Kernels for Phylogenetic Trees?

Jean-Philippe.Vert@mines.org

Ecole des Mines de Paris
Computational Biology group

American Institute of Mathematics, May 6, 2003.

Outline

1. About kernels
2. What can be done with a kernel
3. Kernel trick example
4. Making kernels for phylogenetic trees

Part 1

About kernels

Definition

- Let $\mathcal{X}$ be a set (e.g., $\mathbb{R}^n$, set of trees, ...)
- A (Mercer) kernel is a mapping $K : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ which is:
  - ⋆ symetric : $K(x, y) = K(y, x)$,
  - ⋆ positive semi-definite: $\sum_{i,j} a_i a_j K(x_i, x_j) \geq 0$ for all $a_i \in \mathbb{R}$ and $x_i \in \mathcal{X}$

Example

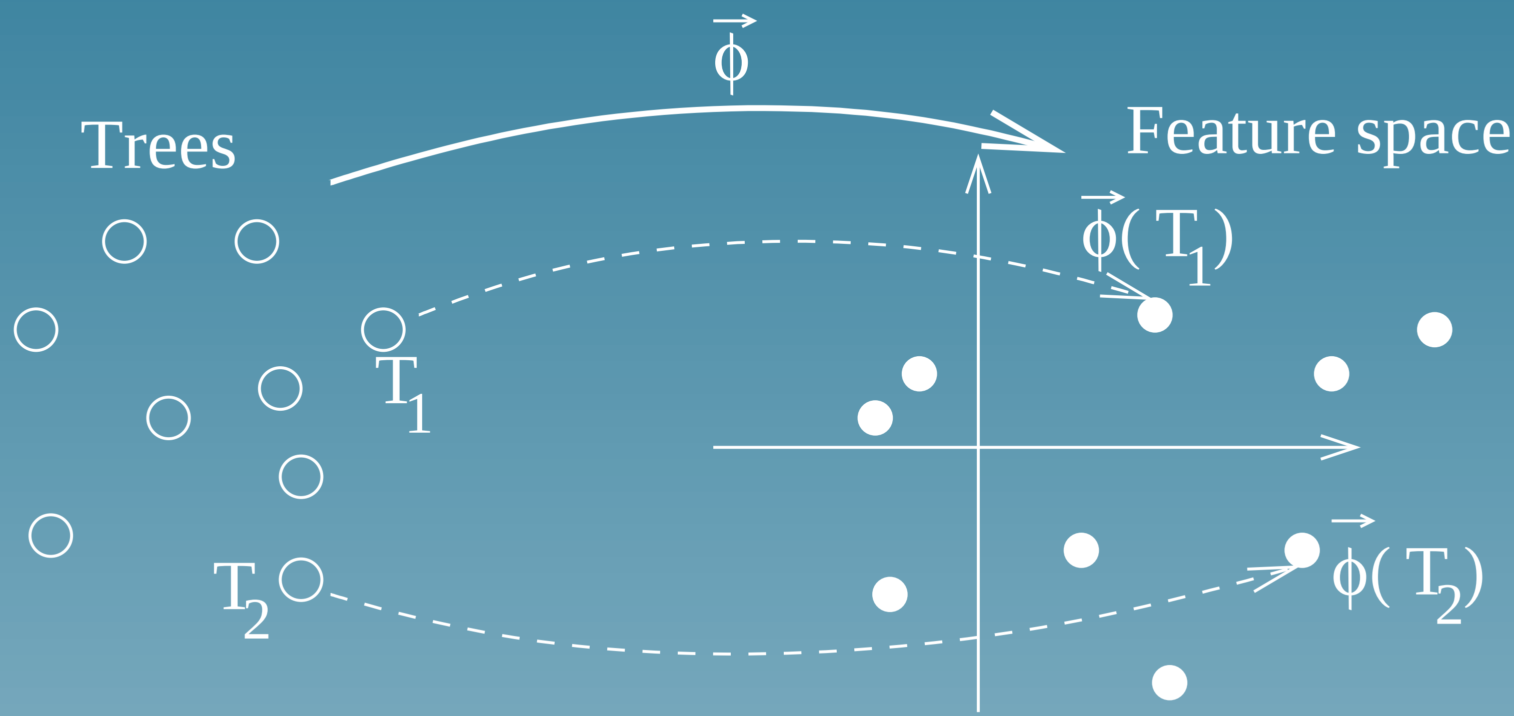
- Suppose $\mathcal{X} = \mathbb{R}^d$. Then the following is a valid kernel:

