Kernel design and learning

Jean-Philippe Vert
Jean-Philippe.Vert@mines-paristech.fr

Mines ParisTech / Institut Curie / Inserm

Symposium on Learning and Data Science, Paris, April 1-3, 2009 .

Outline

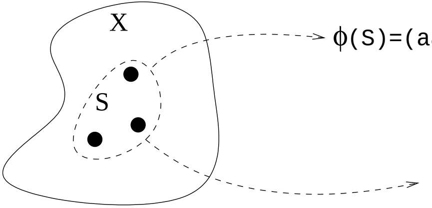
1. Learning with kernels
2. Making kernels
3. Choosing and combining kernels
4. Conclusion

Outline

1. Learning with kernels
2. Making kernels
3. Choosing and combining kernels
4. Conclusion

Motivations

- Develop versatile algorithms to process and analyze data
- No hypothesis made regarding the type of data (vectors, strings, graphs, images, ...)
- Instead we study methods based on pairwise comparisons.



$\phi(S)=(\texttt{aatcgagtcac},\texttt{atggacgtct},\texttt{tgcactact})$

$$K=\begin{pmatrix} 1 & 0.5 & 0.3 \\ 0.5 & 1 & 0.6 \\ 0.3 & 0.6 & 1 \end{pmatrix}$$

Positive Definite Kernels

Definition

A positive definite (p.d.) kernel on the set $\mathcal{X}$ is a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ symmetric:

$$\forall \left(\mathbf{x}, \mathbf{x}'\right) \in \mathcal{X}^2, \quad K\left(\mathbf{x}, \mathbf{x}'\right) = K\left(\mathbf{x}', \mathbf{x}\right),$$

and which satisfies, for all $N \in \mathbb{N}$, $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \in \mathcal{X}^N$ et $(a_1, a_2, \ldots, a_N) \in \mathbb{R}^N$:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) \geq 0.$$

Examples

Classical kernels for vectors ($\mathcal{X} = \mathbb{R}^p$) include:

- The linear kernel

$$K_{lin}\left(\mathbf{x}, \mathbf{x}'\right) = \mathbf{x}^\top \mathbf{x}' \,.$$

- The polynomial kernel

$$K_{poly}\left(\mathbf{x}, \mathbf{x}'\right) = \left(\mathbf{x}^\top \mathbf{x}' + a\right)^d \,.$$

- The Gaussian RBF kernel:

$$K_{Gaussian}\left(\mathbf{x}, \mathbf{x}'\right) = \exp\left(-\frac{\parallel \mathbf{x} - \mathbf{x}' \parallel^2}{2\sigma^2}\right) \,.$$

Geometric interpretation : Kernels as Inner Products

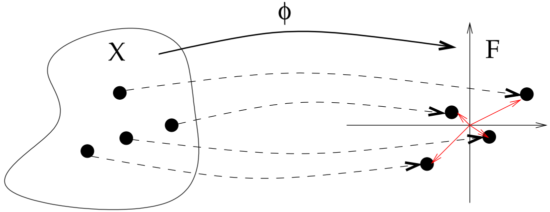
Theorem (Aronszajn, 1950)

K is a p.d. kernel on the set $\mathcal{X}$ if and only if there exists a Hilbert space $\mathcal{H}$ and a mapping

$$\Phi : \mathcal{X} \mapsto \mathcal{H} \,,$$

such that, for any $\mathbf{x}, \mathbf{x}'$ in $\mathcal{X}$:

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \left\langle \Phi\left(\mathbf{x}\right), \Phi\left(\mathbf{x}'\right)\right\rangle_{\mathcal{H}} \ .$$



Functional interpretation: Reproducing Kernel Hilbert Space

- To each p.d. kernel on $\mathcal{X}$ is associated a unique Hilbert space of function $\mathcal{X} \to \mathbb{R}$, called the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$.
- Typical functions are:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}) \,,$$

with norm

$$\| f \|_{\mathcal{H}}^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) \,.$$

Examples: Gaussian RBF kernel

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

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \exp\left(-\frac{\| \mathbf{x} - \mathbf{x}_i \|^2}{2\sigma^2}\right) ,$$

$$\| f \|_{\mathcal{H}}^2 = \int \left| \hat{f}(\omega) \right|^2 e^{\frac{\sigma^2 \omega^2}{2}} d\omega .$$

Small norm $\Longrightarrow$ slow variations.

