Support Vector Machines (SVMs) in bioinformatics

Jean-Philippe Vert

Bioinformatics Center, Kyoto University, Japan

Jean-Philippe.Vert@mines.org

Mathematical Modeling and Statistical Analysis in Biomedical Research, Hiroshima, Japan, Feb. 1-2, 2002.

Outline

1. Introduction to SVMs

2. SVMs in bioinformatics

3. New kernels for bioinformatics

4. Example: signal peptide cleavage site prediction

Part 1

Introduction to SVMs

What are SVMs?

- Learning algorithms for binary classification (idem NN)
- Input: a training set of labelled examples:

$$\{(x_1, y_1), \ldots, (x_n, y_n)\}$$

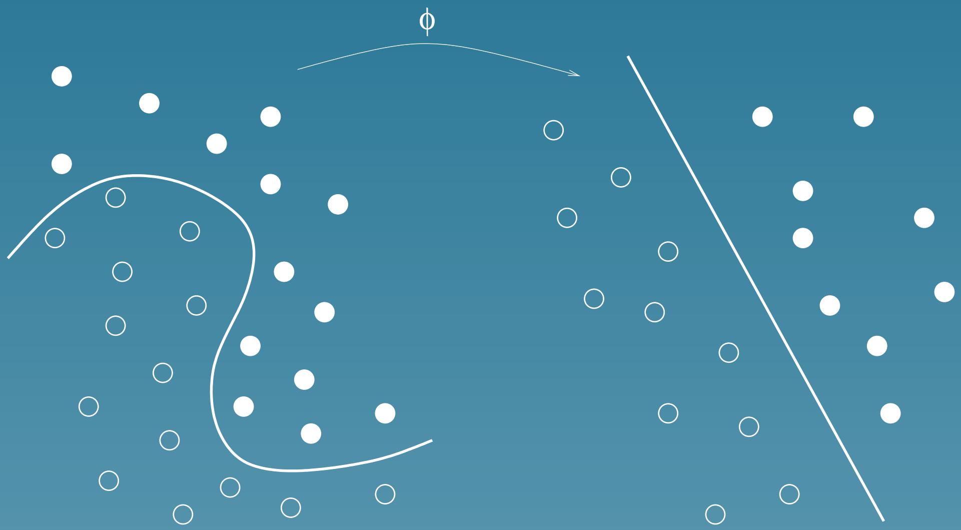
where x_i are objects and y_i are the labels $(+1$ or $-1)$

- Output: a classifier $h : \mathcal{X} \to \{0, 1\}$

Examples

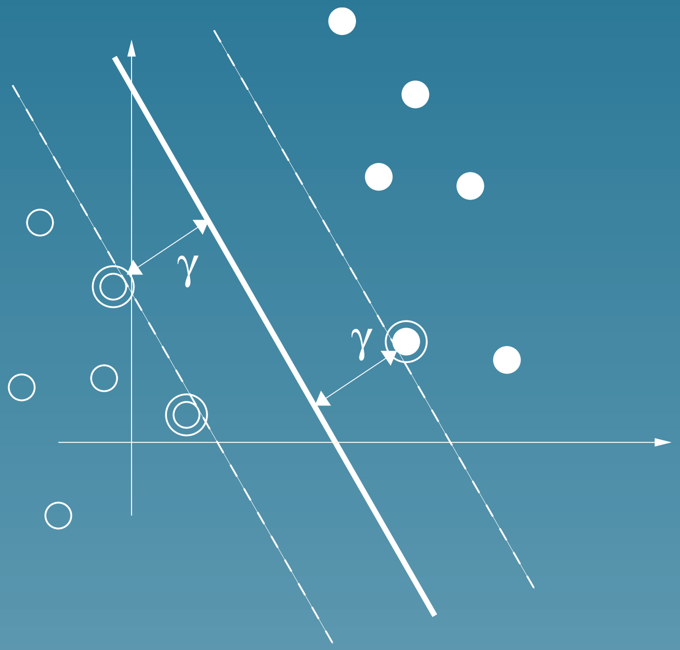
- Character recognition (OCR): x is an image, y is a letter
- Face recognition: x is an image, y indicates the presence of a face in the picture
- Text classification: x is a text, y is a category (topic, spam / non spam...)
- Medical diagnosis: x is a set of features (age, sex, blood type, genome...),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