$$K(\vec{x}, \vec{y}) = \vec{x}.\vec{y}$$

- Indeed:
  - ⋆ $\vec{x}.\vec{y} = \vec{y}.\vec{x}$
  - ⋆ $\sum_{i,j} a_i a_j \vec{x_i}.\vec{x_j} = || \sum_i a_i \vec{x_i} ||^2 \geq 0$

Example: kernel in feature space

$$K(T_i, T_j) \stackrel{def}{=} \vec{\Phi}(T_i).\vec{\Phi}(T_j)$$



All kernels are inner product

- If $K(.,.)$ is a kernel, then there exists a Hilbert space $\mathcal{H}$ and a mapping $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ such that:

$$K(x, y) = < \Phi(x), \Phi(y) >_{\mathcal{H}} .$$

- Proof: by diagonalizing the kernel operator

Avenues we won't explore today

- Functional analysis in Reproducing Kernel Hilbert Spaces (RKHS)
- Solving ill-posed problems via regularization, theory of splines
- Gaussian processes, spatial statistics

Part 2

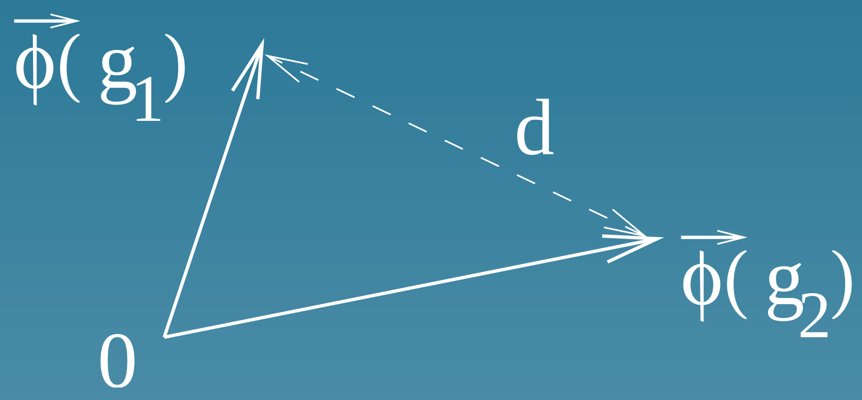
What can you do with a kernel

Overview

Let $K(x, y)$ be a given kernel. Then is it possible to perform various algorithms implicitly in the feature space, such as:

- Computing distances
- Principal component analysis (PCA)
- Canonical correlation analysis (CCA)
- Classification by Support vector machines (SVM)

