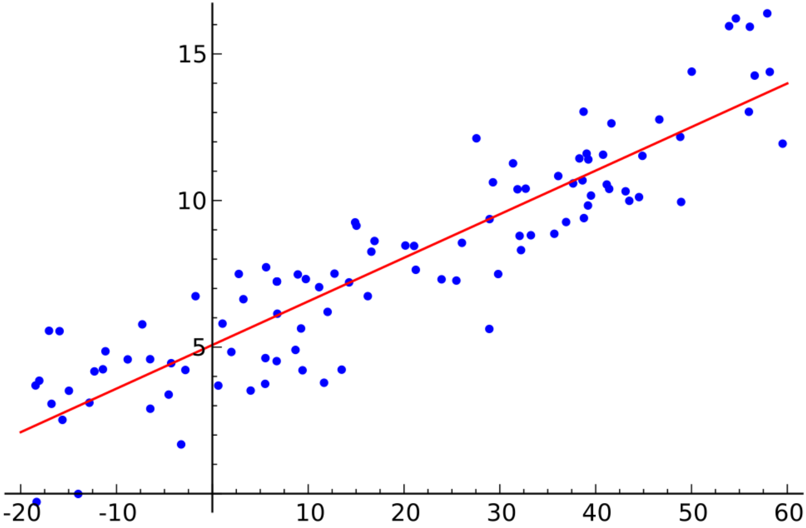
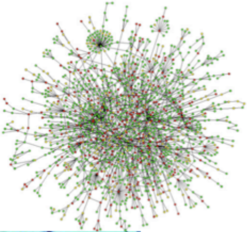
Kernel methods and computational biology

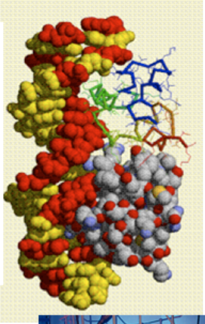
Jean-Philippe Vert
Jean-Philippe.Vert@mines.org

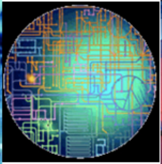
What we know how to solve



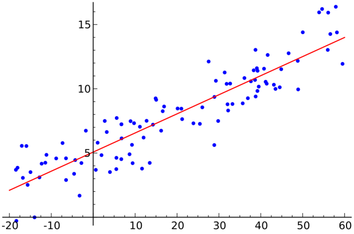
But real data are often more complicated...

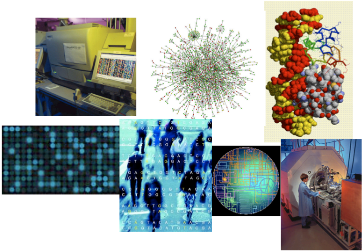






Main goal of this course





Extend well-understood, linear statistical learning techniques to nonlinear techniques for real-world, complicated, structured, high-dimensional data (images, texts, time series, graphs, distributions, permutations...)

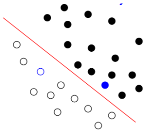
Outline

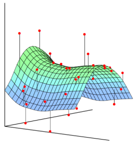
1. Penalized empirical risk minimization
2. Learning with ℓ_2 regularization
3. Kernel methods
4. Positive definite kernels and RKHS
5. Kernel examples
6. Learning molecular classifiers with network information
7. Data integration with kernels

Outline

1. Penalized empirical risk minimization
2. Learning with ℓ_2 regularization
3. Kernel methods
4. Positive definite kernels and RKHS
5. Kernel examples
6. Learning molecular classifiers with network information
7. Data integration with kernels

General learning framework





Input

- $\mathcal{X}$ the space of patterns or data (typically, $\mathcal{X} = \mathbb{R}^p$)
- $\mathcal{Y}$ the space of response or labels
  - Classification or pattern recognition : $\mathcal{Y} = \{-1, 1\}$
  - Regression : $\mathcal{Y} = \mathbb{R}$
- $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ a training set in $(\mathcal{X} \times \mathcal{Y})^n$

Output

- A function $f : \mathcal{X} \to \mathcal{Y}$ to predict the output associated to any new pattern $x \in \mathcal{X}$ by $f(x)$

Empirical risk minimization (ERM)

- Define $\mathcal{F}$ a class of functions $f : \mathcal{X} \to \mathcal{Y}$ (or $f : \mathcal{X} \to \mathbb{R}$)
- Define $\ell(t, y) \in \mathbb{R}$ the loss when we predict t and the true response is y
- For a candidate function $f \in \mathcal{F}$, its empirical risk is

$$R_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell\left(f(x_i), y_i\right)$$

- The ERM estimator is

$$\hat{f} \in \operatorname*{argmin}_{f \in \mathcal{F}} R_n(f)$$

Example: ordinary least squares (OLS)

- $\mathcal{X} = \mathbb{R}^p$
- $\mathcal{F}$ is the set of linear functions of the form

$$f_\beta(x) = \sum_{i=1}^{p} \beta_i x_i = x^\top \beta \quad \text{for} \quad \beta \in \mathbb{R}^p$$

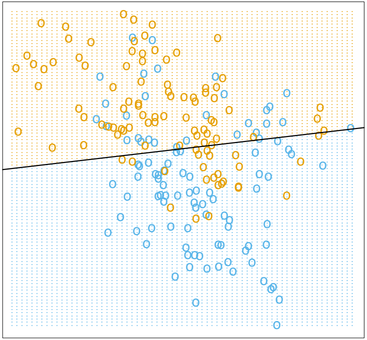
- $\ell(t, y) = (t - y)^2$ is the squared error
- The empirical risk is the mean squared error (MSE):

$$R_n(\beta) = \frac{1}{n} \sum_{i=1}^{n} (f_\beta(x_i) - y_i)^2 = \frac{1}{n} (\mathbf{Y} - \mathbf{X}\beta)^\top (\mathbf{Y} - \mathbf{X}\beta)$$

- When $\mathbf{X}^\top \mathbf{X}$ is non-singular, the ERM estimator is

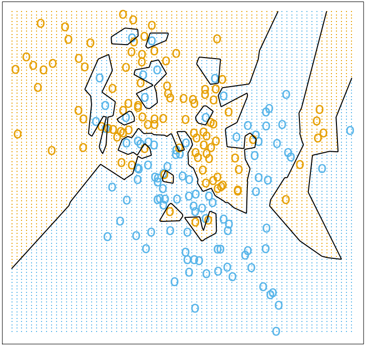
$$\hat{\beta} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{Y}$$

The curse of dimensionality



Small dimension

(Hastie et al. The elements of statistical learning. Springer, 2001.)



Large dimension

In high dimensions, ERM overfits the data and gives poor estimators, even for simple linear models.

Solution: penalized ERM (aka shrinkage estimators)

- Define $\Omega : \mathcal{F} \to \mathbb{R}$ a penalty function
- The penalized ERM estimator is

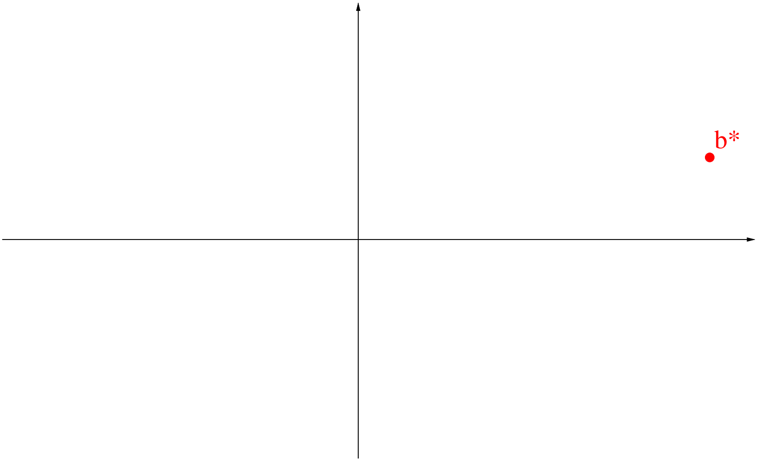
$$\hat{f} \in \operatorname*{argmin}_{f \in \mathcal{F}} R_n(f) \quad \text{such that} \quad \Omega(f) \leq C$$

or

$$\hat{f} \in \operatorname*{argmin}_{f \in \mathcal{F}} \{R_n(f) + \lambda \Omega(f)\}$$

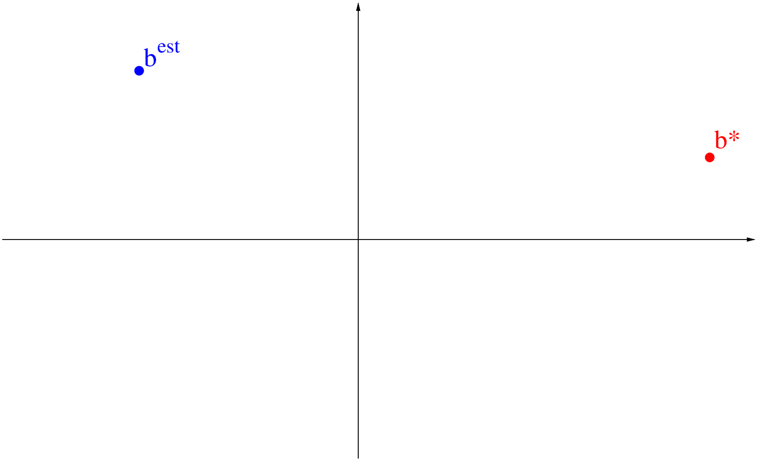
Why skrinkage classifiers?

$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C\,.$$



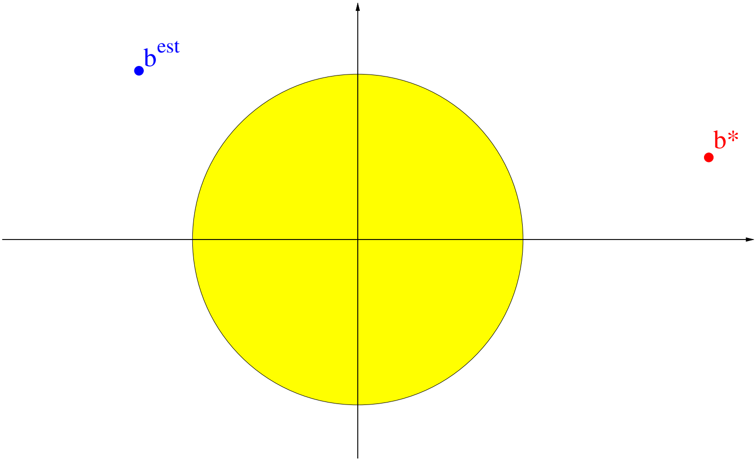
Why skrinkage classifiers?

$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C\,.$$



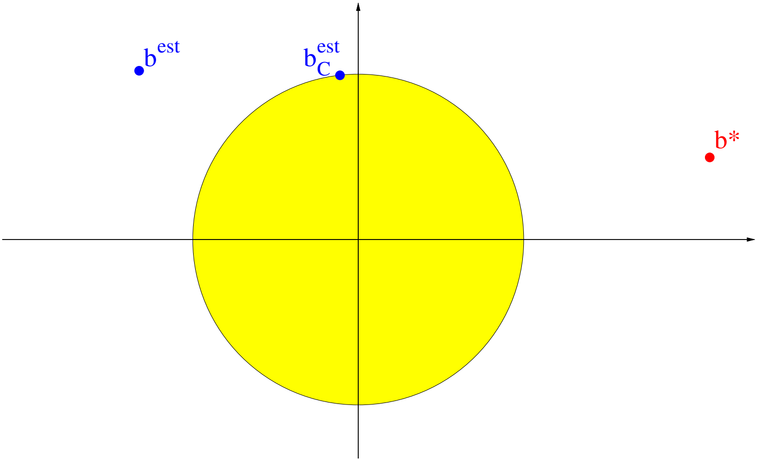
Why skrinkage classifiers?

