Kernels for Protein Sequences

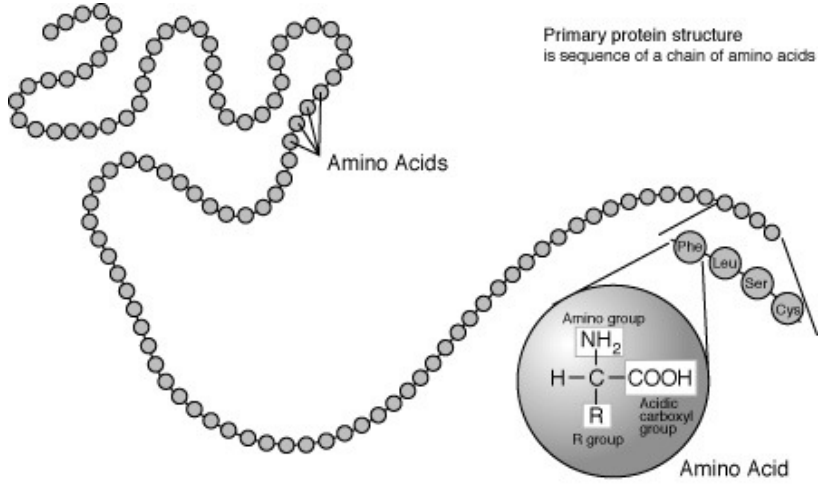
Kernel Methods and Structured Domains workshop, Whistler, Canada, December 10th, 2005.

Jean-Philippe Vert

Ecole des Mines de Paris - ParisTech

Jean-Philippe.Vert@ensmp.fr

Protein sequences



A : Alanine	*V* : Valine	*L* : Leucine
F : Phenylalanine	*P* : Proline	*M* : Méthionine
E : Acide glutamique	*K* : Lysine	*R* : Arginine
T : Threonine	*C* : Cysteine	*N* : Asparagine
H : Histidine	*V* : Thyrosine	*W* : Tryptophane
I : Isoleucine	*S* : Sérine	*Q* : Glutamine
D : Acide aspartique	*G* : Glycine	

Challenges with protein sequences

- A protein sequences can be seen as a *variable-length sequence* over the *20-letter alphabet* of amino-acids, e.g., insuline:
  FVNQHLCGSHLVEALYLVCGERGFFYTPKA
- These sequences are produced at a fast rate (result of the *sequencing programs*)
- Need for algorithms to *compare, classify, analyze* these sequences
- Applications: classification into *functional or structural* classes, prediction of *cellular localization* and *interactions*, ...

Kernels for protein sequences

- *Kernel methods* have been widely investigated since Jaakkola et al.'s seminal paper (1998).
- What is a *good kernel*?
  - it should be *mathematically valid* (symmetric, p.d. or c.p.d.)
  - *fast to compute*
  - *adapted to the problem* (give good performances)

Kernel engineering for protein sequences

- Define a (possibly high-dimensional) *feature space* of interest
  - Physico-chemical kernels
  - Spectrum, mismatch, substring kernels
  - Pairwise, motif kernels
- Derive a kernel from a *generative model*
  - Fisher kernel
  - Mutual information kernel
  - Marginalized kernel
- Derive a kernel from a *similarity measure*
  - Local alignment kernel

Define a (possibly high-dimensional) feature space of interest

Physico-chemical kernels

How to embed explicitly a sequence $\mathbf{x} \in \mathcal{X}$ into a vector $\Phi(\mathbf{x}) \in \mathbb{R}^n$?
Extract *relevant features*, such as:

- length of the sequence
- *time series analysis of numerical physico-chemical properties* of amino-acids along the sequence (e.g., polarity, hydrophobicity), using for example:
  - Fourier transforms (Wang et al., 2004)
  - Autocorrelation functions (Zhang et al., 2003)

$$r_j = \frac{1}{n-j} \sum_{i=1}^{n-j} h_i h_{i+j}$$

Substring indexation

Alternatively, index the feature space by fixed-length strings, i.e.,

$$\Phi(\mathbf{x}) = (\Phi_u(\mathbf{x}))_{u \in \mathcal{A}^k}$$

where $\Phi_u(\mathbf{x})$ can be:

- the number of occurrences of u in $\mathbf{x}$ (without gaps) : *spectrum kernel* (Leslie et al., 2002)
- the number of occurrences of u in $\mathbf{x}$ up to m mismatches (without gaps) : *mismatch kernel* (Leslie et al., 2004)
- the number of occurrences of u in $\mathbf{x}$ allowing gaps, with a weight decaying exponentially with the number of gaps : *substring kernel* (Lohdi et al., 2002)

Substring indexation in practice

- Implementation in $O(|\mathbf{x}| + |\mathbf{x}'|)$ in memory and time for the spectrum and mismatch kernels (with suffix trees)
- Implementation in $O(|\mathbf{x}| \times |\mathbf{x}'|)$ in memory and time for the substring kernels
- The feature space has high dimension ($|\mathcal{A}|^k$), so learning requires *regularized methods* (such as SVM)

Dictionary-based indexation

- Chose a *dictionary* of sequences $\mathcal{D} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$
- Chose a *measure of similarity* $s(\mathbf{x}, \mathbf{x}')$
- Define the mapping $\Phi_{\mathcal{D}}(\mathbf{x}) = (s(\mathbf{x}, \mathbf{x}_i))_{\mathbf{x}_i \in \mathcal{D}}$

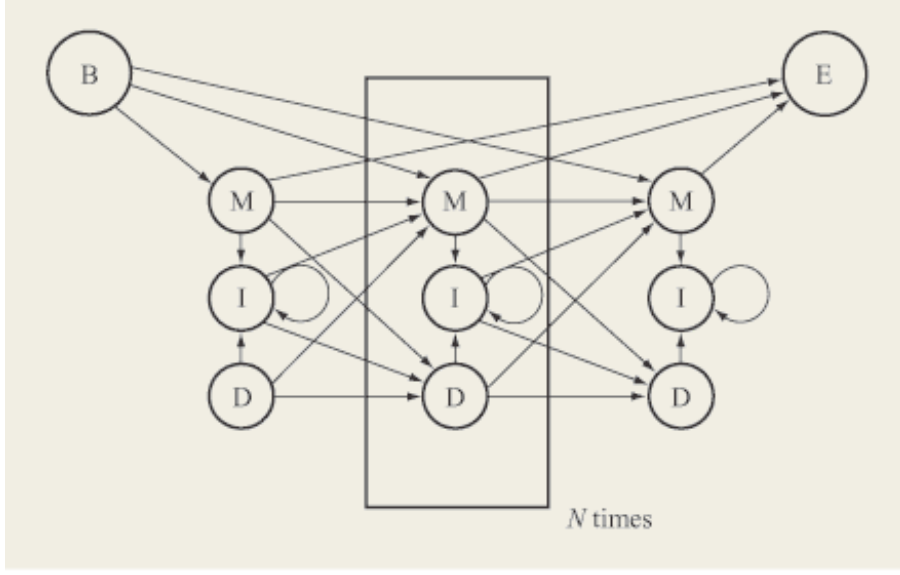
This includes:

- *Motif kernels* (Logan et al., 2001): the dictionary is a library of motifs, the similarity function is a matching function
- *Pairwise kernel* (Liao & Noble, 2003): the dictionary is the training set, the similarity is a classical measure of similarity between sequences.

Derive a kernel from a generative model

Probabilistic models for sequences

Probabilistic modeling of biological sequences is older than kernel designs. Important models include *HMM* for protein sequences, *SCFG* for RNA sequences.



A *model* is a family of distribution

$$\{P_\theta, \theta \in \Theta \subset \mathbb{R}^m\} \subset \mathcal{M}_1^+(\mathcal{X})$$

Fisher kernel

- *Fix a parameter* $\theta_0 \in \Theta$ (e.g., by maximum likelihood over a training set of sequences)
- For each sequence $\mathbf{x}$, compute the *Fisher score vector*:

$$\Phi_{\theta_0}(\mathbf{x}) = \nabla_\theta \log P_\theta(\mathbf{x})|_{\theta=\theta_0}$$

- Form the kernel (Jaakkola et al., 1998):

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \Phi_{\theta_0}(\mathbf{x})^\top I(\theta_0)^{-1} \Phi_{\theta_0}(\mathbf{x}'),$$

where $I(\theta_0) = E_{\theta_0}\left[\Phi_{\theta_0}(\mathbf{x})\Phi_{\theta_0}(\mathbf{x})^\top\right]$ is the Fisher information matrix.