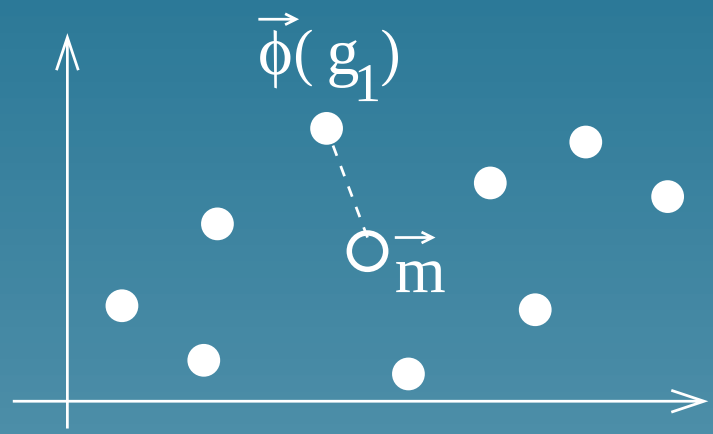
Compute the distance between objects



$$
\begin{aligned}
d(g_1, g_2)^2 &= \|\vec{\Phi}(g_1) - \vec{\Phi}(g_2)\|^2 \\
&= \left(\vec{\Phi}(g_1) - \vec{\Phi}(g_2)\right) . \left(\vec{\Phi}(g_1) - \vec{\Phi}(g_2)\right) \\
&= \vec{\Phi}(g_1).\vec{\Phi}(g_1) + \vec{\Phi}(g_2).\vec{\Phi}(g_2) - 2\vec{\Phi}(g_1).\vec{\Phi}(g_2)
\end{aligned}
$$

$$d(g_1, g_2)^2 = K(g_1, g_1) + K(g_2, g_2) - 2K(g_1, g_2)$$

Distance to the center of mass

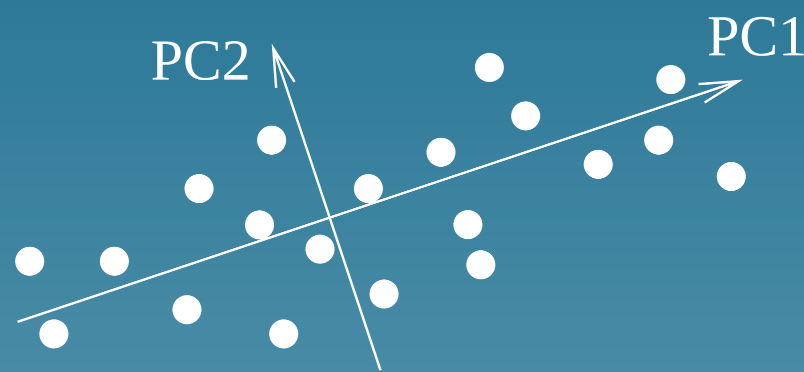


Center of mass: $\vec{m} = \frac{1}{N}\sum_{i=1}^{N}\vec{\Phi}(g_i)$, hence:

$$\|\vec{\Phi}(g_1) - \vec{m}\|^2 = \vec{\Phi}(g_1).\vec{\Phi}(g_1) - 2\vec{\Phi}(g_1).\vec{m} + \vec{m}.\vec{m}$$

$$= K(g_1, g_1) - \frac{2}{N}\sum_{i=1}^{N} K(g_1, g_i) + \frac{1}{N^2}\sum_{i,j=1}^{N} K(g_i, g_j)$$