$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C\,.$$



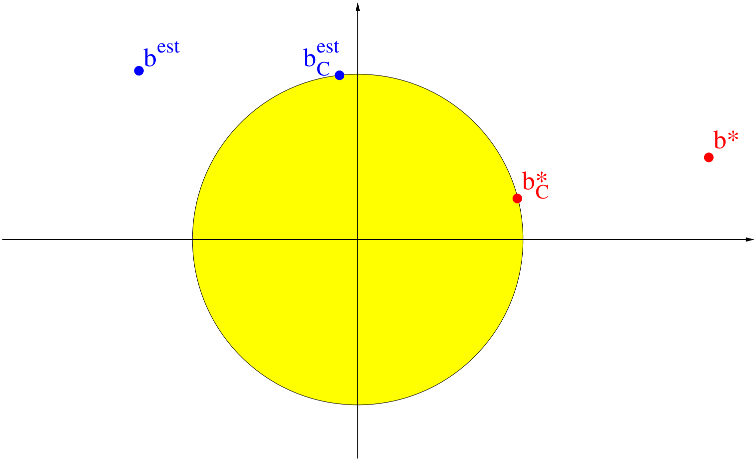
Why skrinkage classifiers?

$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C\,.$$



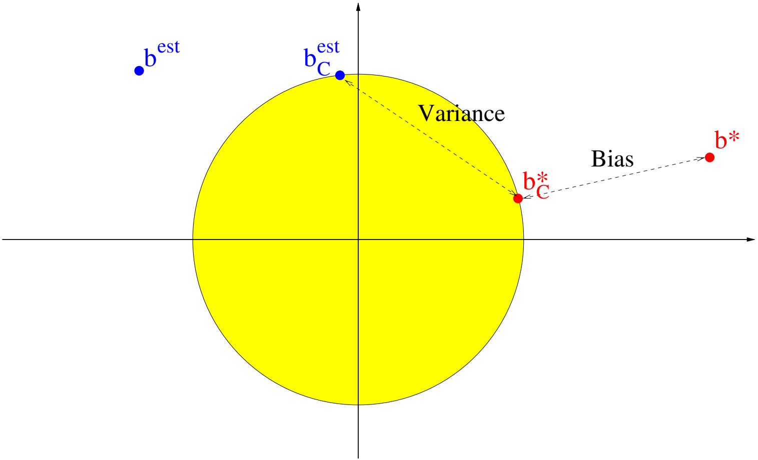
Why skrinkage classifiers?

$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C\,.$$



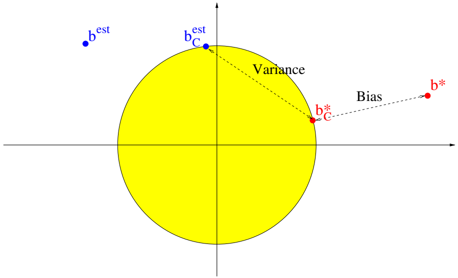
Why skrinkage classifiers?

$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C\,.$$



Why skrinkage classifiers?

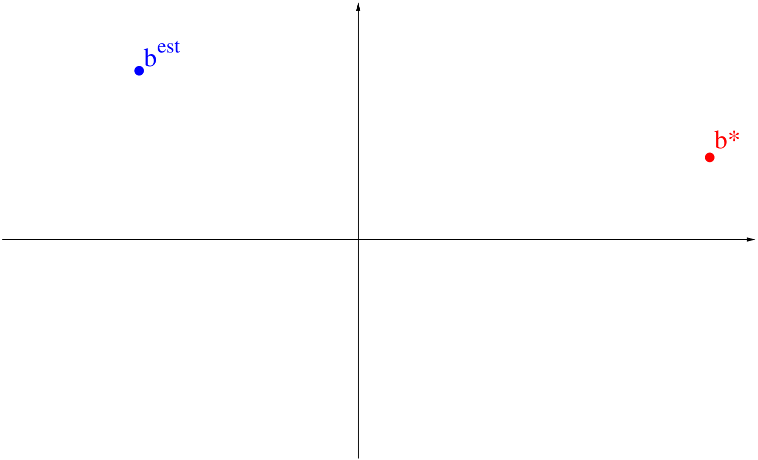
$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C\,.$$



- "Increase bias but decrease variance"
- Variance dominates in high dimension

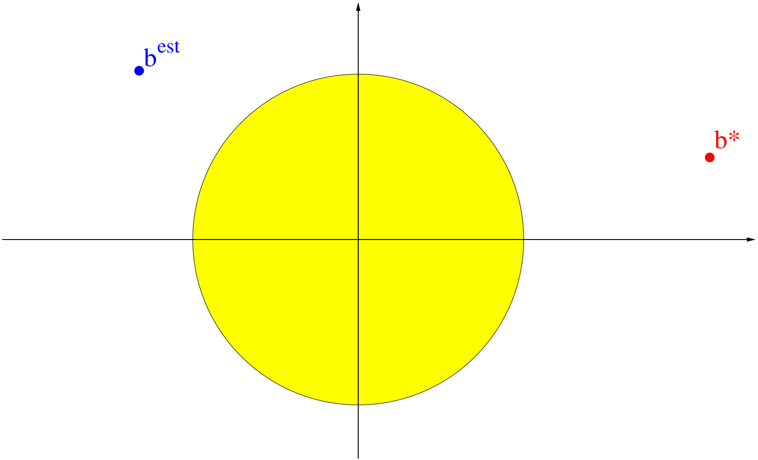
Choice of Ω can decrease the bias

$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \le C\,.$$



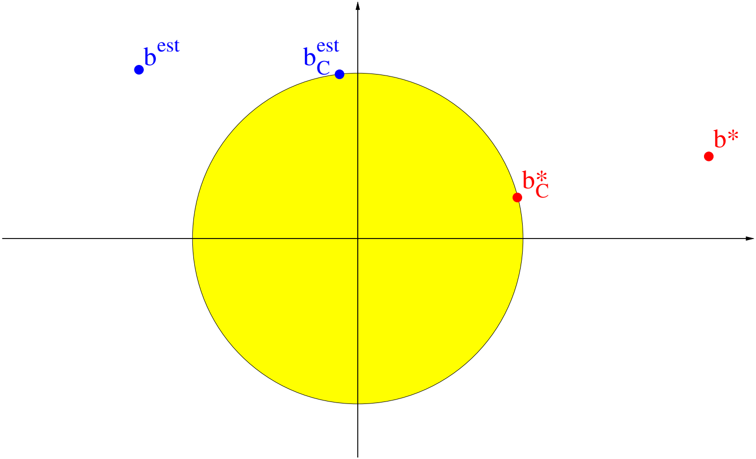
Choice of Ω can decrease the bias

$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C\,.$$



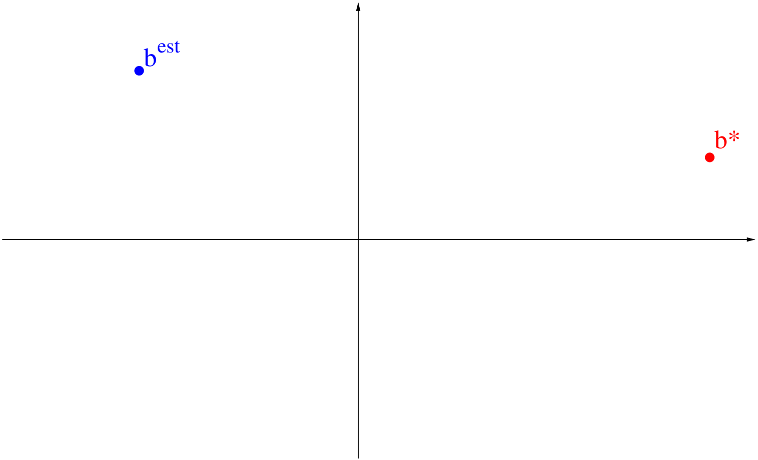
Choice of Ω can decrease the bias

$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C\,.$$



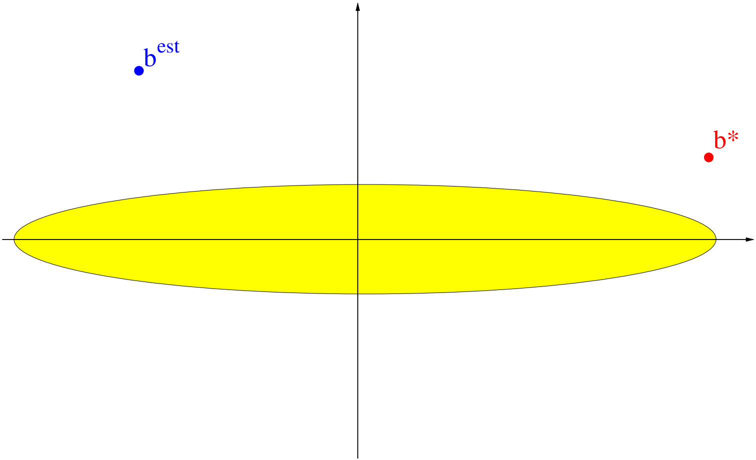
Choice of Ω can decrease the bias

$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C\,.$$



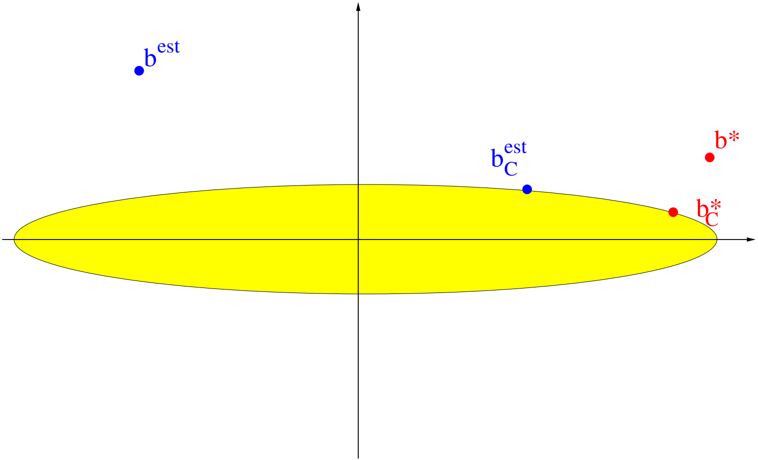
Choice of Ω can decrease the bias

$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C\,.$$

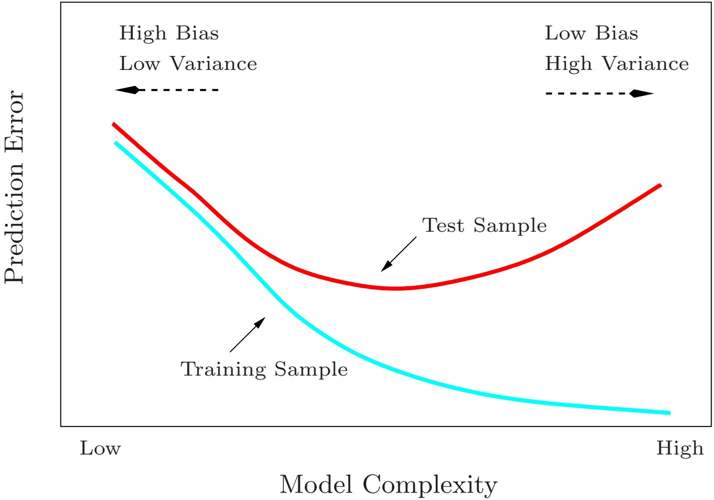


Choice of Ω can decrease the bias

$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C\,.$$



Choice of C or λ



(Hastie et al. The elements of statistical learning. Springer, 2001.)

Cross-validation

A simple and systematic procedure to estimate the risk (and to optimize the model's parameters)

1. Randomly divide the training set (of size n) into K (almost) equal portions, each of size K/n
2. For each portion, fit the model with different parameters on the $K-1$ other groups and test its performance on the left-out group
3. Average performance over the K groups, and take the parameter with the smallest average performance.