Fisher kernel in practice

- $\Phi_{\theta_0}(\mathbf{x})$ can be computed explicitly for many models (e.g., HMMs)
- $I(\theta_0)$ is often replaced by the identity matrix
- Several different models (i.e., different θ_0) can be trained and combined
- Feature vectors are explicitly computed

Mutual information kernels

- Chose a prior $w(d\theta)$ on the measurable set Θ
- Form the kernel (Seeger, 2002):

$$K\left(\mathbf{x},\mathbf{x}'\right)=\int_{\theta\in\Theta}P_\theta(\mathbf{x})P_\theta(\mathbf{x}')w(d\theta)$$

- *No explicit computation* of a finite-dimensional feature vector
- $K\left(\mathbf{x},\mathbf{x}'\right)=<\phi\left(\mathbf{x}\right),\phi\left(\mathbf{x}'\right)>_{L_2(w)}$ with

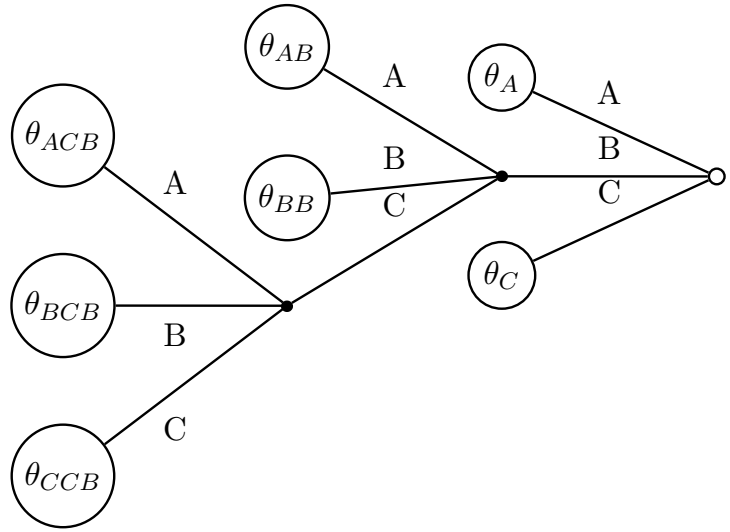
$$\phi\left(\mathbf{x}\right)=\left(P_\theta\left(\mathbf{x}\right)\right)_{\theta\in\Theta}$$

The context-tree kernel

Consider a *variable-memory Markov chain*:

$$P_{\mathcal{D},\theta}(\mathbf{x}) = P_{\mathcal{D},\theta}\left(x_1 \dots x_D\right) \prod_{i=D+1}^{n} P_{\mathcal{D},\theta}\left(x_i \,|\, x_{i-D} \dots x_{i-1}\right)$$

- $\mathcal{D}$ is a suffix tree
- $\theta \in \Sigma^{\mathcal{D}}$ is a set of conditional probabilities (multinomials)



The context-tree kernel (cont.)

- For particular choices of priors, the context-tree kernel:

$$K\left(\mathbf{x},\mathbf{x}'\right)=\sum_{\mathcal{D}}\int_{\theta\in\Sigma^{\mathcal{D}}}P_{\mathcal{D},\theta}(\mathbf{x})P_{\mathcal{D},\theta}(\mathbf{x}')w(d\theta|\mathcal{D})\pi(\mathcal{D})$$

  can be computed in $O(|\mathbf{x}|+|\mathbf{x}'|)$ with a variant of the *Context-Tree Weighting algorithm* (Cuturi et al., 2004).

- This is a *valid mutual information kernel*.
- The similarity is related to information-theoretical measure of *mutual information* between strings.

