in\mathcal{F}}(R(f) - R_{emp}(f)) > \epsilon\right\} \leq 4\exp\left(\log E[\mathcal{N}(\mathcal{F}, S_n)] - \frac{m\epsilon^2}{8}\right)$$

If $\log E[\mathcal{N}(\mathcal{F}, S_n)]$ (the annealed entropy) grows sublinearly, one-sided ULLN holds.

Other capacity concepts: VC entropy

The VC entropy is $H_{\mathcal{F}}(n) = E \log \mathcal{N}(\mathcal{F}, S_n)$. The convergence

$$\lim_{n \to +\infty} \frac{H_{\mathcal{F}}(n)}{n} = 0$$

is equivalent to uniform two-sided convergence of risk:

$$\forall \epsilon > 0, \quad \lim_{n \to +\infty} P \left\{ \sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| > \epsilon \right\} = 0$$

This implies consistency of ERM.

Other capacity concepts: annealed entropy

The annealed entropy is $H_{\mathcal{F}}^{ann}(n) = \log E\mathcal{N}(\mathcal{F}, S_n)$ (larger than VC entropy by Jensen). The convergence

$$\lim_{n \to +\infty} \frac{H_{\mathcal{F}}^{ann}(n)}{n} = 0$$

is equivalent to exponentially fast convergence for the risk:

$$P\left\{ \sup_{f \in \mathcal{F}} |R(f) - R_{emp}(f)| > \epsilon \right\} \leq 4 \exp\left(\left(\frac{H_{\mathcal{F}}^{ann}(n)}{n} - \epsilon^2 \right) .m \right).$$

Other capacity concepts: growth function

The shattering coefficient is $\mathcal{N}(\mathcal{F}, n) = max_{S_n} \mathcal{N}(\mathcal{F}, S_n)$, and the growth function is $G_{\mathcal{F}}(n) = \log \mathcal{N}(\mathcal{F}, n)$. The convergence

$$\lim_{n \to +\infty} \frac{G_{\mathcal{F}}(n)}{n} = 0$$

is equivalent to exponentially fast convergence of risk for all underlying distributions P.

VC dimension

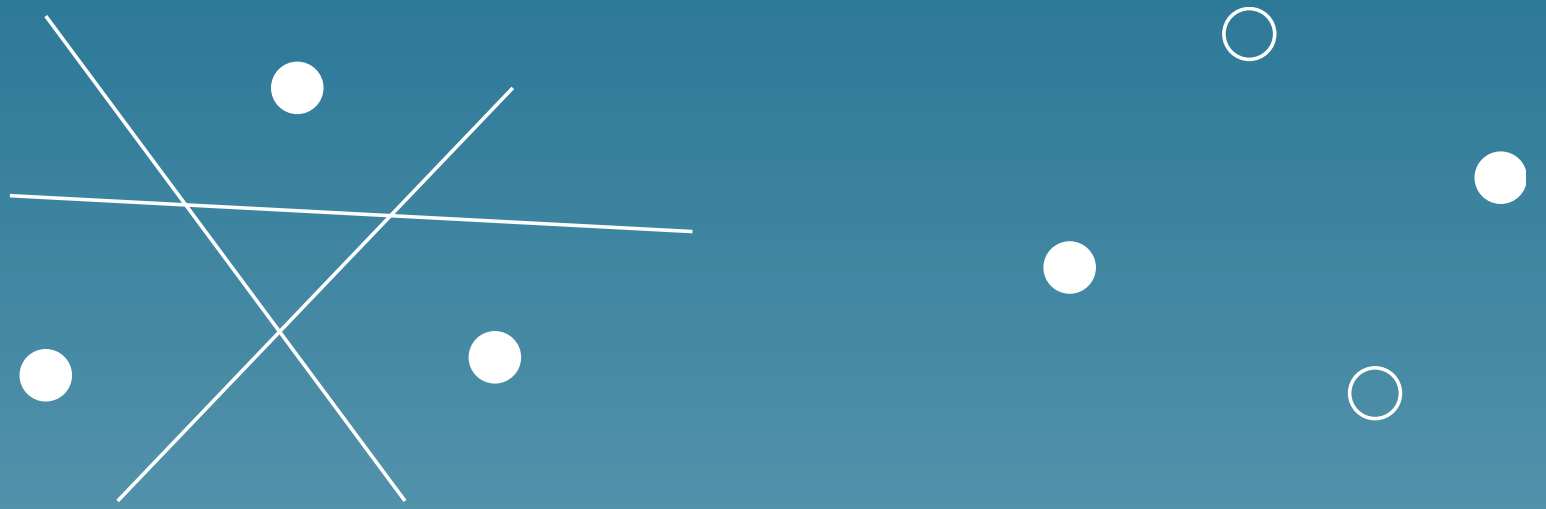
The growth function obviously satisfies $G_{\mathcal{F}}(n) \leq n \log(2)$. The VC dimension h is the maximum number of points which can be shattered (separated in 2^n ways). Then the following holds:

$$G_{\mathcal{F}}(n) \leq h \left(\log \frac{n}{h} + 1 \right).$$

Therefore:

- the growth function increases linearly up to $n = h$
- then in increases logarithmically for $n > h$

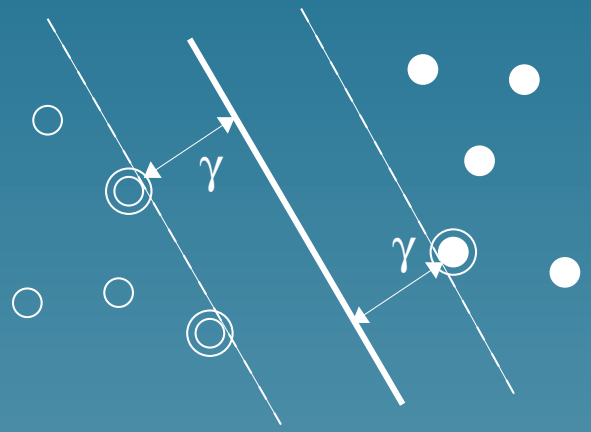
Example: VC dimension of hyperplane classifiers



$$h = 3$$

For hyperplanes in dimension N, the VC dimension is $h = N + 1$.
em This is not interesting in large dimensions

VC dimension of large margin hyperplane classifiers



The set of hyperplanes classifiers with margin at least γ has a VC dimension upper bounded by

$$h \leq \frac{R^2}{M^2}$$

where R is the radius of the smallest sphere containing all x.

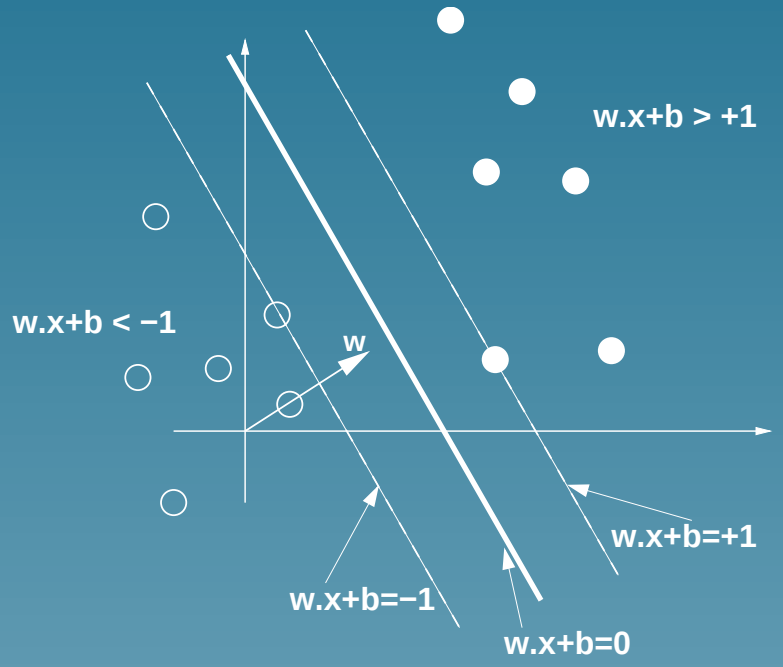
Conclusion about SLT

- Learning algorithms can learn if the capacity of the set of classifiers they produce is controlled
- Vapnik introduced a simple measure of capacity, the VC dimension, whose finiteness is equivalent to the consistency of the learning algorithm
- For hyperplane classifiers, the VC dimension can be controlled irrespective of the dimension of the space, by controlling the margin ... no curse of dimensionality?

Part 2

Support Vector Machines (SVM)

Linear classifiers for separable data



The width of the tube is $\frac{1}{||w||}$. It should be as large a possible, according to SLT

Maximum margin hyperplane

We should maximize $||w||$ under the constraints:

$$\begin{cases} w.x_i + b \geq 1 & \text{if } y_1 = 1, \\ w.x_i + b \leq -1 & \text{if } y_1 = -1, \end{cases}$$

which can be rewritten as: minimize $||w||^2$ under the constraints

$$\forall i \in \{1, \ldots, n\}, \quad y_i(w.x_i + b) - 1 \geq 0.$$

This is a classical quadratic program.