Taking $K = 5$ or 10 is recommended as a good default choice.

Outline

General setting

- $\mathcal{X} = \mathbb{R}^p$
- $\mathcal{F}$ the set of linear functions $f_\beta(x) = x^\top \beta$
- Penalty $\Omega(\beta) = \beta^\top \beta = \| \beta \|^2$
- A general ℓ_2-penalized estimator is of the form

$$\min_{\beta} \left\{ R(\beta) + \lambda \|\beta\|_2^2 \right\} , \tag{1}$$

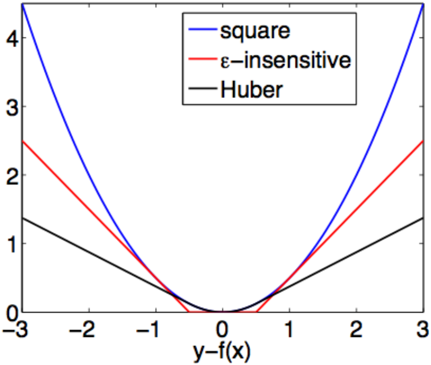
where

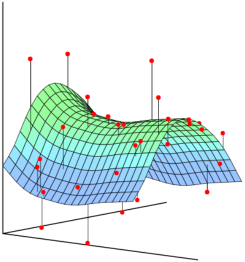
$$R(\beta) = \frac{1}{n} \sum_{i=1}^{n} \ell(f_\beta(x_i), y_i)$$

for some general loss functions ℓ.

Loss for regression

- Square loss : $\ell(u, y) = (u - y)^2$
- ϵ-insensitive loss : $\ell(u, y) = (|\, u - y \,| - \epsilon)_+$
- Huber loss : mixed quadratic/linear





Example: Ridge regression (Hoerl and Kennard, 1970)

For $\mathcal{Y} = \mathbb{R}$, take the squared error loss

$$\ell(t, y) = (t - y)^2 \,.$$

Then:

$$\begin{aligned} R(\beta) + \lambda\Omega(\beta) &= \frac{1}{n}\sum_{i=1}^{n} \left(f_\beta\left(x_i\right) - x_i\right)^2 + \lambda \sum_{i=1}^{p} \beta_i^2 \\ &= \frac{1}{n}\left(Y - X\beta\right)^\top \left(Y - X\beta\right) + \lambda\beta^\top\beta \,. \end{aligned}$$

Explicit minimizer:

$$\hat{\beta}_\lambda^{\text{ridge}} = \arg\min_{\beta\in\mathbb{R}^p} \left\{R(\beta) + \lambda\Omega(\beta)\right\} = \left(X^\top X + \lambda n I\right)^{-1} X^\top Y \,.$$

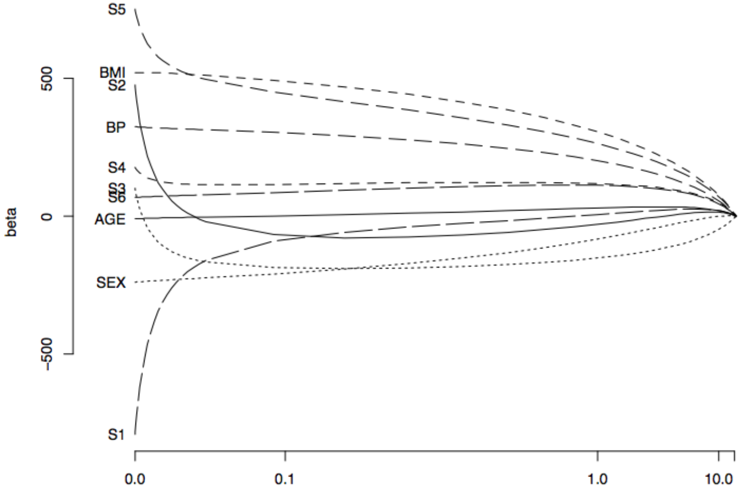
Limit cases

$$\hat{\beta}_\lambda^{\text{ridge}} = \left(X^\top X + \lambda n I\right)^{-1} X^\top Y$$

Corollary

- As $\lambda \to 0$, $\hat{\beta}_\lambda^{\text{ridge}} \to \hat{\beta}^{\text{OLS}}$ (low bias, high variance).
- As $\lambda \to +\infty$, $\hat{\beta}_\lambda^{\text{ridge}} \to 0$ (high bias, low variance).

Ridge regression example



(From Hastie et al., 2001)

Ridge regression with correlated features

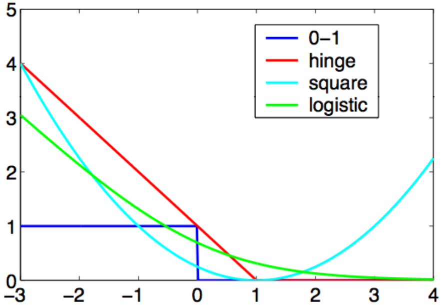
Ridge regression is particularly useful in the presence of correlated features:

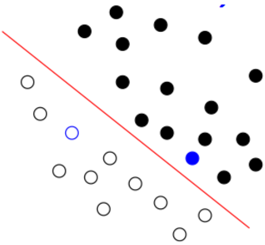
```
> library(MASS) # for the lm.ridge command
> x1 <- rnorm(20)
> x2 <- rnorm(20,mean=x1,sd=.01)
> y <- rnorm(20,mean=3+x1+x2)
> lm(y~x1+x2)$coef
(Intercept)          x1          x2
   3.070699   25.797872  -23.748019
> lm.ridge(y~x1+x2,lambda=1)
                  x1        x2
3.066027 1.015862 0.956560
```

Loss for pattern recognition

Large margin classifiers

- For pattern recognition $\mathcal{Y} = \{-1, 1\}$
- Estimate a function $f : \mathcal{X} \to \mathbb{R}$.
- The margin of the function f for a pair (x, y) is: $yf(x)$.
- The loss function is usually a decreasing function of the margin : $\ell(f(x), y) = \phi(yf(x))$,





Example: Ridge logistic regression (Le Cessie and van Houwelingen, 1992)

$$\ell_{\text{logistic}}(u, y) = \ln\left(1 + e^{-yu}\right)$$

$$\min_{\beta} J(\beta) = \frac{1}{n}\sum_{i=1}^{n} \ln\left(1 + e^{-y_i \beta^\top x_i}\right) + \lambda \|\beta\|_2^2$$

Probabilistic interpretation

$$\min_{\beta} J(\beta) = \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + e^{-y_i \beta^\top x_i}\right) + \lambda \|\beta\|_2^2$$

Exercice

Show that ridge logistic regression finds the penalized maximum likelihood estimator:

$$\max_{\beta} \frac{1}{n} \sum_{i=1}^{n} \ln P_\beta(Y = y_i \,|\, X = x_i) - \lambda \|\beta\|_2^2 \,,$$

for the following model:

$$\begin{cases} P_\beta(Y = 1 \,|\, X = x) = \frac{e^{\beta^\top x}}{1 + e^{\beta^\top x}} \\ P_\beta(Y = -1 \,|\, X = x) = \frac{1}{1 + e^{\beta^\top x}} \end{cases}$$

Solving ridge logistic regression

$$\min_{\beta} J(\beta) = \frac{1}{n}\sum_{i=1}^{n} \ln\left(1 + e^{-y_i \beta^\top x_i}\right) + \lambda \|\beta\|_2^2$$

No explicit solution, but convex problem with:

$$\begin{aligned}
\nabla_\beta J(\beta) &= -\frac{1}{n}\sum_{i=1}^{n} \frac{y_i x_i}{1 + e^{y_i \beta^\top x_i}} + 2\lambda\beta \\
&= -\frac{1}{n}\sum_{i=1}^{n} y_i \left[1 - P_\beta(y_i \mid x_i)\right] x_i + 2\lambda\beta \\
\nabla_\beta^2 J(\beta) &= \frac{1}{n}\sum_{i=1}^{n} \frac{x_i x_i^\top e^{y_i \beta^\top x_i}}{\left(1 + e^{y_i \beta^\top x_i}\right)^2} + 2\lambda I \\
&= \frac{1}{n}\sum_{i=1}^{n} P_\beta(1 \mid x_i)\left(1 - P_\beta(1 \mid x_i)\right) x_i x_i^\top + 2\lambda I
\end{aligned}$$

Solving ridge logistic regression (cont.)

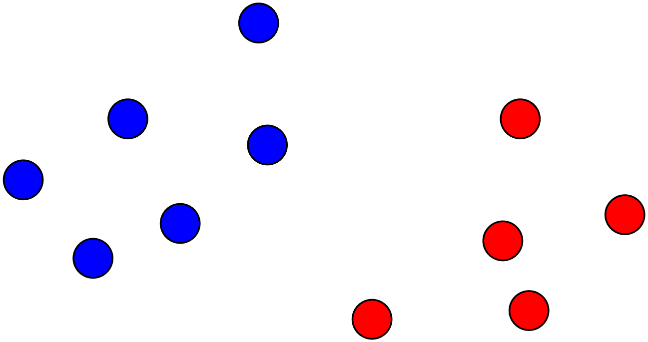
$$\min_\beta J(\beta) = \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + e^{-y_i \beta^\top x_i}\right) + \lambda \|\beta\|_2^2$$

- The solution can then be found by Newton-Raphson iterations:

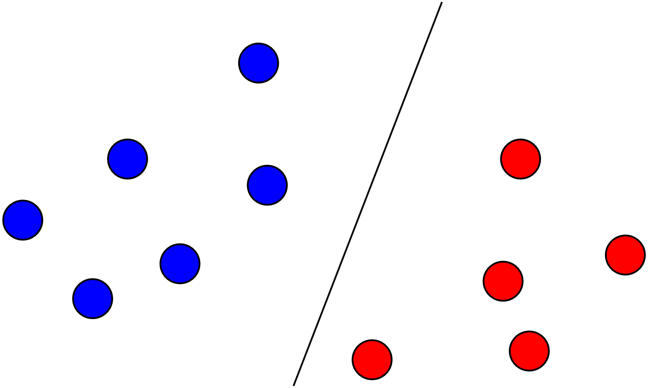
$$\beta^{new} \leftarrow \beta^{old} - \left[\nabla_\beta^2 J\left(\beta^{old}\right)\right]^{-1} \nabla_\beta J\left(\beta^{old}\right) .$$

- Each step is equivalent to solving a weighted ridge regression problem (emphleft as exercise)
- This method is therefore called iteratively reweighted least squares (IRLS).