Marginalized kernels

- For any *observed data* $\mathbf{x} \in \mathcal{X}$, let a *latent variable* $\mathbf{y} \in \mathcal{Y}$ be associated probabilistically through a *conditional probability* $P_{\mathbf{x}}(d\mathbf{y})$.
- Let $K_{\mathcal{Z}}$ be a *kernel for the complete data* $\mathbf{z} = (\mathbf{x}, \mathbf{y})$
- Then the following kernel is a valid kernel on $\mathcal{X}$, called a *marginalized kernel* (Tsuda et al., 2002):

$$K_{\mathcal{X}}\left(\mathbf{x}, \mathbf{x}'\right) := E_{P_{\mathbf{x}}(d\mathbf{y}) \times P_{\mathbf{x}'}(d\mathbf{y}')} K_{\mathcal{Z}}\left(\mathbf{z}, \mathbf{z}'\right)$$

$$= \int \int K_{\mathcal{Z}}\left((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')\right) P_{\mathbf{x}}(d\mathbf{y}) P_{\mathbf{x}'}\left(d\mathbf{y}'\right)$$

Marginalized kernels in practice

- Spectrum kernel on the hidden states of a HMM for *protein sequences* (Tsuda et al., 2002)
- Kernels for *RNA sequences* based on SCFG (Kin et al., 2002)
- Kernels for *graphs* based on random walks on graphs (Kashima et al., 2004)
- Kernels for *multiple alignments* based on phylogenetic models (Vert et al., 2005)

Derive a kernel from a similarity measure

Sequence alignment

How to compare 2 sequences?

$\mathbf{x}_1$ = CGGSLIAMMWFGV

$\mathbf{x}_2$ = CLIVMMNRLMWFGV

Find a good *alignment*:

```
CGGSLIAMM---WFGV
|...||||.....||||
C---LIVMMNRLMWFGV
```

Alignment score

In order to quantify the relevance of an alignment π, define:

- a *substitution matrix* $S \in \mathbb{R}^{\mathcal{A} \times \mathcal{A}}$
- a *gap penalty* function $g : \mathbb{N} \rightarrow \mathbb{R}$

Any alignment is then scored as follows

```
CGGSLIAMM----WFGV
|...|||||....||||
C---LIVMMNRLMWFGV
```

$$s_{S,g}(\pi) = S(C,C) + S(L,L) + S(I,I) + S(A,V) + 2S(M,M) + S(W,W) + S(F,F) + S(G,G) + S(V,V) - g(3) - g(4)$$

Local alignment kernel

- The widely-used Smith-Waterman local alignment score is defined by:

$$SW_{S,g}(\mathbf{x},\mathbf{y}) := \max_{\pi \in \Pi(\mathbf{x},\mathbf{y})} s_{S,g}(\pi).$$

- It is symmetric, but not positive definite...
- The local alignment kernel:

$$K_{LA}^{(\beta)}(\mathbf{x},\mathbf{y}) = \sum_{\pi \in \Pi(\mathbf{x},\mathbf{y})} \exp\left(\beta s\left(\mathbf{x},\mathbf{y},\pi\right)\right),$$

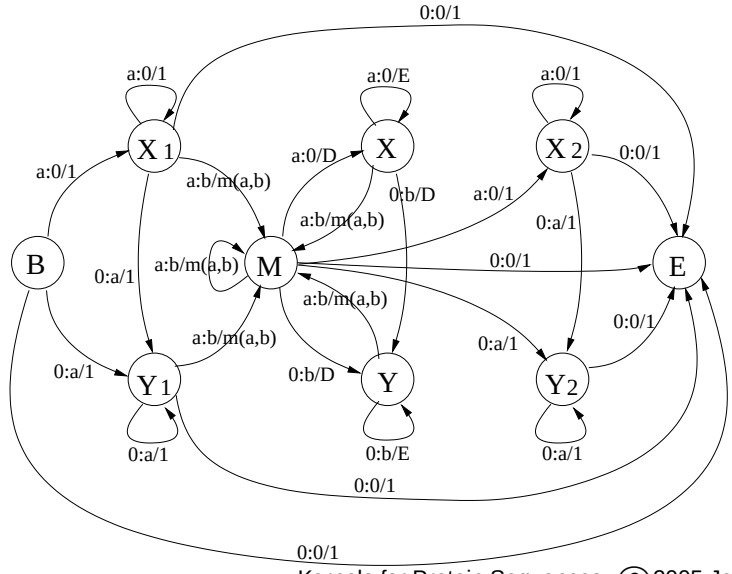
is symmetric positive definite (Saigo et al., 2004).