Dual formulation

We introduce Lagrange multipliers $\alpha_1, \ldots, \alpha_n$ and consider the Lagrangian:

$$L(w, b, \alpha) = \frac{1}{2} w'w - \sum_{i=1}^{n} \alpha_i \left[y_i (w.x_i + b) - 1 \right].$$

The dual function is $L(\alpha) = \inf_{w,b} L(w, b, \alpha)$ can be computed by differentiating the Lagrangian. This leads to the dual problem: maximize

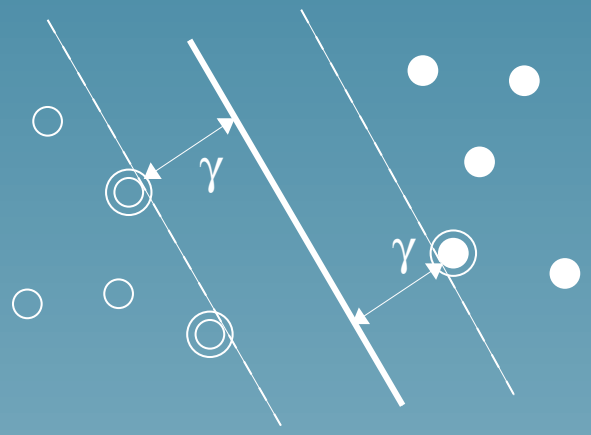
$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i x_j,$$

under the constraints $\alpha \geq 0$ and $\sum_{i=1}^{n} y_i \alpha_i = 0$.

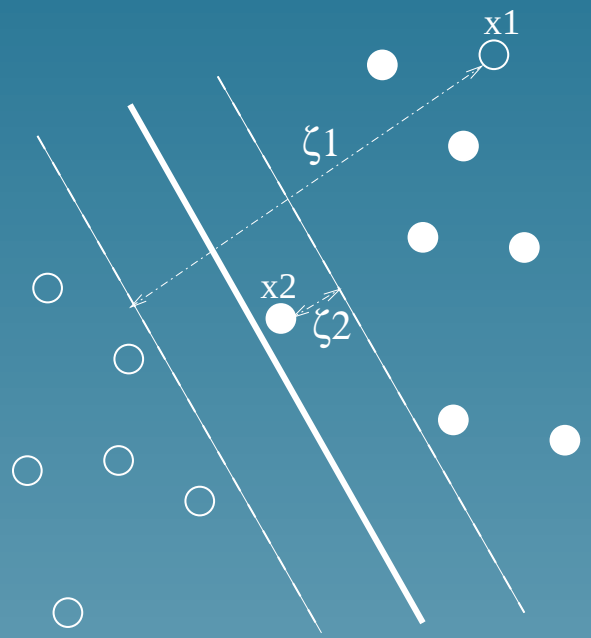
Interpretation of Lagrange multipliers

KKT condition:

- $\alpha_i > 0 \implies y_i(w.x_i + b) = 1$: support vectors
- $\alpha_i = 0 \implies y_i(w.x_i + b) \geq 1$: useless vectors



Linear soft margin SVM



Introduce slack variables $\zeta_1, \ldots, \zeta_n$ to allow misclassification.

Trade-off between large margin and misclassification.

Linear soft margin SVM

Maximize

$$||w||^2 + C \sum_{i=1}^{n} \zeta_i,$$

under the constraints:

$$\begin{cases} \zeta_i \geq 0, \\ y_i(w.x_i + b) \geq 1 - \zeta_i, \end{cases}$$

for $i = 1, \ldots, n$.

Dual formulation of soft margin

Expressing the Lagrangian leads to the dual problem:maximize

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j x_i x_j,$$

under the constraints:

$$\begin{cases} 0 \leq \alpha_i \leq C, \forall i \in \{1, \ldots, n\}, \\ \sum_{i=1}^{n} y_i \alpha_i = 0. \end{cases}$$

Interpretation of the dual variable

SVM classification

w is recovered from α by:

$$w = \sum_{i=1}^{n} \alpha_i x_i.$$

The classification of a new observation $x \in \mathcal{X}$ is based on the sign of:

$$f(x) = w.x + b(\alpha) = \sum_{i=1}^{n} \alpha_i x_i.x + b(\alpha).$$

Only support vectors are used!