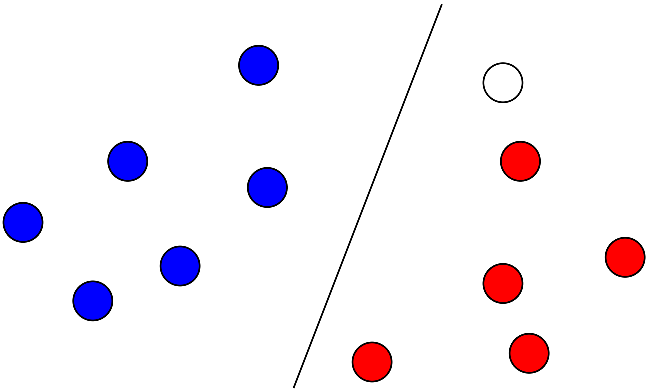
Example: hard-margin SVM



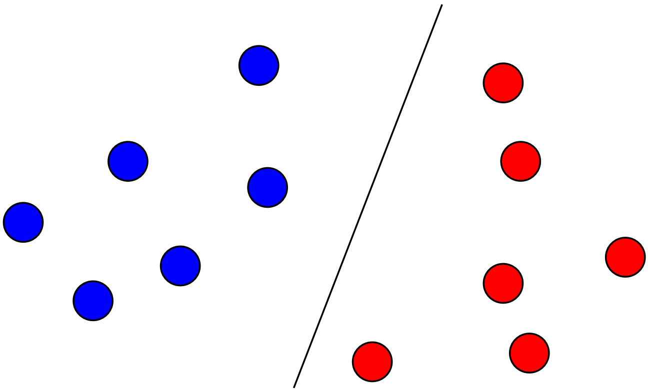
Example: hard-margin SVM



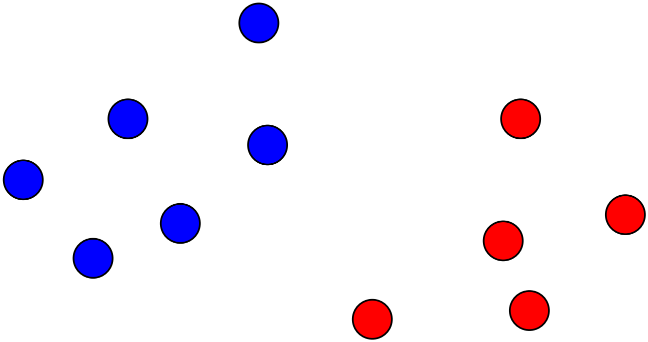
Example: hard-margin SVM



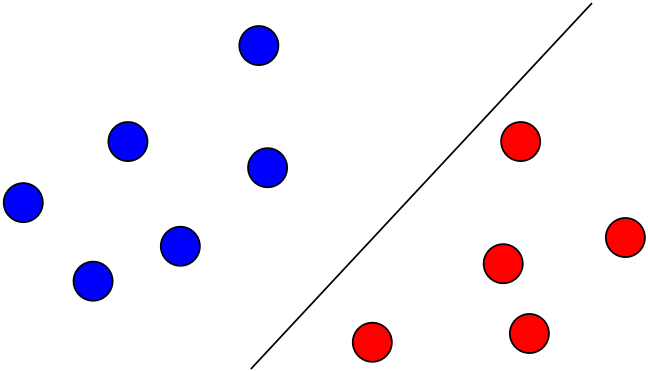
Example: hard-margin SVM



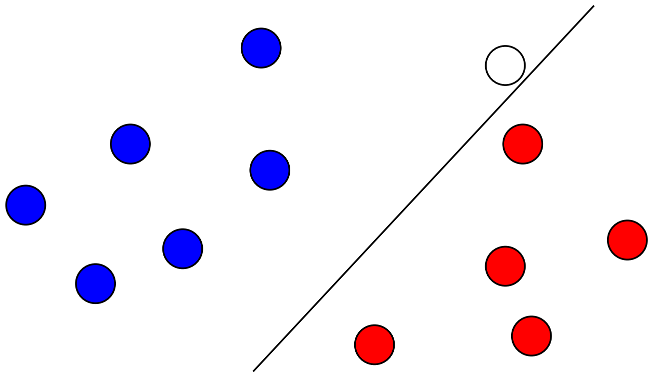
Example: hard-margin SVM



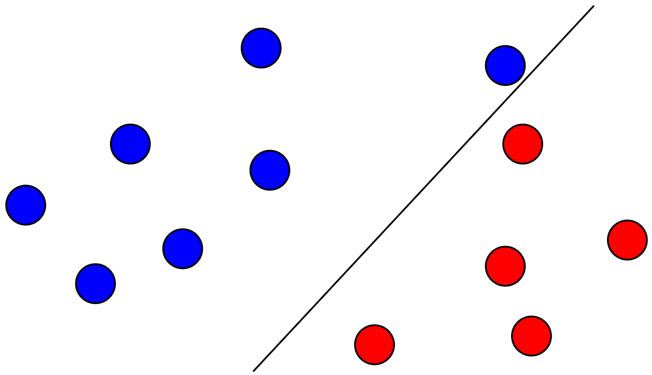
Example: hard-margin SVM



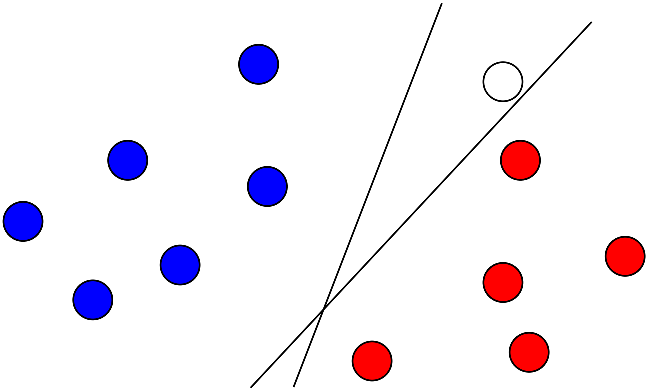
Example: hard-margin SVM



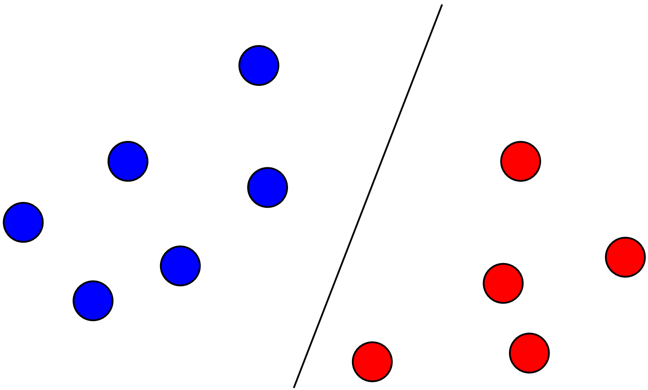
Example: hard-margin SVM



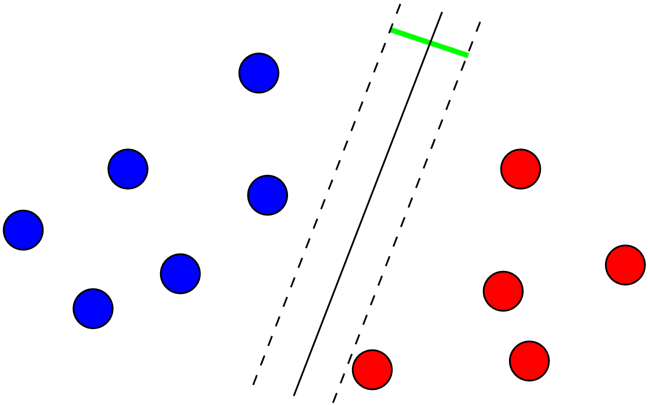
Example: hard-margin SVM



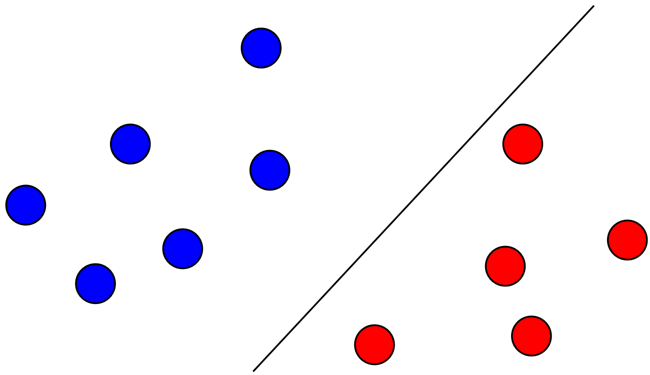
Example: hard-margin SVM



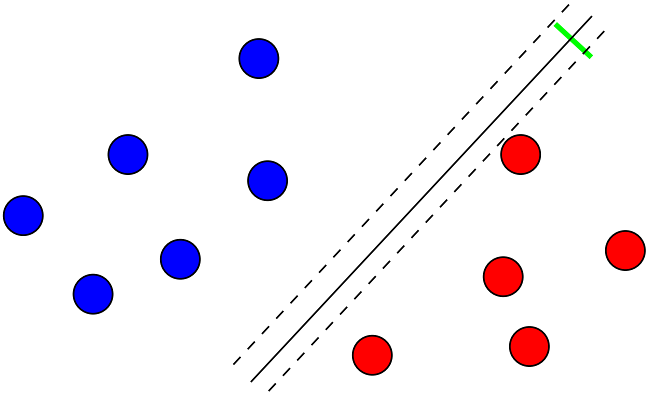
Example: hard-margin SVM



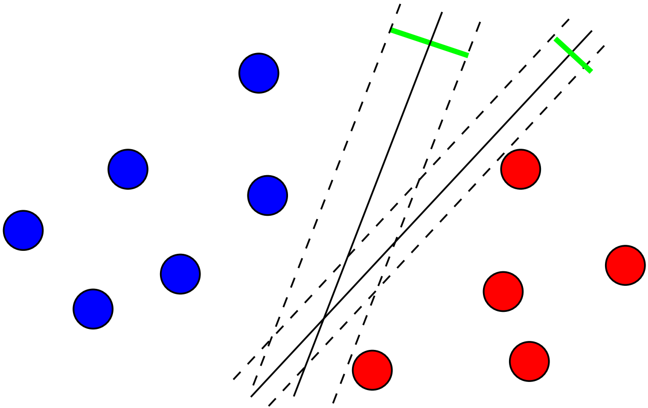
Example: hard-margin SVM



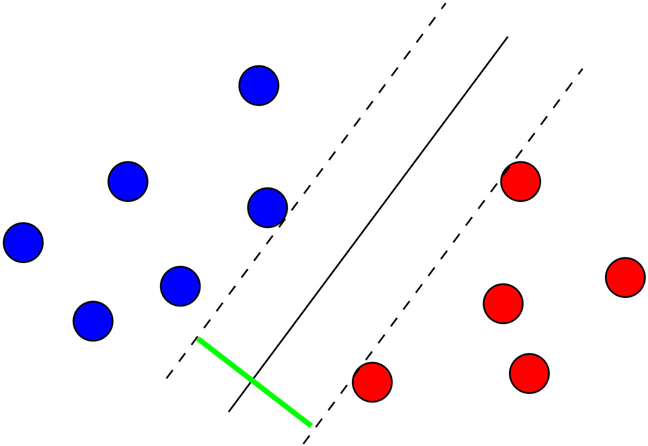
Example: hard-margin SVM



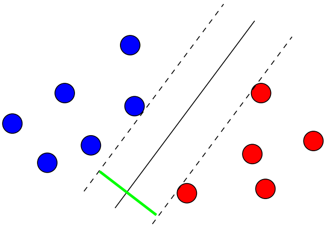
Example: hard-margin SVM



Example: hard-margin SVM



Hard-margin SVM is an ℓ_2-regularized method



Exercice

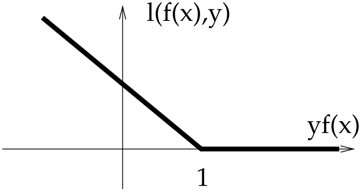
Show that hard-margin SVM solve a problem of the form

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \left\{ \sum_{i=1}^{n} \phi_{\text{hard}} \left(y_i f_\beta(x_i) \right) + \lambda \| \beta \|_2^2 \right\} .$$

What is ϕ_{hard}?