Examples: Gaussian RBF kernel

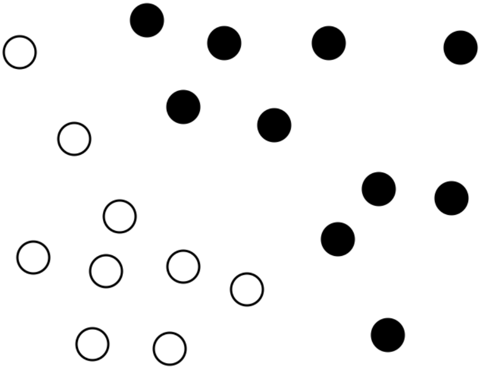
$$K_{Gaussian}\left(\mathbf{x},\mathbf{x}'\right) = \exp\left(-\frac{\parallel \mathbf{x}-\mathbf{x}' \parallel^2}{2\sigma^2}\right) \,,$$

$$f\left(\mathbf{x}\right) = \sum_{i=1}^{n} \alpha_i \exp\left(-\frac{\parallel \mathbf{x}-\mathbf{x}_i \parallel^2}{2\sigma^2}\right) \,,$$

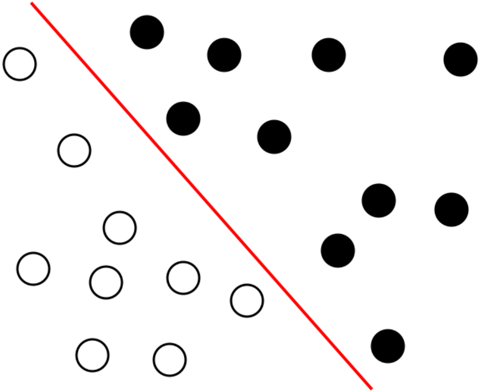
$$\parallel f \parallel_{\mathcal{H}}^2 = \int \left| \hat{f}(\omega) \right|^2 e^{\frac{\sigma^2\omega^2}{2}} d\omega \,.$$

Small norm $\implies$ slow variations.

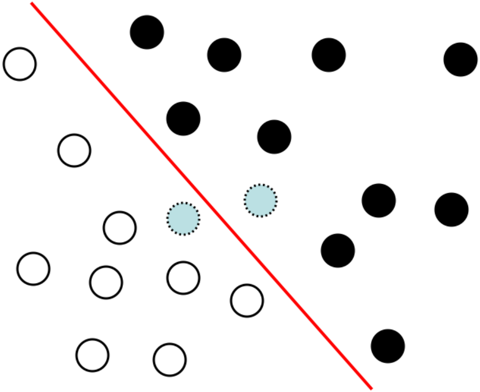
Pattern recognition, *aka* supervised classification



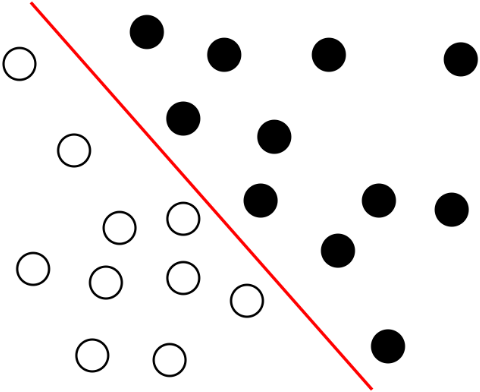
Pattern recognition, *aka* supervised classification



Pattern recognition, *aka* supervised classification



Pattern recognition, *aka* supervised classification

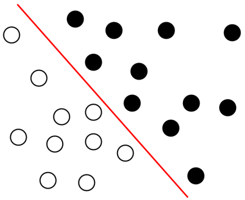


Kernel methods

1. Define an empirical risk function $R(f)$
2. Solve the problem:

$$\min_{f \in \mathcal{H}} \left\{ R(f) + \lambda \| f \|_{\mathcal{H}}^2 \right\} .$$

λ controls the trade-off between fitting the data and being a smooth function.



Learning with kernels: Summary

- Feature point of view: A kernel is an inner product with respect to particular features.
- Geometric point of view : A kernel defines an implicit geometry on the space of data, although data do not need to have any prior geometric/algebric structure
- Functional point of view : Kernel methods learn functions that tend to be smooth with respect to this geometry
- Kernel engineering is the problem of designing specific kernel for specific data and specific tasks. Good place to put prior knowledge!