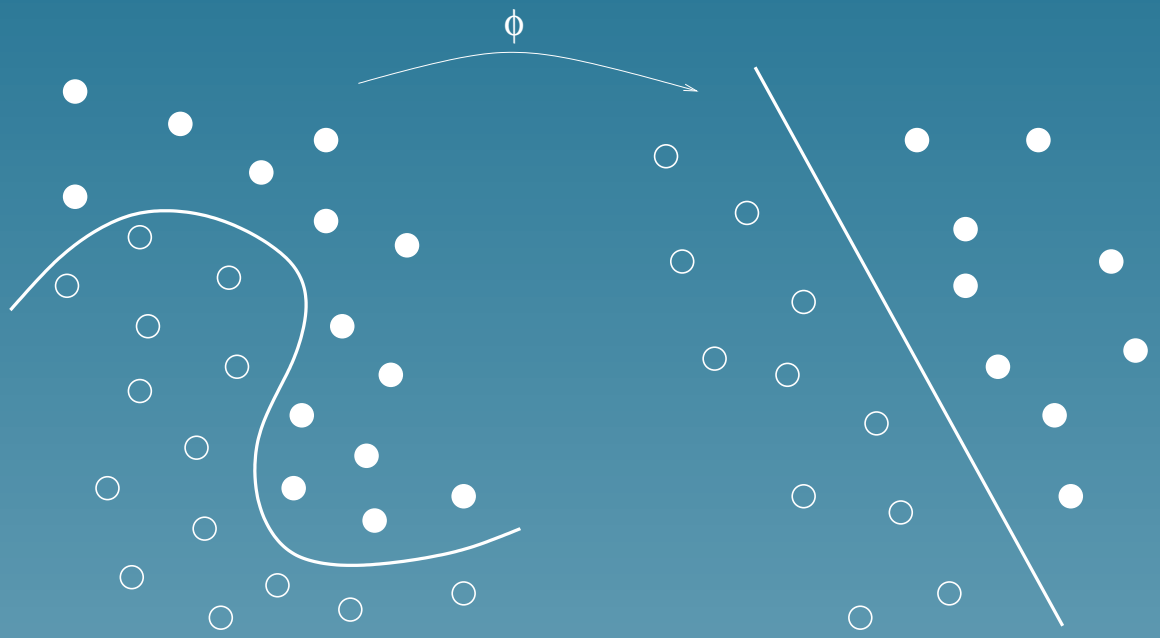
Conclusion about linear SVM

- SVM find a trade-off between large margin and misclassification
- The dual formulation is a quadratic program (convex functional, linear constraints) easy to solve
- The final classifier only uses support vectors (compression..)

Part 3

Nonlinear SVM and kernels

Feature space



$\Phi : \mathcal{X} \rightarrow \mathbb{R}^N$ a non-linear mapping. SVM can be performed in the feature space.

Kernel function

Let $K : \mathcal{X}^2 \to \mathbb{R}$ be defined by:

$$\forall (x, x') \in \mathcal{X}^2, \quad K(x, x') = \Phi(x).\Phi(x').$$

K is called a kernel function.

Kernel trick

The function to minimize is:

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \Phi(x_i)\Phi(x_j)$$

$$= \sum_{i=1}^{n} \alpha_i - \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j)$$

The classifier is:

$$f(x) = \sum_{i=1}^{n} \alpha_i \Phi(x_i).\Phi(x) + b(\alpha) = \sum_{i=1}^{n} \alpha_i K(x_i, x) + b(\alpha).$$

Kernel trick

No Φ anymore! Only $K(.,.)$ is used!

SVM works implicitly in the feature space through the kernel function.

Kernels are often more convenient to work with than explicit Φ!

Kernel examples

- Polynomial:

$$K(x, x') = (x.x')^d$$

- Gaussian radial basis function

$$K(x, x') = \exp\left(\frac{||x - x'||^2}{2\sigma^2}\right)$$

- Sigmoid

$$K(x, x') = \tanh(\kappa x.x' + \theta)$$

How to make a kernel?

Theorem 2. [Mercer] *A function $K : \mathcal{X}^2 \to \mathbb{R}$ is a valid kernel if and only if, for any $p \in \mathbb{N}$ and $(x_1, \ldots, x_p)$, the Gram matrix:*

$$K_{x,x'} = K(x, x')$$

is positive semidefinite.

Many similarity functions on general object spaces are valid kernels!
The validity of a candidate kernel can easily be checked!

Kernel engineering

Design a kernel for a particular application, by including some prior knowledge (invariance,...)

- Translation and rotation invariance for images in OCR
- Periodicity in coding DNA sequences
- Sharing of rare features

More kernel tricks

Any algorithm which can be expressed only in terms of dot product can be kernelized!