Example: (soft-margin) SVM

- The hinge loss

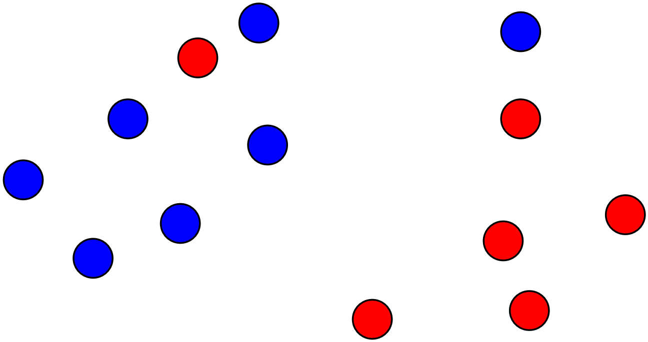


$$\phi_{\text{hinge}}(u) = \max(1-u, 0) = \begin{cases} 0 & \text{if } u \geq 1, \\ 1-u & \text{otherwise.} \end{cases}$$

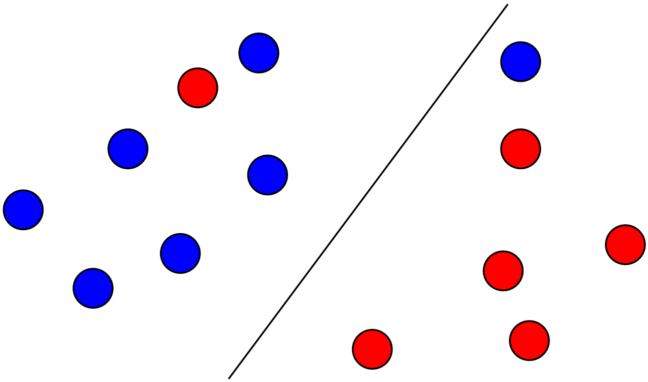
- SVM solves:

$$\min_{\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} \phi_{\text{hinge}}\left(y_i f_{\beta}\left(x_i\right)\right) + \lambda \| \beta \|_2^2 \right\} .$$

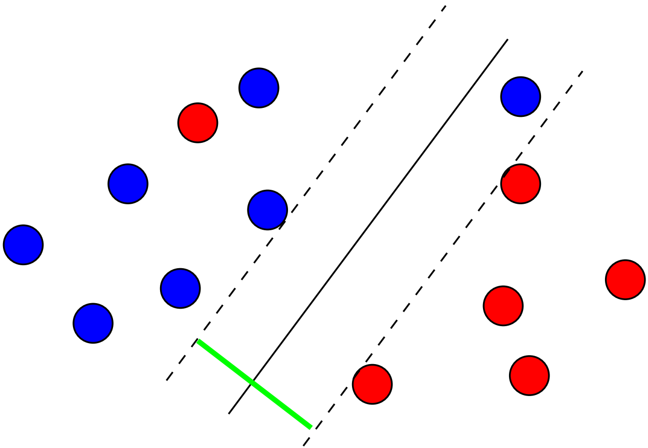
SVM: graphical interpretation



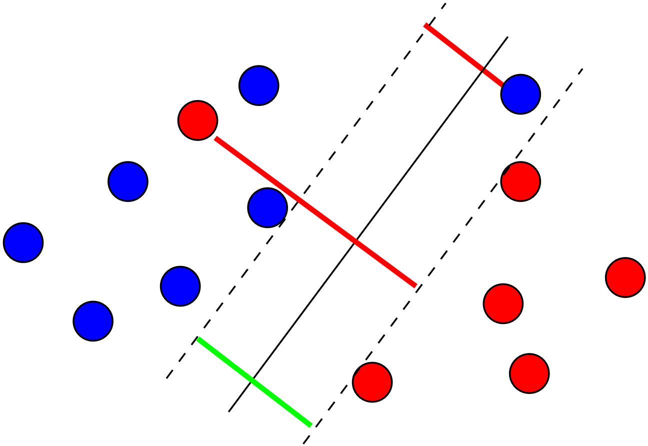
SVM: graphical interpretation



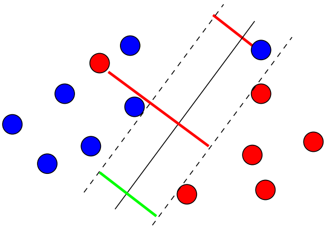
SVM: graphical interpretation



SVM: graphical interpretation



SVM: graphical interpretation



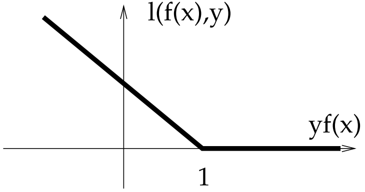
Exercice

Show that SVM finds a trade-off between large margin and few errors, by minimizing a function of the form:

$$\min_f \left\{ \frac{1}{margin(f)} + \gamma \times errors(f) \right\}$$

Explicit γ and $errors(f)$.

SVM reformulation as a quadratic program (QP)



- Note that for any $u \in \mathbb{R}$,

$$\phi_{\text{hinge}}(u) = \min_{\xi \in \mathbb{R}} \xi \quad \text{such that} \quad \begin{cases} \xi \geq 0 \\ \xi \geq 1 - u \end{cases}$$

- Therefore SVM solves the QP

$$\min_{\beta \in \mathbb{R}^p, \xi \in \mathbb{R}^n} \left\{ \frac{1}{n} \sum_{i=1}^{n} \xi_i + \lambda \| \beta \|_2^2 \right\} \quad \text{s. t. } \forall i \in [1, n], \quad \begin{cases} \xi_i \geq 0 \\ \xi_i \geq 1 - y_i x_i^\top \beta \end{cases}$$

Dual formulation

Form the Lagrangian:

$$L(\beta, \xi, \alpha, \gamma) = \frac{1}{2n\lambda} \sum_{i=1}^{n} \xi_i + \frac{1}{2} \| \beta \|_2^2 - \sum_{i=1}^{n} \alpha_i \left(y_i x_i^\top \beta + \xi_i - 1 \right) - \gamma^\top \xi$$

Minimize in the primal variables (β, ξ):

$$\nabla_\beta L = \beta - \sum_{i=1}^{n} \alpha_i y_i x_i \quad \Longrightarrow \quad \beta = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\nabla_{\xi_i} L = \frac{1}{2n\lambda} - \alpha_i - \gamma_i \quad \Longrightarrow \quad \alpha_i + \gamma_i = \frac{1}{2n\lambda}$$

Dual problem, with $C = \frac{1}{2n\lambda}$:

$$\max_{0 \leq \alpha \leq C} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^\top x_j \right\}$$

SVM with affine function (exercice)

$$f_{\beta,b}(x) = \beta^\top x + b\,, \qquad \min_{\beta,b} \left\{ \frac{1}{n} \sum_{i=1}^n \phi_{\text{hinge}}\left(y_i f_{\beta,b}(x_i)\right) + \lambda \| \beta \|_2^2 \right\}$$

is equivalent to

$$\max_{\alpha \in \mathbb{R}^n} \left\{ \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^\top x_j \right\}$$

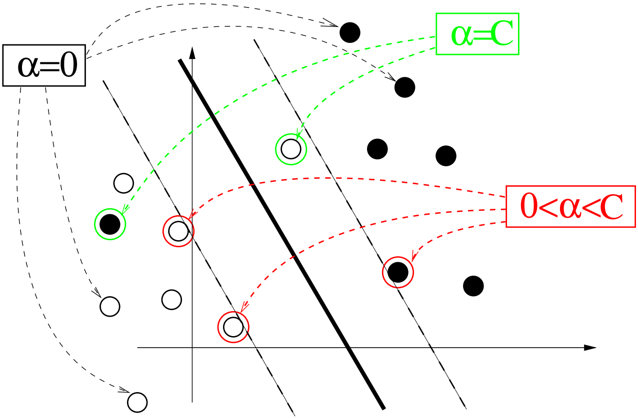
under the constraints:

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

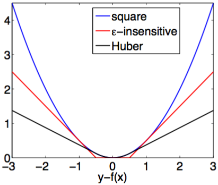
and the solution is

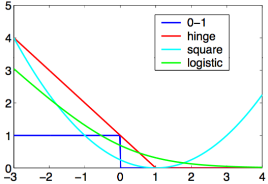
$$f(x) = \sum_{i=1}^n \alpha_i y_i x_i^\top x + b^*\,.$$

Interpretation: support vectors ($C = 1/2n\lambda$)



Summary: ℓ_2-regularized linear methods



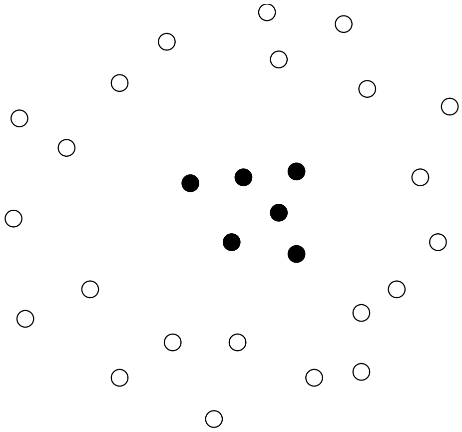


$$f_\beta(x) = \beta^\top x \,, \quad \min_\beta \left\{ \frac{1}{n} \sum_{i=1}^n \ell(f_\beta(x_i), y_i) + \lambda \|\beta\|_2^2 \right\}$$

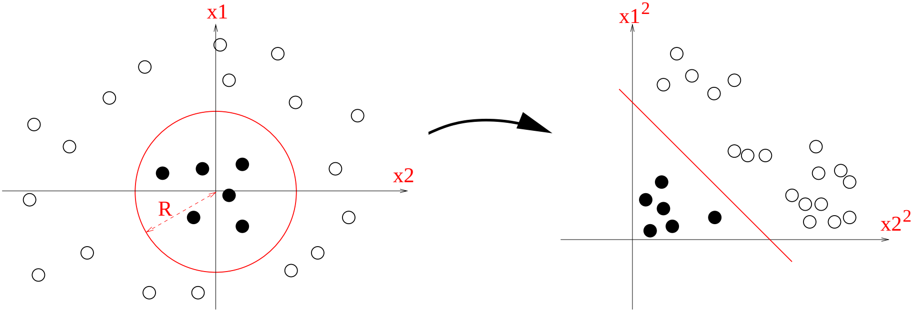
- Many popular methods for regression and classification are obtained by changing the loss function: ridge regression, logistic regression, SVM...
- Needs to solve numerically a convex optimization problem, well adapted to large datasets (stochastic gradient...)
- In practice, very similar performance between the different variants in general

Outline

Sometimes linear methods are not interesting



Solution: nonlinear mapping to a feature space



For $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ let $\Phi(x) = \begin{pmatrix} x_1^2 \\ x_2^2 \end{pmatrix}$. The decision function is:

$$f(x) = x_1^2 + x_2^2 - R^2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}^\top \begin{pmatrix} x_1^2 \\ x_2^2 \end{pmatrix} - R^2 = \beta^\top \Phi(x) + b\,.$$

Kernel = inner product in the feature space

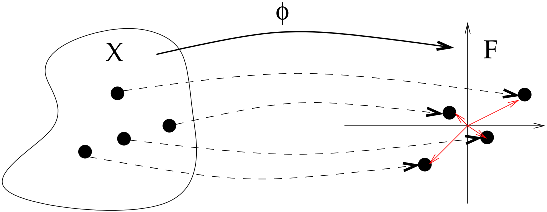
Definition

For a given mapping

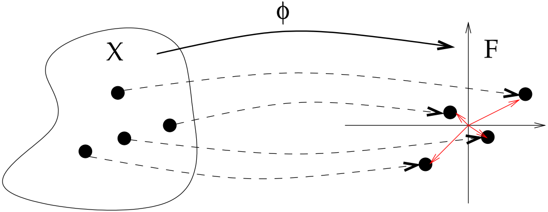
$$\Phi : \mathcal{X} \mapsto \mathcal{H}$$

from the space of objects $\mathcal{X}$ to some Hilbert space of features $\mathcal{H}$, the kernel between two objects x and x' is the inner product of their images in the features space:

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



Example

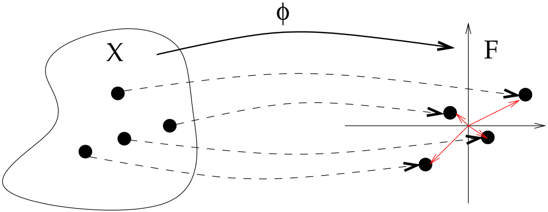


Let $\mathcal{X} = \mathcal{H} = \mathbb{R}^2$ and for $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ let $\Phi(x) = \begin{pmatrix} x_1^2 \\ x_2^2 \end{pmatrix}$