How SVMs work



- Objects are mapped to a high-dimensional vector space through $\Phi : \mathcal{X} \to \mathbb{R}^D$ (the feature space)
- A linear discrimination is found in the feature space

Linear discrimination



- Largest margin separation in the feature space
- to avoid overfitting (Vapnik, Statistical learning theory)

Finding the optimal hyperplane

- Maximizing the margin is a convex constrained optimization problem
- The dual problem (Lagrangian multipliers) is to find $(\alpha_1, \ldots, \alpha_n)$ solution of:

$$\begin{cases} \max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \Phi(x_i).\Phi(x_j) \\ \forall i = 1, \ldots, n \quad 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

Predicting the class of a new example

- Let $(\alpha_1^*, \ldots, \alpha_n^*)$ be the solution of the problem
- The optimal hyperplane is:

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

- The prediction of the class of a new sample x is:

$$h(x) = sign(f(x))$$

Kernel trick

- Instead of $x \longrightarrow \Phi(x)$ all you need to know is the kernel function:

$$(x, x') \rightarrow K(x, x') \overset{def}{=} \Phi(x).\Phi(x')$$

- Simple kernels can represent complex feature spaces!

Kernel examples

- Linear

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

- Polynomial

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

- Gaussian

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

Conclusion about SVMs

- Work in high-dimensional feature space, but:
  - ⋆ few overfitting thanks to the large margin
  - ⋆ computationaly tractable thanks to the kernel trick
- Very efficient in real-world applications
- Kernels can be engineered for any kind of data based on prior knowledge (to shape the geometry of the feature space, see later)

Part 2

SVMs in bioinfomatics

Microarray data analysis

- Gene functional classification: Brown et al. (2000), Pavlidis et al. (2001)

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)

Proteins

- Family prediction: Jaakkoola et al. (1998)
- Fold recognition : Ding et al. (2001)

Proteins

- Family prediction: Jaakkoola et al. (1998)
- Fold recognition : Ding et al. (2001)
- Protein-protein interaction prediction: Bock 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)

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)

New kernels for bioinformatics

- Fisher kernel (Jaakkoola and Haussler 1998)
- Convolution kernels (Haussler 99, Watkins 1999)

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)

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)

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)

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)

Kernel engineering

Use prior knowledge to build the geometry of the feature space through $K(.,.)$

Part 3

New kernels for bioinfomatics

The problem

- $\mathcal{X}$ a set of (structured) objects

The problem

- $\mathcal{X}$ a set of (structured) objects
- $p(x)$ a probability distribution on $\mathcal{X}$

The problem

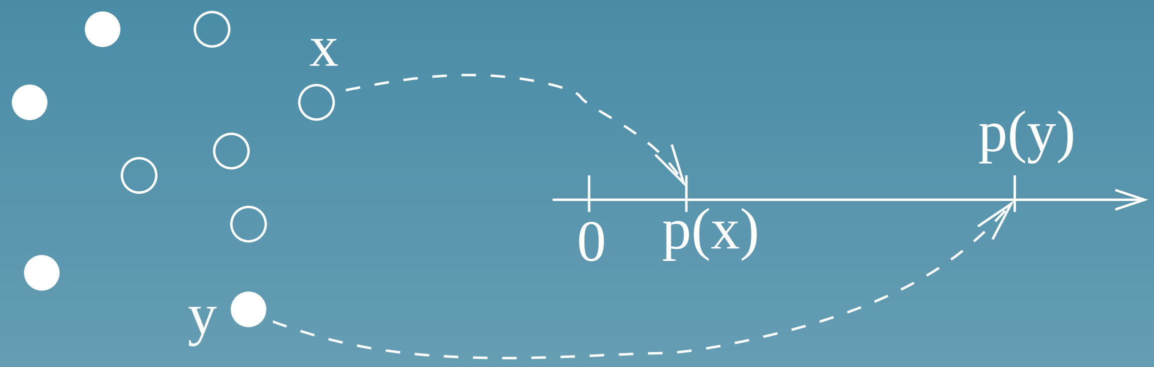
- $\mathcal{X}$ a set of (structured) objects
- $p(x)$ a probability distribution on $\mathcal{X}$
- How to build $K(x, y)$ from $p(x)$?