- kernel-PCA
- kernel-CCA
- kernel-ICA
- kernel-clustering
- kernel-Fisher discriminant

Conclusion about kernels

- Like other kernel methods, SVM works implicitly in a high-dimensional feature space
- Overfitting is avoided thanks to the choice of large margin
- Computation is tractable thanks to the kernel trick
- Good performance in real-wold applications

Part 4

Applications in bioinformatics

Microarray data analysis

- Gene functional classification: Brown et al. (2000), Pavlidis et al. (2001)
- Tissue classification: Mukherje et al. (1999), Furey et al. (2000), Guyon et al. (2001)

Proteins

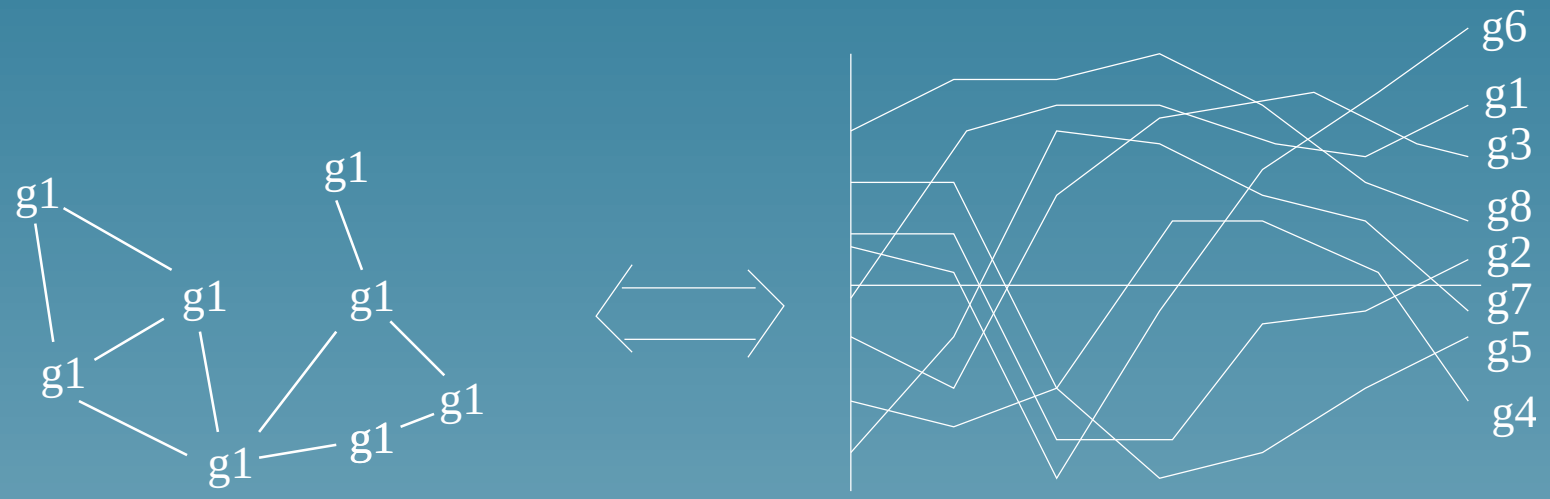
- Family prediction: Jaakkoola et al. (1998)
- Fold recognition : Ding et al. (2001)
- Protein-protein interaction prediction: Bock et al. (2001)
- Secondary structure prediction: Hua et al. (2001)
- Subcellular localization prediction: Hua et al. (2001)

New kernels for bioinformatics

- Fisher kernel (Jaakkoola and Haussler 1998)
- Convolution kernels (Haussler 99, Watkins 1999)
- Kernel for translation initiation site (Zien et al. 2000)
- String kernel (Lodhi et al. 2000)
- Spectrum kernel (Leslie et al., 2002)
- Interpolated kernel (Vert 2002)

More kernel methods and exotic kernels

Graph-driven feature extraction from microarray data (Vert and Kanehisa, 2002)



Gene network

Expression profiles

Graph-driven features extraction

- Diffusion kernel $K_1 = \exp(-\tau L)$, where $L = D - A$ is the Laplace matrix of the graph, to encode the graph topology.
- Linear kernel K_2 between the expression profiles.
- Perform kernel-CCA to extract correlations between the feature spaces of K_1 and K_2, i.e., between the topology of the graph and the expression profiles

Conclusion

Conclusion

- Sound theoretical foundations and good results in real-world applications
- Modularity: any kernel works with any kernel method
- New possibility: engineer kernel for strings, graphs, genes, molecules,...
- More info: `http://www.kernel-machines.org`