Then

$$K(x, x') = \Phi(x)^\top \Phi(x') = (x_1)^2 (x_1')^2 + (x_2)^2 (x_2')^2 \,.$$

The kernel tricks



2 tricks

1. Many linear algorithms (in particular ℓ_2-regularized methods) can be performed in the feature space of $\Phi(x)$ without explicitly computing the images $\Phi(x)$, but instead by computing kernels $K(x, x')$.
2. It is sometimes possible to easily compute kernels which correspond to complex large-dimensional feature spaces: $K(x, x')$ is often much simpler to compute than $\Phi(x)$ and $\Phi(x')$

Trick 1 illustration: SVM in the original space

- Train the SVM by maximizing

$$\max_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \,,$$

under the constraints:

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

- Predict with the decision function

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} + b^* \,.$$

Trick 1 illustration: SVM in the feature space

- Train the SVM by maximizing

$$\max_{\alpha\in\mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \Phi\left(x_i\right)^\top \Phi\left(x_j\right) ,$$

  under the constraints:

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

- Predict with the decision function

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

Trick 1 illustration: SVM in the feature space with a kernel

- Train the SVM by maximizing

$$\max_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K\left(x_i, x_j\right) ,$$

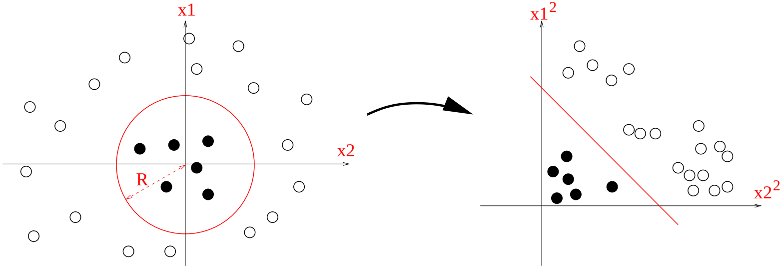
under the constraints:

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

- Predict with the decision function

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

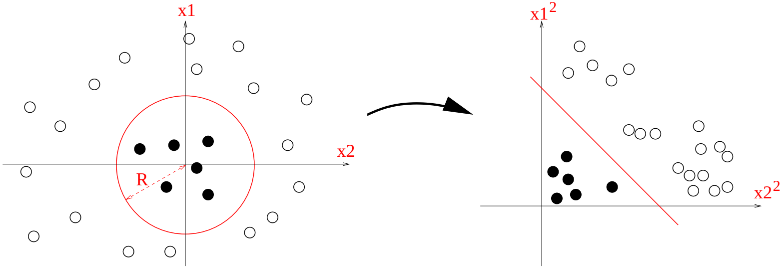
Trick 2 illustration: polynomial kernel



For $x = (x_1, x_2)^\top \in \mathbb{R}^2$, let $\Phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \in \mathbb{R}^3$:

$$\begin{aligned} K(x, x') &= x_1^2 x_1'^2 + 2x_1x_2x_1'x_2' + x_2^2x_2'^2 \\ &= \left(x_1x_1' + x_2x_2'\right)^2 \\ &= \left(x^\top x'\right)^2 . \end{aligned}$$

Trick 2 illustration: polynomial kernel



More generally, for $x, x' \in \mathbb{R}^p$,

$$K(x, x') = \left(x^\top x' + 1\right)^d$$

is an inner product in a feature space of all monomials of degree up to d *(left as exercice.)*

Combining tricks: learn a polynomial discrimination rule with SVM

- Train the SVM by maximizing

$$\max_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left(x_i^\top x_j + 1 \right)^d ,$$

under the constraints:

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

- Predict with the decision function

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

Illustration: toy nonlinear problem

```
> plot(x,col=ifelse(y>0,1,2),pch=ifelse(y>0,1,2))
```

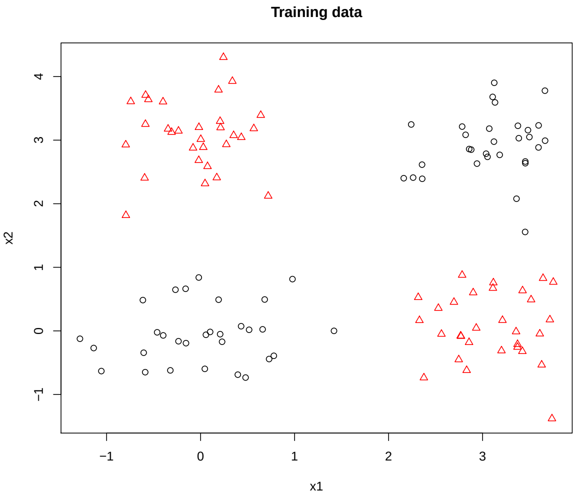


Illustration: toy nonlinear problem, linear SVM

```
> library(kernlab)
> svp <- ksvm(x,y,type="C-svc",kernel='vanilladot')
> plot(svp,data=x)
```

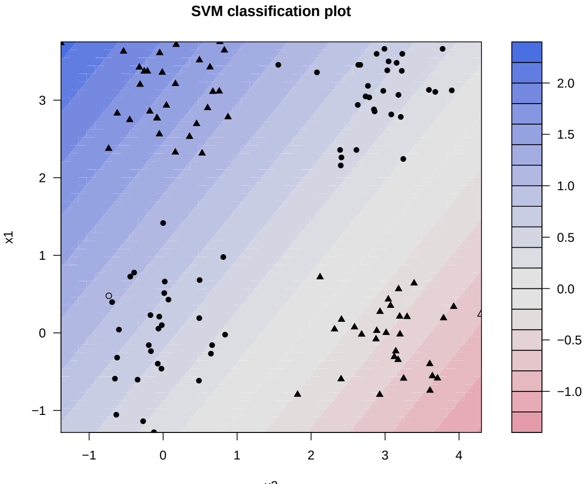
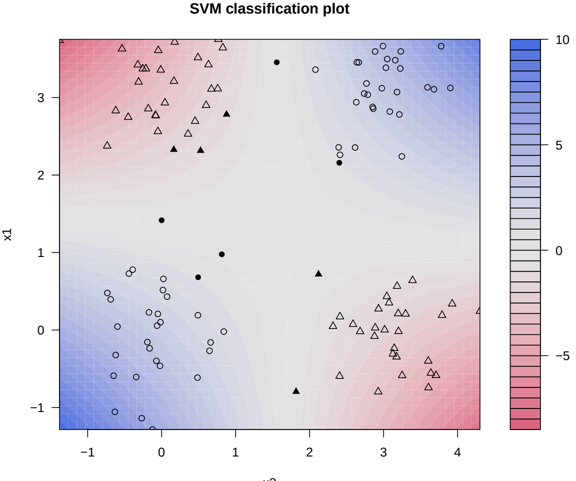


Illustration: toy nonlinear problem, polynomial SVM

```
> svp <- ksvm(x,y,type="C-svc", ...
                        kernel=polydot(degree=2))
> plot(svp,data=x)
```



More generally: trick 1 for ℓ_2-regularized estimators

Representer theorem

Let $f_\beta(x) = \beta^\top \Phi(x)$. Then any solution $\hat{f}_\beta$ of

$$\min_\beta \frac{1}{n} \sum_{i=1}^{n} \ell(f_\beta(x_i), y_i) + \lambda \|\beta\|_2^2$$

can be expanded as

$$\hat{f}_\beta(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)\,,$$

where $\boldsymbol{\alpha} \in \mathbb{R}^n$ is a solution of:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \ell\left(\sum_{j=1}^{n} \alpha_j K(x_i, x_j), y_i \right) + \lambda \sum_{i,j=1}^{n} \alpha_i \alpha_j K(x_i, x_j)\,.$$

Representer theorem: proof

- For any $\beta \in \mathbb{R}^p$, decompose $\beta = \beta_\mathcal{S} + \beta_\perp$ where $\beta_\mathcal{S} \in \mathit{span}(\Phi(x_1), \dots, \Phi(x_n))$ and $\beta_\perp$ is orthogonal to it.
- On any point x_i of the training set, we have:

$$f_\beta(x_i) = \beta^\top \Phi(x_i) = \beta_\mathcal{S}^\top \Phi(x_i) + \beta_\perp^\top \Phi(x_i) = \beta_\mathcal{S}^\top \Phi(x_i) = f_{\beta_\mathcal{S}}(x_i)\,.$$

- On the other hand, we have $\| \beta \|_2^2 = \| \beta_\mathcal{S} \|_2^2 + \| \beta_\perp \|_2^2 \geq \| \beta_\mathcal{S} \|_2^2$, with strict inequality if $\beta_\perp \neq 0$.
- Consequently, $\beta_\mathcal{S}$ is always as good as β in terms of objective function, and strictly better if $\beta_\perp \neq 0$. This implies that at any minimum, $\beta_\perp = 0$ and therefore $\beta = \beta_\mathcal{S} = \sum_{i=1}^n \alpha_i \Phi(x_i)$ for some $\boldsymbol{\alpha} \in \mathbb{R}^n$.
- We then just replace β by this expression in the objective function, noting that

$$\|\beta\|_2^2 = \| \sum_{i=1}^n \alpha_i \Phi(x_i) \|_2^2 = \sum_{i,j=1}^n \alpha_i \alpha_j \Phi(x_i)^\top \Phi(x_j) = \sum_{i,j=1}^n \alpha_i \alpha_j K(x_i, x_j)\,.$$

Example: kernel ridge regression

- Let $\Phi : \mathcal{X} \to \mathbb{R}^p$ be a feature mapping from the space of data to a Euclidean or Hilbert space.
- Let $f_\beta(x) = \beta^\top \Phi(x)$ and K the corresponding kernel.
- By the representer theorem, any solution of:

$$\hat{f} = \arg\min_{f_\beta} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - f_\beta\left(x_i\right)\right)^2 + \lambda \| \beta \|_2^2$$

can be expanded as:

$$\hat{f} = \sum_{i=1}^{n} \alpha_i K\left(x_i, x\right).$$

Example: kernel ridge regression

- Let $Y = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$ the vector of response variables.
- Let $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^\top \in \mathbb{R}^n$ the unknown coefficients.
- Let K be the $n \times n$ Gram matrix: $K_{i,j} = K\left(x_i, x_j\right)$.
- We can then write in matrix form:

$$\left(\hat{f}\left(x_1\right), \dots, \hat{f}\left(x_n\right)\right)^\top = K\boldsymbol{\alpha},$$

- Moreover,

$$\| \beta \|_2^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) = \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}.$$

Example: kernel ridge regression

- The problem is therefore equivalent to:

$$\arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \frac{1}{n}\left(K\boldsymbol{\alpha} - Y\right)^\top \left(K\boldsymbol{\alpha} - Y\right) + \lambda\boldsymbol{\alpha}^\top K\boldsymbol{\alpha}\,.$$

- This is a convex and differentiable function of $\boldsymbol{\alpha}$. Its minimum can therefore be found by setting the gradient in $\boldsymbol{\alpha}$ to zero:

$$\begin{aligned} 0 &= \frac{2}{n}K\left(K\boldsymbol{\alpha} - Y\right) + 2\lambda K\boldsymbol{\alpha} \\ &= K\left[\left(K + \lambda n I\right)\boldsymbol{\alpha} - Y\right] \end{aligned}$$

Example: kernel ridge regression

- K being a symmetric matrix, it can be diagonalized in an orthonormal basis and $Ker(K) \perp Im(K)$.
- In this basis we see that $(K + \lambda nI)^{-1}$ leaves $Im(K)$ and $Ker(K)$ invariant.
- The problem is therefore equivalent to:

$$
\begin{aligned}
& (K + \lambda nI)\,\boldsymbol{\alpha} - Y \in Ker(K) \\
\Leftrightarrow & \boldsymbol{\alpha} - (K + \lambda nI)^{-1}\, Y \in Ker(K) \\
\Leftrightarrow & \boldsymbol{\alpha} = (K + \lambda nI)^{-1}\, Y + \epsilon, \text{ with } K\epsilon = 0.
\end{aligned}
$$

Example: kernel ridge regression

- However, if $\boldsymbol{\alpha}' = \boldsymbol{\alpha} + \epsilon$ with $K\epsilon = 0$, then:

$$\| \beta - \beta' \|_2^2 = \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}'\right)^\top K \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}'\right) = 0\,,$$

  therefore $\beta = \beta'$.