Outline

1 Learning with kernels

2 Making kernels

3 Choosing and combining kernels

4 Conclusion

Example: supervised sequence classification

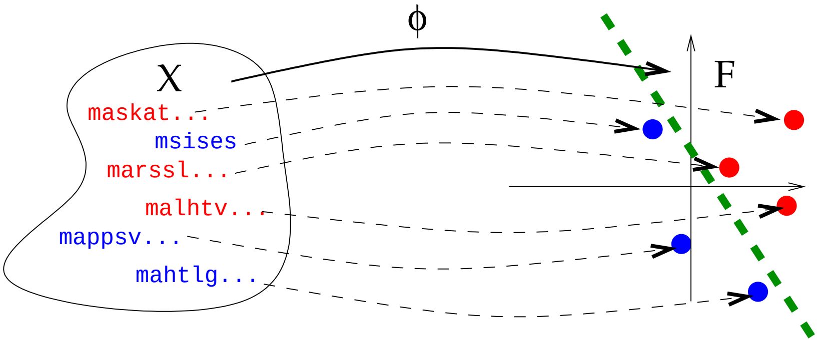
Data (training)

- Secreted proteins:
  MASKATLLLAFTLLFATCIARHQQRQQQQNQCQLQNIEA...
  MARSSLFTFLCLAVFINGCLSQIEQQSPWEFQGSEVW...
  MALHTVLIMLSLLPMLEAQNPEHANITIGEPITNETLGWL...
  ...
- Non-secreted proteins:
  MAPPSVFAEVPQAQPVLVFKLIADFREDPDPRKVNLGVG...
  MAHTLGLTQPNSTEPHKISFTAKEIDVIEWKGDILVVG...
  MSISESYAKEIKTAFRQFTDFPIEGEQFEDFLPIIGNP..
  ...

Goal

- Build a classifier to predict whether new proteins are secreted or not.

Kernel for biological sequences?



What is a GOOD kernels?

- Mathematically valid (?)
- Fast to compute
- Lead to good performances
- other?

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

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

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

Example 1: substring indexation

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)

Example 2: Mutual information kernels

- Parametric statistical model:

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

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

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

- See, e.g., Cuturi and V. (2004) for a fast mutual information kernel based on variable-length Markov models.

Sequence alignment

Motivation

How to compare 2 sequences?

$$\mathbf{x}_1 = \text{CGGSLIAMMWFGV}$$
$$\mathbf{x}_2 = \text{CLIVMMNRLMWFGV}$$

Find a good alignment:

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

Example 3: Local alignment kernel

Smith-Waterman score

- 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...

LA kernel

The local alignment kernel:

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

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

Example 3: Local alignment kernel

Smith-Waterman score

- 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...

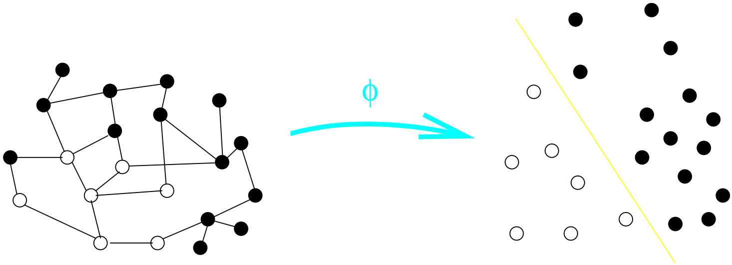
LA kernel

The local alignment kernel:

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

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

Example 4 : Kernel on a graph



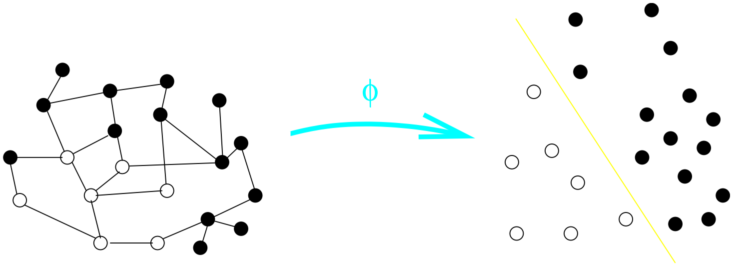
Laplacian-based kernel

The set $\mathcal{H} = \left\{ f \in \mathbb{R}^m : \sum_{i=1}^m f_i = 0 \right\}$ endowed with the norm:

$$\Omega(f) = \sum_{i \sim j} \left(f(\mathbf{x}_i) - f(\mathbf{x}_j) \right)^2$$

is a RKHS whose reproducing kernel is the pseudo-inverse of the graph Laplacian.

Example 4 : Kernel on a graph



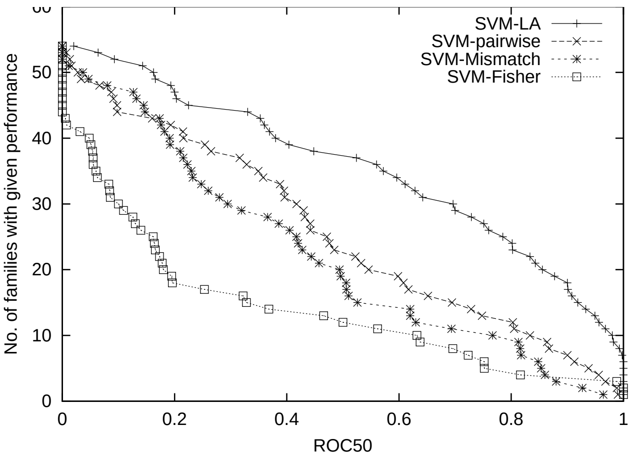
Laplacian-based kernel

The set $\mathcal{H} = \left\{ f \in \mathbb{R}^m : \sum_{i=1}^{m} f_i = 0 \right\}$ endowed with the norm:

$$\Omega(f) = \sum_{i \sim j} \left(f(\mathbf{x}_i) - f(\mathbf{x}_j) \right)^2$$

is a RKHS whose reproducing kernel is the pseudo-inverse of the graph Laplacian.

The choice of kernel makes a difference



Performance on the SCOP superfamily recognition benchmark.

Outline

1. Learning with kernels
2. Making kernels
3. Choosing and combining kernels
4. Conclusion

Motivation

- We can imagine plenty of kernels for a given application
- Which one to use?
- Perhaps we can combine them to make better than each one individually?

Example: sum kernels

- Consider p kernels $K_1, \ldots, K_p$
- Form the sum:

$$K = \sum_{i=1}^{p} K_i \,.$$

- Equivalently, work in the RKHS $\mathcal{H} = \mathcal{H}_1 \oplus \ldots \oplus \mathcal{H}_p$ with

$$\| f \|_{\mathcal{H}}^2 = \inf_{f = f_1 + \ldots + f_p} \sum_{i=1}^{p} \| f_i \|_{\mathcal{H}_i}^2 \,.$$

Example: multiple kernel learning (MKL)

- Form the convex combination:

$$K = \sum_{i=1}^{p} \eta_i K_i .$$

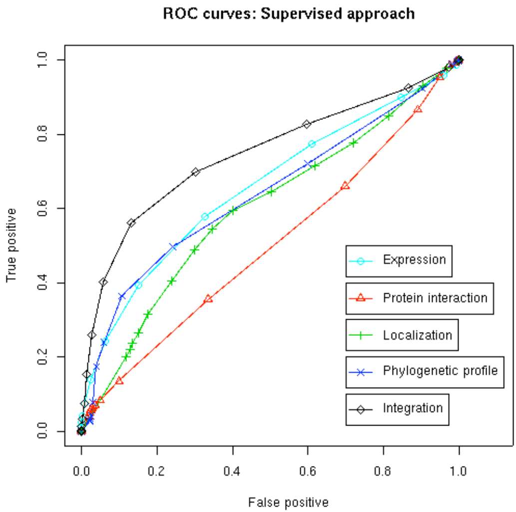
where the weights are chosen to minimize the following convex function under the constraint $tr(K) = 1$ (Lanckriet et al., 2003):

$$h(K) = \inf_{f \in \mathcal{H}_K} \left\{ R(f) + \lambda \| f \|_{\mathcal{H}_K} \right\}$$

- Equivalently, work in the RKHS $\mathcal{H} = \mathcal{H}_1 \oplus \ldots \oplus \mathcal{H}_p$ with non-Hilbertian group L_1 norm (Bach et al., 2004):

$$\| f \|_{\mathcal{H}} = \inf_{f = f_1 + \ldots + f_p} \sum_{i=1}^{p} \| f_i \|_{\mathcal{H}_i} .$$

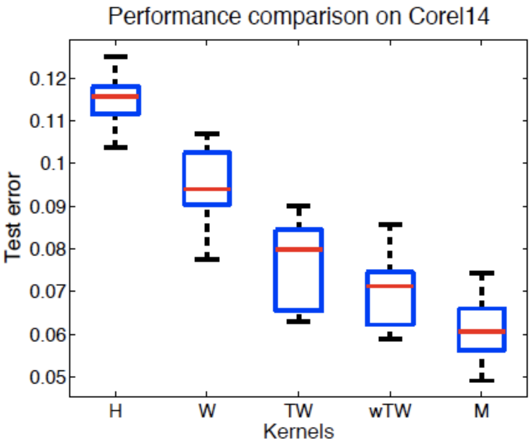
Application: gene network reconstruction



From Yamanishi et al., 2005.

Application: image classification

- Histogram kernels (**H**)
- Walk kernels (**W**)
- Tree-walk kernels (**TW**)
- Weighted tree-walks (**wTW**)
- MKL (**M**)



From Bach et al., 2007.

Outline

1 Learning with kernels

2 Making kernels

3 Choosing and combining kernels

4 Conclusion

Questions

- Kernel design: which principles? Which objective? Which criteria?
- Kernel selection / combination: same question + which algorithms?
- Kernel learning : where to go beyond linear combinations of pre-defined kernels?