Product kernel

$$K_{prod}(x, y) = p(x)p(y)$$

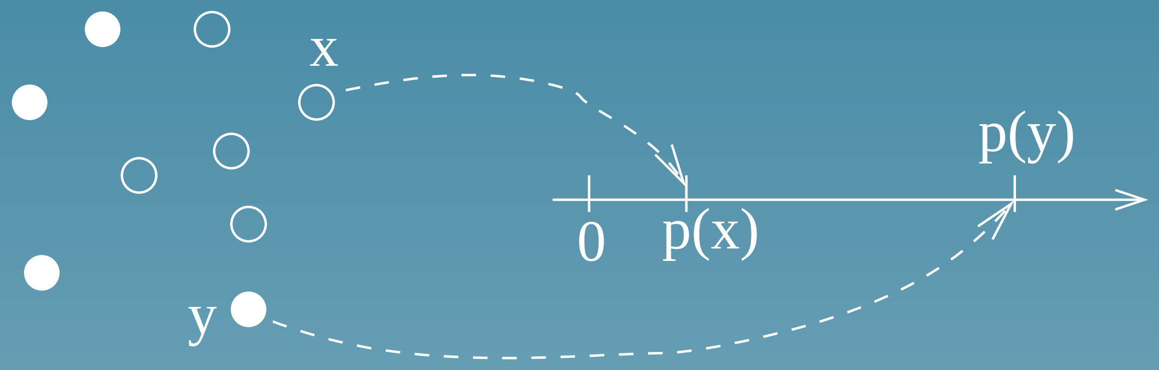
Product kernel

$$K_{prod}(x, y) = p(x)p(y)$$



Product kernel

$$K_{prod}(x, y) = p(x)p(y)$$



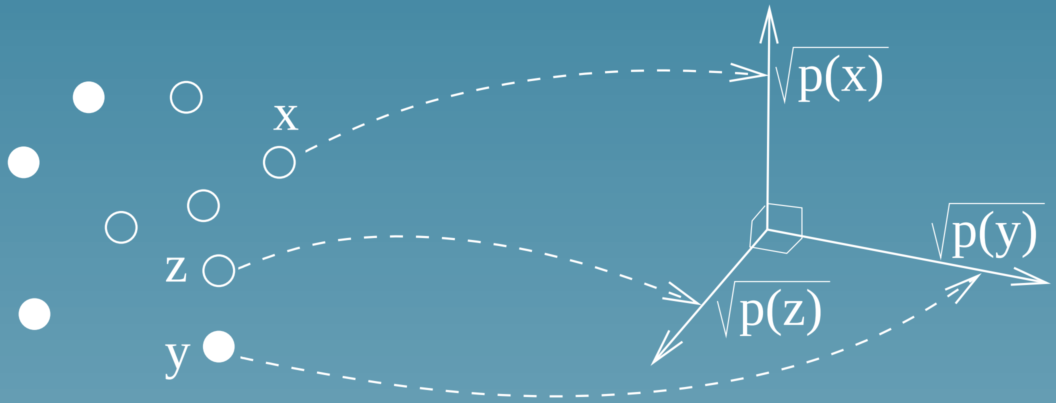
SVM = Bayesian classifier

Diagonal kernel

$$K_{diag}(x, y) = p(x)\delta(x, y)$$

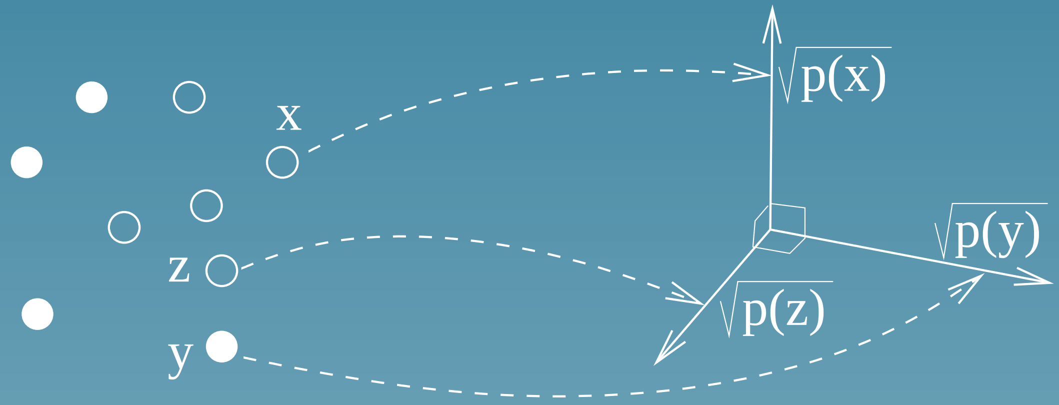
Diagonal kernel

$$K_{diag}(x, y) = p(x)\delta(x, y)$$



Diagonal kernel

$$K_{diag}(x, y) = p(x)\delta(x, y)$$



No learning

Interpolated kernel

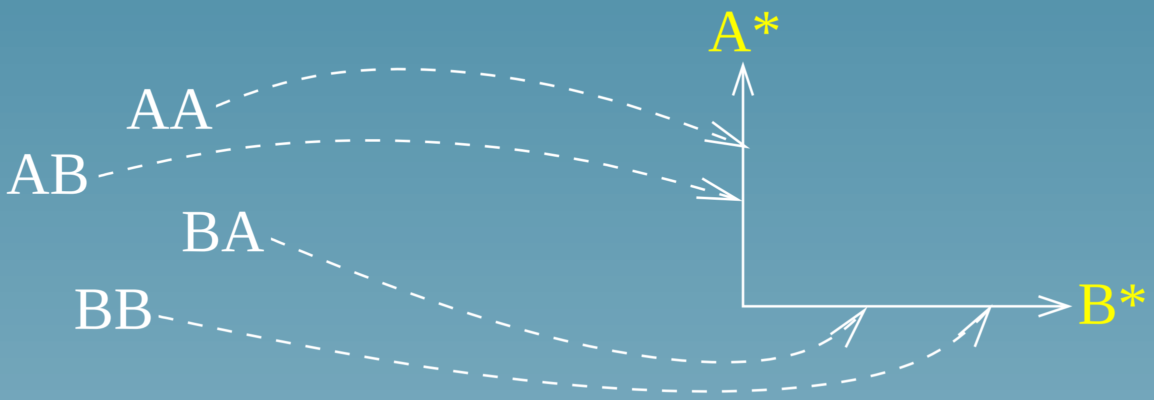
If objects are composite: $x = (x_1, x_2)$:

$$K(x, y) = K_{diag}(x_1, y_1) K_{prod}(x_2, y_2)$$

Interpolated kernel

If objects are composite: $x = (x_1, x_2)$:

$$\begin{aligned} K(x,y) &= K_{diag}(x_1,y_1)K_{prod}(x_2,y_2) \\ &= p(x_1)\delta(x_1,y_1) \times p(x_2|x_1)p(y_2|y_1) \end{aligned}$$



General interpolated kernel

- Composite objects $x = (x_1, \ldots, x_n)$

General interpolated kernel

- Composite objects $x = (x_1, \ldots, x_n)$
- A list of index subsets: $\mathcal{V} = \{I_1, \ldots, I_v\}$ where $I_i \subset \{1, \ldots, n\}$