- One solution to the initial problem is therefore:

$$\hat{f} = \sum_{i=1}^{n} \alpha_i K\left(x_i, x\right)\,,$$

  with

$$\boldsymbol{\alpha} = \left(K + \lambda n I\right)^{-1} Y\,.$$

Comparison with "standard" ridge regression

- Let X the $n \times p$ data matrix, $K = XX^\top$ the kernel Gram matrix.
- In "standard" ridge regression, we have $\hat{f}(x) = \hat{\beta}^\top x$ with

$$\hat{\beta} = \left(X^\top X + n\lambda I\right)^{-1} X^\top Y .$$

- In "kernel" ridge regression, we have $\tilde{f}(x) = \sum_{i=1}^n \alpha_i x_i^\top x = \tilde{\beta}^\top x$ with

$$\tilde{\beta} = \sum_{i=1}^n \alpha_i x_i = X^\top \boldsymbol{\alpha} = X^\top \left(XX^\top + \lambda n I\right)^{-1} Y .$$

- Of course $\hat{\beta} = \tilde{\beta}$! (*left as exercise: use the SVD decomposition of X).*
- Standard RR is better when $p < n$ (big data), kernel RR is better when $n < p$ (high-dimension).

Generalization

- We learn the function $f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)$ by solving in $\boldsymbol{\alpha}$ the following optimization problem, with adequate loss function ℓ:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \ell\left(\sum_{j=1}^{n} \alpha_j K(x_i, x_j), y_i \right) + \lambda \sum_{i,j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) \,.$$

- No explicit solution, but convex optimization problem
- Note that the dimension of the problem is now n instead of p (useful when $n < p$)

The case of SVM

- Soft-margin SVM with a kernel solves:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n, b\in\mathbb{R}} \left\{ \sum_{i=1}^{n} \ell_{\text{hinge}} \left(\sum_{j=1}^{n} \alpha_j K(x_i, x_j), y_i \right) + \lambda \sum_{i,j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) \right\} .$$

- By Lagrange duality we saw that this is equivalent to

$$\max_{\alpha\in\mathbb{R}^n} L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) + b ,$$

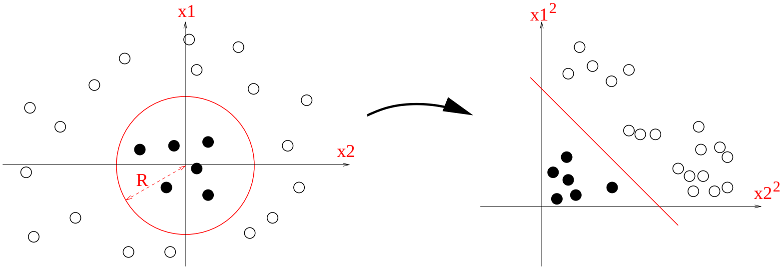
under the constraints:

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

- This is not a surprise, both problems are also dual to each other (*exercise*).

Outline

Remember: polynomial kernel



$$\forall x, x' \in \mathbb{R}^p, \qquad K(x, x') = \left(x^\top x' + 1\right)^d$$

is an inner product in a feature space of all monomials of degree up to d

Which functions $K(x, x')$ are kernels?

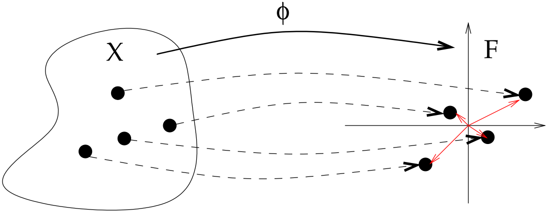
Definition

A function $K(x, x')$ defined on a set $\mathcal{X}$ is a kernel if and only if there exists a features space (Hilbert space) $\mathcal{H}$ and a mapping

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

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

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



Reminder ...

- An inner product on an $\mathbb{R}$-vector space $\mathcal{H}$ is a mapping $(f, g) \mapsto \langle f, g \rangle_{\mathcal{H}}$ from $\mathcal{H}^2$ to $\mathbb{R}$ that is bilinear, symmetric and such that $\langle f, f \rangle > 0$ for all $f \in \mathcal{H} \backslash \{0\}$.
- A vector space endowed with an inner product is called pre-Hilbert. It is endowed with a norm defined by the inner product as $\| f \|_{\mathcal{H}} = \langle f, f \rangle_{\mathcal{H}}^{\frac{1}{2}}$.
- A Hilbert space is a pre-Hilbert space complete for the norm defined by the inner product.

Positive Definite (p.d.) functions

Definition

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

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

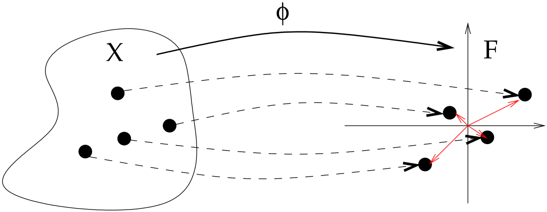
and which satisfies, for all $N \in \mathbb{N}$, $(x_1, x_2, \dots, x_N) \in \mathcal{X}^N$ et $(a_1, a_2, \dots, a_N) \in \mathbb{R}^N$:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j K\left(x_i, x_j\right) \geq 0.$$

Kernels are p.d. functions

Theorem (Aronszajn, 1950)

K is a kernel if and only if it is a positive definite function.



Proof: kernel $\implies$ p.d. (easy)

Let

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

be a kernel. It is p.d. because:

- $K(x, x') = \langle \Phi\left(x\right), \Phi\left(x'\right)\rangle_{\mathcal{H}} = \langle \Phi\left(x'\right), \Phi\left(x\right)\rangle_{\mathcal{H}} = K(x', x)$,
- $\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j \left\langle \Phi\left(x_i\right), \Phi\left(x_j\right)\right\rangle_{\mathcal{H}} = \| \sum_{i=1}^{N} a_i \Phi\left(x_i\right) \|_{\mathcal{H}}^2 \geq 0$.

Proof: p.d. $\implies$ kernel when $\mathcal{X}$ is finite

- Suppose $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ is finite of size N.
- Any p.d. kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is entirely defined by the $N \times N$ symmetric positive semidefinite matrix $[K]_{ij} := K\left(x_i, x_j\right)$.
- It can therefore be diagonalized on an orthonormal basis of eigenvectors $(u_1, u_2, \ldots, u_N)$, with non-negative eigenvalues $0 \leq \lambda_1 \leq \ldots \leq \lambda_N$, i.e.,

$$K\left(x_i, x_j\right) = \left[\sum_{l=1}^{N} \lambda_l u_l u_l^{\top}\right]_{ij} = \sum_{l=1}^{N} \lambda_l u_l(i) u_l(j) = \left\langle \Phi\left(x_i\right), \Phi\left(x_j\right)\right\rangle_{\mathbb{R}^N},$$

with

$$\Phi\left(x_i\right) = \begin{pmatrix} \sqrt{\lambda_1} u_1(i) \\ \vdots \\ \sqrt{\lambda_N} u_N(i) \end{pmatrix}. \qquad \square$$

Proof: p.d. $\implies$ kernel in the general case

- Mercer (1909) for $\mathcal{X} = [a, b] \subset \mathbb{R}$ (more generally $\mathcal{X}$ compact) and K continuous (the so-called Mercer kernels).
- Kolmogorov (1941) for $\mathcal{X}$ countable.
- Aronszajn (1944, 1950) for the general case, using the theory of RKHS.

RKHS

Definition

Let $\mathcal{X}$ be a set and $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ be a class of functions forming a (real) Hilbert space with inner product $\langle .,. \rangle_{\mathcal{H}}$. The function $K : \mathcal{X}^2 \mapsto \mathbb{R}$ is called a reproducing kernel (r.k.) of $\mathcal{H}$ if

1. $\mathcal{H}$ contains all functions of the form
$$\forall x \in \mathcal{X}, \quad K_x : t \mapsto K(x,t) \ .$$
2. For every $x \in \mathcal{X}$ and $f \in \mathcal{H}$ the reproducing property holds:
$$f(x) = \langle f, K_x \rangle_{\mathcal{H}} \ .$$

If a r.k. exists, then $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS).

An equivalent definition of RKHS

Theorem

The Hilbert space $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ is a RKHS if and only if for any $x \in \mathcal{X}$, the mapping:

$$\begin{aligned} F : \quad \mathcal{H} & \to \mathbb{R} \\ f & \mapsto f(x) \end{aligned}$$

is continuous.

Corollary

Convergence in a RKHS implies pointwise convergence, i.e., if $(f_n)_{n \in \mathbb{N}}$ converges to f in $\mathcal{H}$, then $(f_n(x))_{n \in \mathbb{N}}$ converges to $f(x)$ for any $x \in \mathcal{X}$.

An equivalent definition of RKHS

Theorem

The Hilbert space $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ is a RKHS if and only if for any $x \in \mathcal{X}$, the mapping:

$$\begin{aligned} F: \quad \mathcal{H} &\to \mathbb{R} \\ f &\mapsto f(x) \end{aligned}$$

is continuous.

Corollary

Convergence in a RKHS implies pointwise convergence, i.e., if $(f_n)_{n\in\mathbb{N}}$ converges to f in $\mathcal{H}$, then $(f_n(x))_{n\in\mathbb{N}}$ converges to $f(x)$ for any $x \in \mathcal{X}$.

Proof

If $\mathcal{H}$ is a RKHS then $f \mapsto f(x)$ is continuous

If a r.k. K exists, then for any $(x, f) \in \mathcal{X} \times \mathcal{H}$:

$$\begin{aligned} |\, f(x)\,| &= |\,\langle f, K_x\rangle_{\mathcal{H}}\,| \\ &\leq \|\, f\,\|_{\mathcal{H}}.\|\, K_x\,\|_{\mathcal{H}} \text{ (Cauchy-Schwarz)} \\ &\leq \|\, f\,\|_{\mathcal{H}}.K(x,x)^{\frac{1}{2}}\ , \end{aligned}$$

because $\|\, K_x\,\|_{\mathcal{H}}^2 = \langle K_x, K_x\rangle_{\mathcal{H}} = K(x,x)$. Therefore $f \in \mathcal{H} \mapsto f(x) \in \mathbb{R}$ is a continuous linear mapping. $\square$

Proof (Converse)

If $f \mapsto f(x)$ is continuous then $\mathcal{H}$ is a RKHS

Conversely, let us assume that for any $x \in \mathcal{X}$ the linear form $f \in \mathcal{H} \mapsto f(x)$ is continuous.
Then by Riesz representation theorem there (general property of Hilbert spaces) there exists a unique $g_x \in \mathcal{H}$ such that:

$$f(x) = \langle f, g_x \rangle_{\mathcal{H}}$$

The function $K(x, y) = g_x(y)$ is then a r.k. for $\mathcal{H}$. $\square$

Unicity of r.k. and RKHS

Theorem

- If $\mathcal{H}$ is a RKHS, then it has a unique r.k.
- Conversely, a function K can be the r.k. of at most one RKHS.

Consequence

This shows that we can talk of "the" kernel of a RKHS, or "the" RKHS of a kernel.

Unicity of r.k. and RKHS

Theorem

- If $\mathcal{H}$ is a RKHS, then it has a unique r.k.
- Conversely, a function K can be the r.k. of at most one RKHS.

Consequence

This shows that we can talk of "the" kernel of a RKHS, or "the" RKHS of a kernel.