Principal component analysis

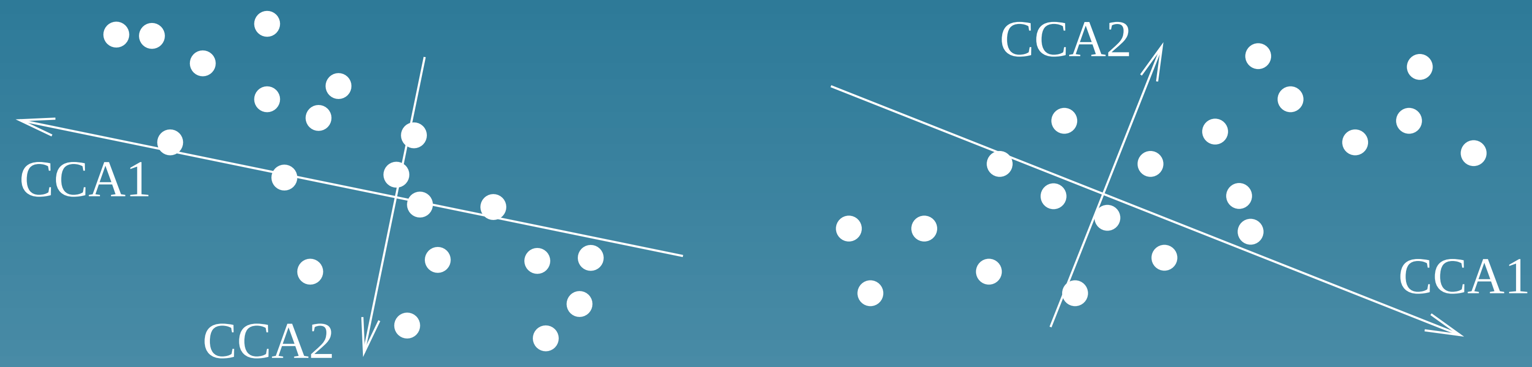


It is equivalent to find the eigenvectors of

$$K = \left(\vec{\Phi}(g_i).\vec{\Phi}(g_j)\right)_{i,j=1\ldots N}$$
$$= \left(K(g_i, g_j)\right)_{i,j=1\ldots N}$$

Useful to project the objects on small-dimensional spaces.

Canonical correlation analysis

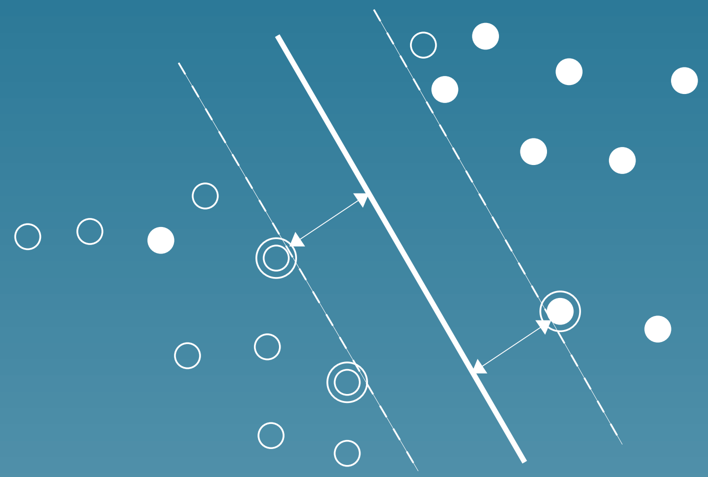


K_1 and K_2 are two kernels for the same objects. CCA can be performed by solving the following generalized eigenvalue problem:

$$\begin{pmatrix} 0 & K_1K_2 \\ K_2K_1 & 0 \end{pmatrix} \vec{\xi} = \rho \begin{pmatrix} K_1^2 & 0 \\ 0 & K_2^2 \end{pmatrix} \vec{\xi}$$

Compare different representations of the same objects.

Support vector machines (SVM)



Find a linear boundary with large margin and few errors

$$\begin{cases} \max_{\vec{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(g_i, g_j) \\ \forall i = 1, \ldots, n \quad 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

Summary

- Kernel trick : once a kernel $K(x, y)$ is given, several analysis can be performed implicitly in the feature space.
- These methods are VERY powerful on many real-world problems
- Modularity: each kernel can work with each method