General interpolated kernel

- Composite objects $x = (x_1, \ldots, x_n)$
- A list of index subsets: $\mathcal{V} = \{I_1, \ldots, I_v\}$ where $I_i \subset \{1, \ldots, n\}$
- Interpolated kernel:

$$K_{\mathcal{V}}(x, y) = \frac{1}{|\mathcal{V}|} \sum_{I \in \mathcal{V}} K_{diag}(x_I, y_I) K_{prod}(x_{I^c}, y_{I^c})$$

Rare common subparts

For a given $p(x)$ and $p(y)$, we have:

$$K_{\mathcal{V}}(x, y) = K_{prod}(x, y) \times \frac{1}{|\mathcal{V}|} \sum_{I \in \mathcal{V}} \frac{\delta(x_I, y_I)}{p(x_I)}$$

Rare common subparts

For a given $p(x)$ and $p(y)$, we have:

$$K_{\mathcal{V}}(x,y) = K_{prod}(x,y) \times \frac{1}{|\mathcal{V}|} \sum_{I \in \mathcal{V}} \frac{\delta(x_I, y_I)}{p(x_I)}$$

x and y get closer in the feature space when they share rare common subparts

Implementation

- Factorization for particular choices of $p(.)$ and $\mathcal{V}$

Implementation

- Factorization for particular choices of $p(.)$ and $\mathcal{V}$
- Example:
  - ⋆ $\mathcal{V} = \mathcal{P}(\{1, \ldots, n\})$ the set of all subsets: $|\mathcal{V}| = 2^n$

Implementation

- Factorization for particular choices of $p(.)$ and $\mathcal{V}$
- Example:
  - ⋆ $\mathcal{V} = \mathcal{P}(\{1, \ldots, n\})$ the set of all subsets: $|\mathcal{V}| = 2^n$
  - ⋆ product distribution $p(x) = \prod_{j=1}^n p_j(x_j)$.