LA kernel in practice

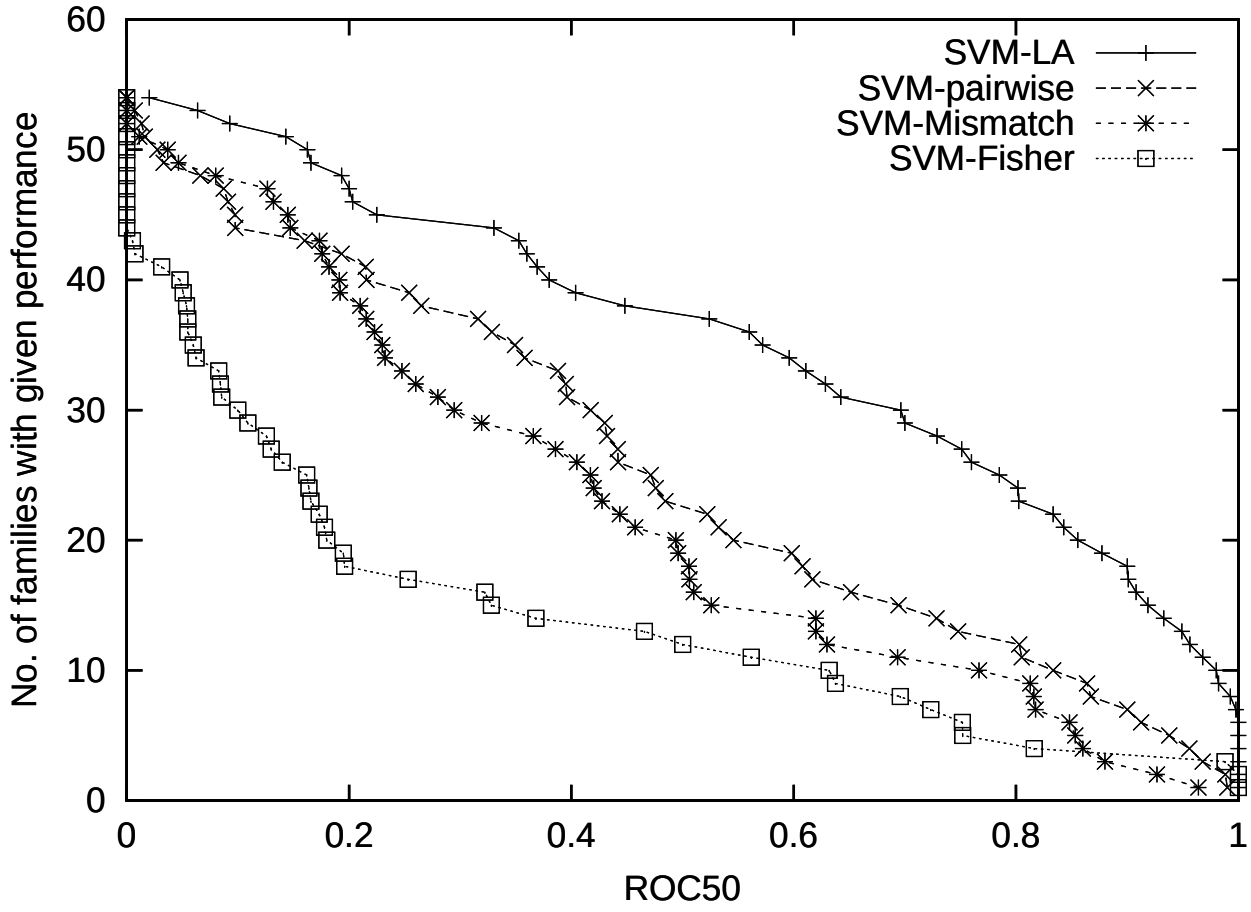
- LA kernel is p.d. because it is a *convolution kernel* (Haussler, 1999):

$$K_{LA}^{(\beta)} = \sum_{n=0}^{\infty} K_0 \star \left(K_a^{(\beta)} \star K_g^{(\beta)} \right)^{(n-1)} \star K_a^{(\beta)} \star K_0.$$

- Implementation by dynamic programming in $O(|\mathbf{x}| \times |\mathbf{x}'|)$



Difference in performance



Performance on the SCOP superfamily recognition benchmark (from Saigo et al., 2004).

Conclusions

- A topic that triggered many interesting developments
- *No universally good kernel*; performance is not the only criterion.
- Current trends
  - *Semi-supervised* approaches (profile kernels, cluster kernels...)
  - *Combination and learning* of kernels