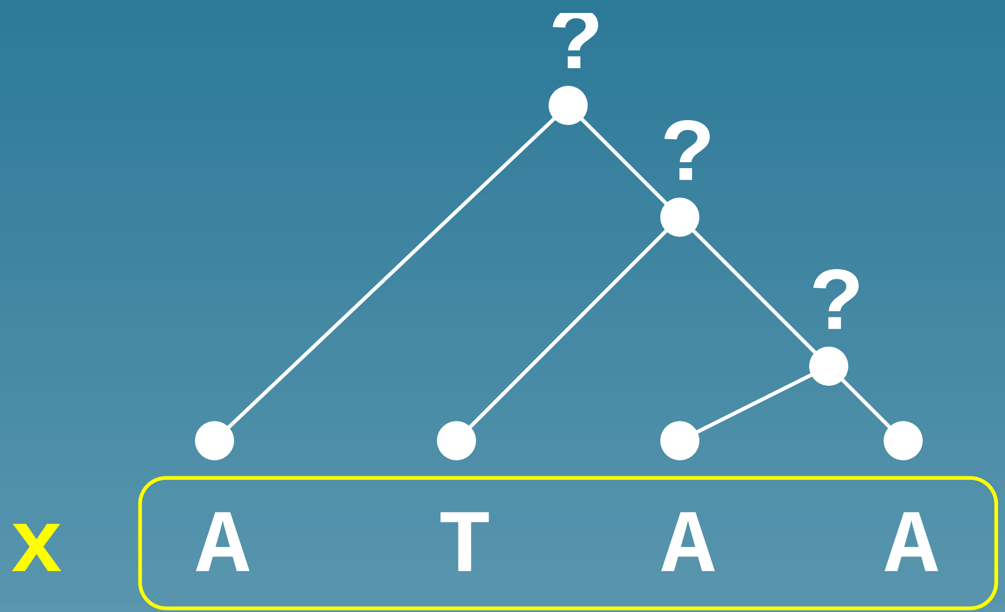
Part 3

Kernel trick example

Kernel for aligned positions

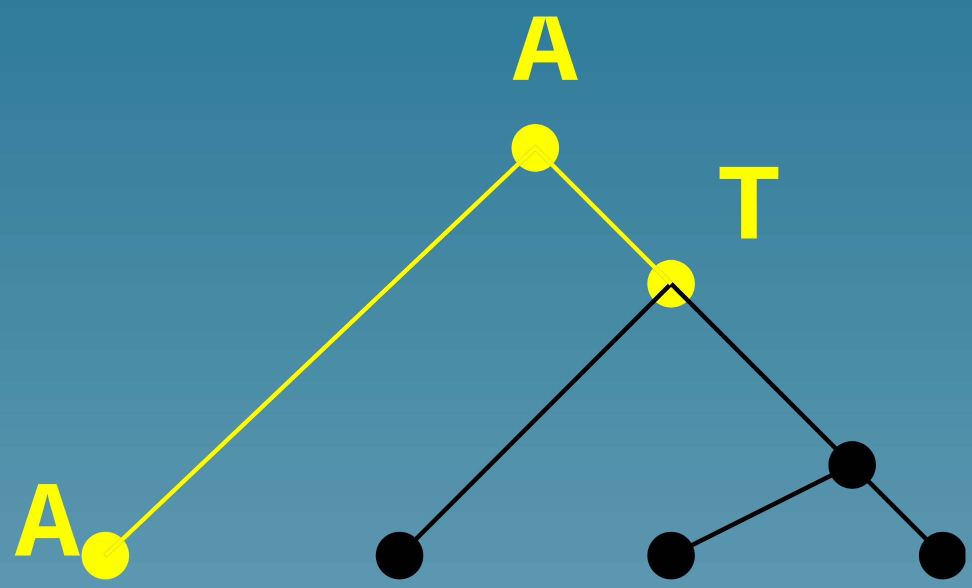
```
AATCGATCGATCGA
ATTCGTTCGATGGA
AATAGTTCCATGCA
TATGGAGCGATTTA
 x    y
```

What we know



We suppose we know a good tree, which defines a probability distribution (e.g., estimated by maximum likelihood)

Evolution pattern



A possible pattern of transmission during evolution defined by a rooted subtree with labeled nodes.