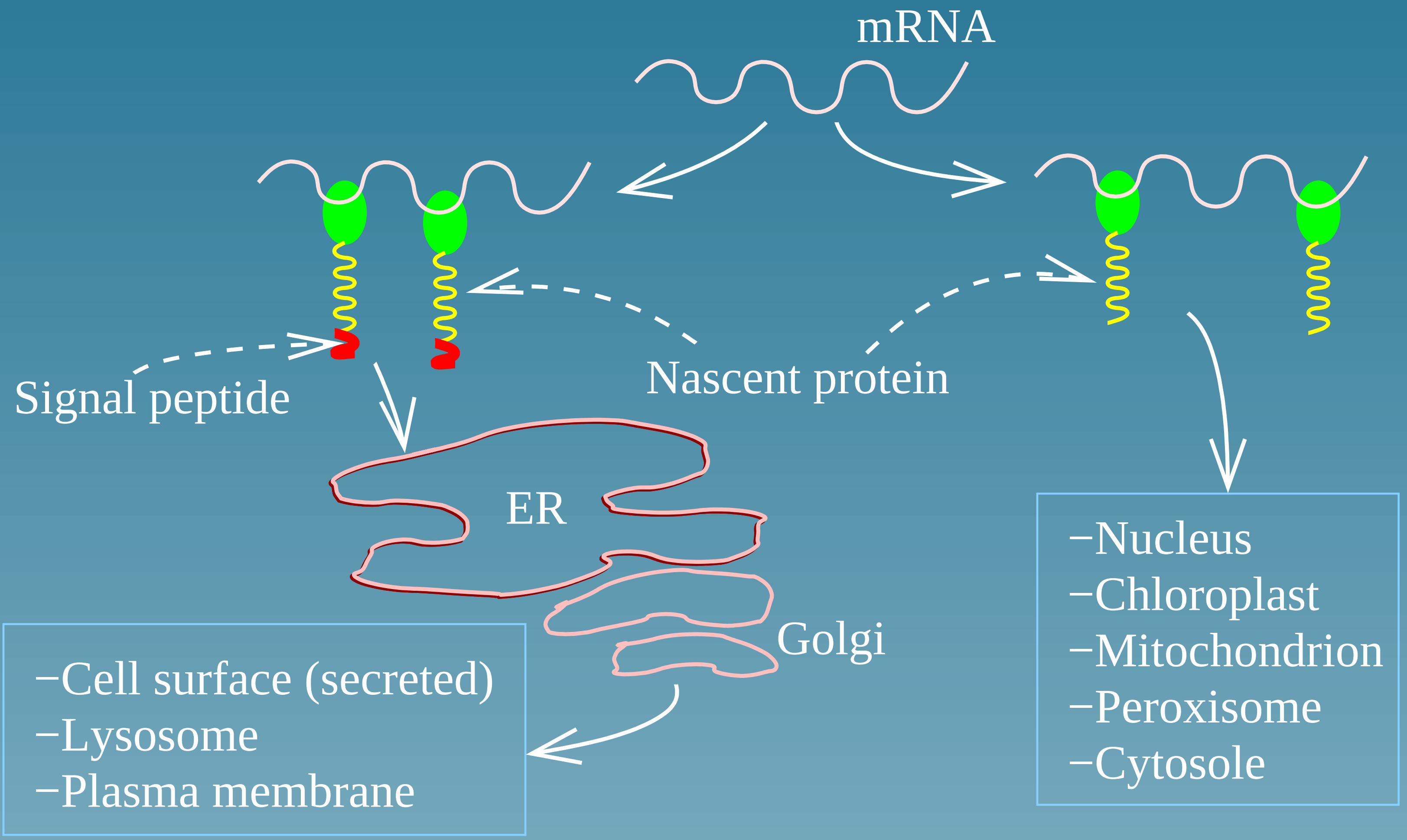
Implementation

- Factorization for particular choices of $p(.)$ and $\mathcal{V}$
- Example:
  - ⋆ $\mathcal{V} = \mathcal{P}(\{1, \ldots, n\})$ the set of all subsets: $|\mathcal{V}| = 2^n$
  - ⋆ product distribution $p(x) = \prod_{j=1}^{n} p_j(x_j)$.
  - ⋆ implementation in $O(n)$

Part 4

Application: SVM prediction of signal peptide cleavage site

Secretory pathway



−Cell surface (secreted)
−Lysosome
−Plasma membrane

−Nucleus
−Chloroplast
−Mitochondrion
−Peroxisome
−Cytosole

Signal peptides

Protein	-1	+1
(1)	MKANAKTIIAGMIALAISHTAMA	EE. . .
(2)	MKQSTIALALLPLLFTPVTKA	RT. . .
(3)	MKATKLVLGAVILGSTLLAG	CS. . .

(1):Leucine-binding protein, (2):Pre-alkaline phosphatase, (3)Pre-lipoprotein

Signal peptides

Protein		-1	+1
(1)	MKANAKTIIAGMIALAISHTAMA		EE...
(2)	MKQSTIALALLPLLFTPVTKA		RT...
(3)	MKATKLVLGAVILGSTLLAG		CS...

(1):Leucine-binding protein, (2):Pre-alkaline phosphatase, (3)Pre-lipoprotein

- 6-12 hydrophobic residues (in yellow)
- (-3,-1) : small uncharged residues

Experiment

- Challenge : classification of aminoacids windows, positive if cleavage occurs between -1 and $+1$:

$$[x_{-8}, x_{-7}, \ldots, x_{-1}, x_1, x_2]$$

Experiment

- Challenge : classification of aminoacids windows, positive if cleavage occurs between -1 and $+1$:

$$[x_{-8}, x_{-7}, \ldots, x_{-1}, x_1, x_2]$$

- 1,418 positive examples, 65,216 negative examples

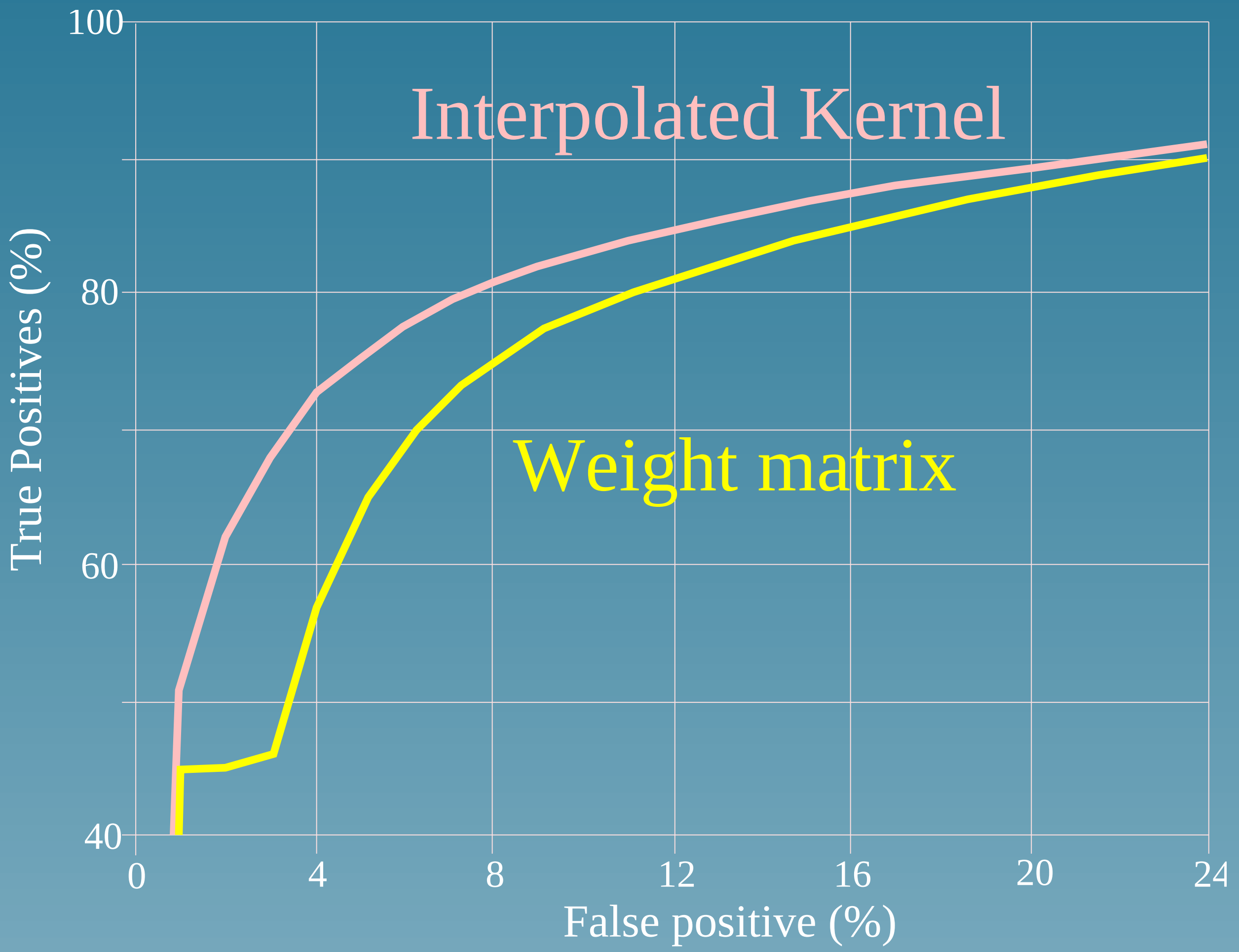
Experiment

- Challenge : classification of aminoacids windows, positive if cleavage occurs between -1 and +1:

$$[x_{-8}, x_{-7}, \ldots, x_{-1}, x_1, x_2]$$

- 1,418 positive examples, 65,216 negative examples
- Computation of a weight matrix:
  SVM + K_{prod} (naive Bayes) vs SVM + $K_{interpolated}$

Result: ROC curves



Conclusion

Conclusion

- SVMs are efficient learning algorithms with a sound theoretical background
- They are gaining popularity in bioinformatics
- Possibility to work with many features and noise
- Poosibility to include biological knowledge in the kernel
- Hot topics: kernel engineering, use of other kernel methods (clustering, PCA, ICA,...)