Representation of a profile in terms of evolution patterns

- Consider all possible evolution patterns $(e_1, \ldots, e_N)$, and represent each gene x by the vector:

$$\Phi(x) = \begin{pmatrix} \sqrt{P(e_1)}P(x|e_1) \\ \vdots \\ \sqrt{P(e_N)}P(x|e_N) \end{pmatrix}$$

- Very rich representation

The kernel

$$K(x,y) = \sum_{e \text{ evolution pattern}} P(e)P(x|e)P(y|e)$$

- The sum involves an exponential number of terms...
- ...but it can be computed in linear time.

Part 4

Kernels for phylogenetic trees?

Several approaches

- Define explicitly an interesting feature space where the inner product can be computed quickly
- Spectral analysis of the tree space $\mathcal{T}$

Euclidean tree space

- If $\mathcal{T} = \mathbb{R}^n$, the heat kernel is a valid kernel:

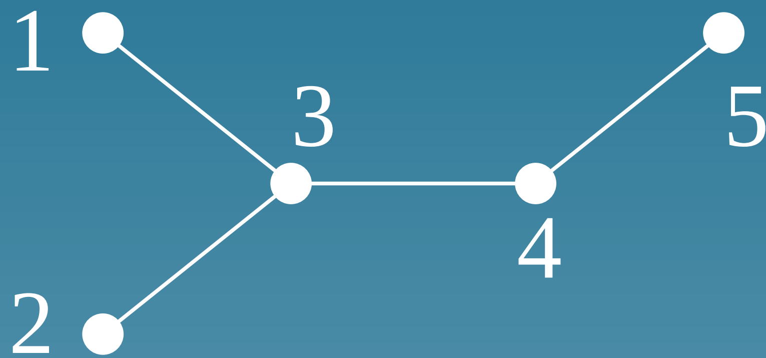
$$K(T_1, T_2) = \exp\left(\frac{||T_1 - T_2||^2}{2\sigma^2}\right).$$

- Related to the Laplacian, Brownian motion etc...

The tree space as a graph

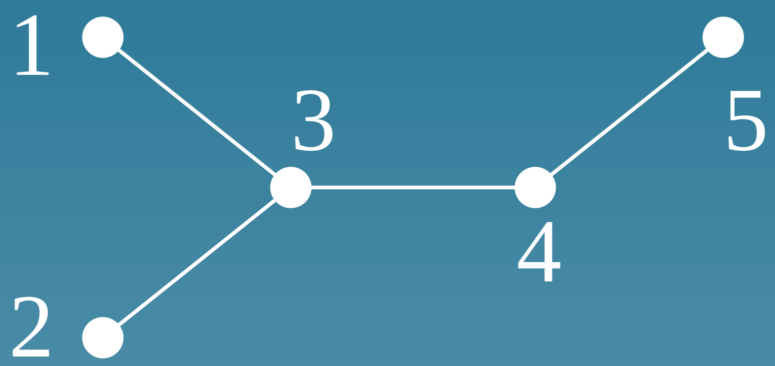
- Nodes are trees, (weighted) edges indicate similarity between two trees
- The discrete heat kernel is a valid kernel for nodes
- $K = \exp(-tL)$, where L is the discrete Laplacian (Kondor and Lafferty, 2002)

Example (1)



$$L = \begin{pmatrix} -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 1 & 1 & -3 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

Example (2)



$$K = \exp(-L) = \begin{pmatrix} 0.49 & 0.12 & 0.23 & 0.10 & 0.03 \\ 0.12 & 0.49 & 0.23 & 0.10 & 0.03 \\ 0.23 & 0.23 & 0.24 & 0.17 & 0.10 \\ 0.10 & 0.10 & 0.17 & 0.31 & 0.30 \\ 0.03 & 0.03 & 0.10 & 0.30 & 0.52 \end{pmatrix}$$

Other tree space

- Riemannian manifold
- Finite group (kernel for permutations...)
- etc?

Conclusion

Conclusion

- A kernel is more than a distance
- Several kernel methods
- Possibility to engineer kernels and obtain useful algorithms