Proof

If a r.k. exists then it is unique

Let K and K' be two r.k. of a RKHS $\mathcal{H}$. Then for any $x \in \mathcal{X}$:

$$\begin{aligned} \| K_x - K'_x \|^2_{\mathcal{H}} &= \langle K_x - K'_x, K_x - K'_x \rangle_{\mathcal{H}} \\ &= \langle K_x - K'_x, K_x \rangle_{\mathcal{H}} - \langle K_x - K'_x, K'_x \rangle_{\mathcal{H}} \\ &= K_x(x) - K'_x(x) - K_x(x) + K'_x(x) \\ &= 0\,. \end{aligned}$$

This shows that $K_x = K'_x$ as functions, i.e., $K_x(y) = K'_x(y)$ for any $y \in \mathcal{X}$. In other words, K=K'. $\square$

The RKHS of a r.k. K is unique

Left as exercice.

Proof

If a r.k. exists then it is unique

Let K and K' be two r.k. of a RKHS $\mathcal{H}$. Then for any $x \in \mathcal{X}$:

$$\begin{aligned}\| K_x - K'_x \|^2_{\mathcal{H}} &= \left\langle K_x - K'_x, K_x - K'_x \right\rangle_{\mathcal{H}} \\ &= \left\langle K_x - K'_x, K_x \right\rangle_{\mathcal{H}} - \left\langle K_x - K'_x, K'_x \right\rangle_{\mathcal{H}} \\ &= K_x\left(x\right) - K'_x\left(x\right) - K_x\left(x\right) + K'_x\left(x\right) \\ &= 0\,.\end{aligned}$$

This shows that $K_x = K'_x$ as functions, i.e., $K_x(y) = K'_x(y)$ for any $y \in \mathcal{X}$. In other words, K=K'. $\square$

The RKHS of a r.k. K is unique

Left as exercice.

An important result

Theorem

A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is p.d. if and only if it is a r.k.

Proof: r.k. $\implies$ p.d.

1. A r.k. is symmetric because, for any $(x, y) \in \mathcal{X}^2$:

$$K(x,y) = \langle K_x, K_y \rangle_{\mathcal{H}} = \langle K_y, K_x \rangle_{\mathcal{H}} = K(y,x)\,.$$

2. It is p.d. because for any $N \in \mathbb{N}$,$(x_1, x_2, \ldots, x_N) \in \mathcal{X}^N$, and $(a_1, a_2, \ldots, a_N) \in \mathbb{R}^N$:

$$\begin{aligned}
\sum_{i,j=1}^{N} a_i a_j K\left(x_i, x_j\right) &= \sum_{i,j=1}^{N} a_i a_j \left\langle K_{x_i}, K_{x_j} \right\rangle_{\mathcal{H}} \\
&= \| \sum_{i=1}^{N} a_i K_{x_i} \|_{\mathcal{H}}^2 \\
&\geq 0. \quad \square
\end{aligned}$$

Proof: p.d. $\Longrightarrow$ r.k. (1/4)

- Let $\mathcal{H}_0$ be the vector subspace of $\mathbb{R}^{\mathcal{X}}$ spanned by the functions $\{K_x\}_{x\in\mathcal{X}}$.
- For any $f, g \in \mathcal{H}_0$, given by:

$$f = \sum_{i=1}^{m} a_i K_{x_i}, \quad g = \sum_{j=1}^{n} b_j K_{y_j},$$

let:

$$\langle f, g\rangle_{\mathcal{H}_0} := \sum_{i,j} a_i b_j K\left(x_i, y_j\right).$$

Proof: p.d. $\implies$ r.k. (2/4)

- $\langle f, g\rangle_{\mathcal{H}_0}$ does not depend on the expansion of f and g because:

$$\langle f, g\rangle_{\mathcal{H}_0} = \sum_{i=1}^{m} a_i g\left(x_i\right) = \sum_{j=1}^{n} b_j f\left(y_j\right) .$$

- This also shows that $\langle ., .\rangle_{\mathcal{H}_0}$ is a symmetric bilinear form.
- This also shows that for any $x \in \mathcal{X}$ and $f \in \mathcal{H}_0$:

$$\langle f, K_x\rangle_{\mathcal{H}_0} = f\left(x\right) .$$

Proof: p.d. $\implies$ r.k. (3/4)

- K is assumed to be p.d., therefore:

$$\| f \|_{\mathcal{H}_0}^2 = \sum_{i,j=1}^{m} a_i a_j K\left(x_i, x_j\right) \geq 0\,.$$

  In particular Cauchy-Schwarz is valid with $\langle ., . \rangle_{\mathcal{H}_0}$.
- By Cauchy-Schwarz we deduce that $\forall x \in \mathcal{X}$:

$$| f(x) | = \left| \langle f, K_x \rangle_{\mathcal{H}_0} \right| \leq \| f \|_{\mathcal{H}_0} . K(x, x)^{\frac{1}{2}} \ ,$$

  therefore $\| f \|_{\mathcal{H}_0} = 0 \implies f = 0$.
- $\mathcal{H}_0$ is therefore a pre-Hilbert space endowed with the inner product $\langle ., . \rangle_{\mathcal{H}_0}$.

Proof: p.d. $\implies$ r.k. (4/4)

- For any Cauchy sequence $(f_n)_{n\geq 0}$ in $\left(\mathcal{H}_0, \langle ., . \rangle_{\mathcal{H}_0}\right)$, we note that:

  $$\forall (x, m, n) \in \mathcal{X} \times \mathbb{N}^2, \quad | f_m(x) - f_n(x) | \leq \| f_m - f_n \|_{\mathcal{H}_0} . K(x, x)^{\frac{1}{2}} \, .$$

  Therefore for any x the sequence $(f_n(x))_{n\geq 0}$ is Cauchy in $\mathbb{R}$ and has therefore a limit.
- If we add to $\mathcal{H}_0$ the functions defined as the pointwise limits of Cauchy sequences, then the space becomes complete and is therefore a Hilbert space, with K as r.k. (up to a few technicalities, left as exercice). $\square$

Application: back to Aronszajn's theorem

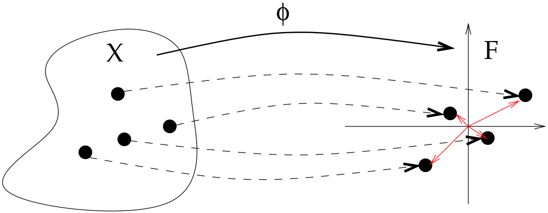
Theorem (Aronszajn, 1950)

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 x, x' in $\mathcal{X}$:

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



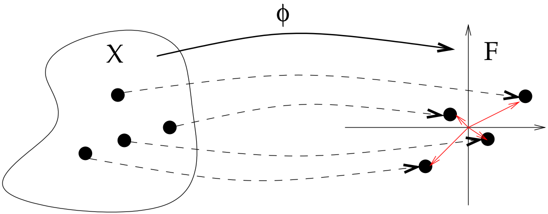
Proof of Aronzsajn's theorem: p.d. $\implies$ kernel

- If K is p.d. over a set $\mathcal{X}$ then it is the r.k. of a Hilbert space $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$.
- Let the mapping $\Phi : \mathcal{X} \to \mathcal{H}$ defined by:

$$\forall x \in \mathcal{X}, \quad \Phi(x) = K_x .$$

- By the reproducing property we have:

$$\forall (x, y) \in \mathcal{X}^2, \quad \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}} = \langle K_x, K_y \rangle_{\mathcal{H}} = K(x, y) . \qquad \square$$



RKHS of the linear kernel

- Let $\mathcal{X} = \mathbb{R}^d$ and $K(x, y) = \langle x, y \rangle_{\mathbb{R}^d}$ be the linear kernel
- The corresponding RKHS consists of functions:

$$x \in \mathbb{R}^d \mapsto f(x) = \sum_i a_i \langle x_i, x \rangle_{\mathbb{R}^d} = \langle w, x \rangle_{\mathbb{R}^d} \ ,$$

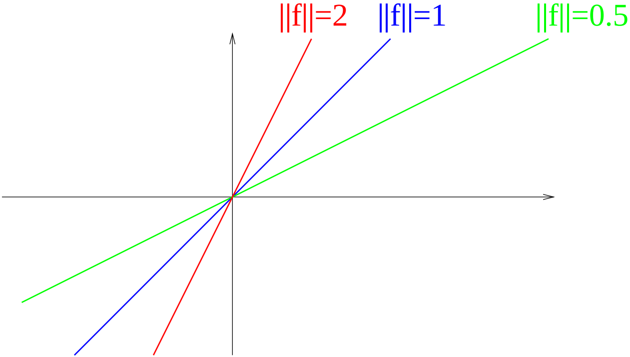
  with $w = \sum_i a_i x_i$.
- The RKHS is therefore the set of linear forms endowed with the following inner product:

$$\langle f, g \rangle_{\mathcal{H}_K} = \langle w, v \rangle_{\mathbb{R}^d} \ ,$$

  when $f(x) = w^\top x$ and $g(x) = v^\top x$.

RKHS of the linear kernel (cont.)

$$\begin{cases} K_{lin}(x, x') & = x^\top x' \, . \\ f(x) & = w^\top x \, , \\ \| f \|_{\mathcal{H}} & = \| w \|_2 \, . \end{cases}$$



ℓ_2-regularized methods in RKHS

$$f_\beta(x) = \beta^\top \Phi(x)\,, \quad \min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(f_\beta(x_i), y_i) + \lambda \|\beta\|_2^2 \right\}$$

is equivalent to

$$\min_{f \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \lambda \|f\|_{\mathcal{H}}^2 \right\}$$

where $\mathcal{H}$ is the RKHS of the kernel $K(x, x') = \Phi(x)^\top \Phi(x')$.

Smoothness functional

A simple inequality

- By Cauchy-Schwarz we have, for any function $f \in \mathcal{H}$ and any two points $x, x' \in \mathcal{X}$:

$$\begin{aligned} \left| f(x) - f\left(x'\right) \right| &= \left| \langle f, K_x - K_{x'} \rangle_{\mathcal{H}} \right| \\ &\leq \| f \|_{\mathcal{H}} \times \| K_x - K_{x'} \|_{\mathcal{H}} \\ &= \| f \|_{\mathcal{H}} \times d_K\left(x, x'\right) . \end{aligned}$$

- The norm of a function in the RKHS controls how fast the function varies over $\mathcal{X}$ with respect to the geometry defined by the kernel (Lipschitz with constant $\| f \|_{\mathcal{H}}$).

Important message

$$\text{Small norm} \implies \text{slow variations}.$$

Kernels and RKHS : Summary

- P.d. kernels can be thought of as inner product after embedding the data space $\mathcal{X}$ in some Hilbert space. As such a p.d. kernel defines a metric on $\mathcal{X}$.
- A realization of this embedding is the RKHS, valid without restriction on the space $\mathcal{X}$ nor on the kernel.
- The RKHS is a space of functions over $\mathcal{X}$. The norm of a function in the RKHS is related to its degree of smoothness w.r.t. the metric defined by the kernel on $\mathcal{X}$.
- ℓ_2-regularized learning in the feature space can be formulated in the RKHS

$$\min_{f \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \lambda \|f\|_{\mathcal{H}}^2 \right\}$$

Outline

Kernel examples

- Polynomial (on $\mathbb{R}^d$):

$$K(x, x') = (x.x' + 1)^d$$

- Gaussian radial basis function (RBF) (on $\mathbb{R}^d$)

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

- Laplace kernel (on $\mathbb{R}$)

$$K(x, x') = \exp\left(-\gamma|x - x'|\right)$$

- Min kernel (on $\mathbb{R}_+$)

$$K(x, x') = \min(x, x')$$

Exercice

Exercice: for each kernel, find a Hilbert space $\mathcal{H}$ and a mapping $\Phi : \mathcal{X} \to \mathcal{H}$ such that $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$

Example: SVM with a Gaussian kernel

- Training:

$$\min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \exp\left(-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma^2} \right)$$

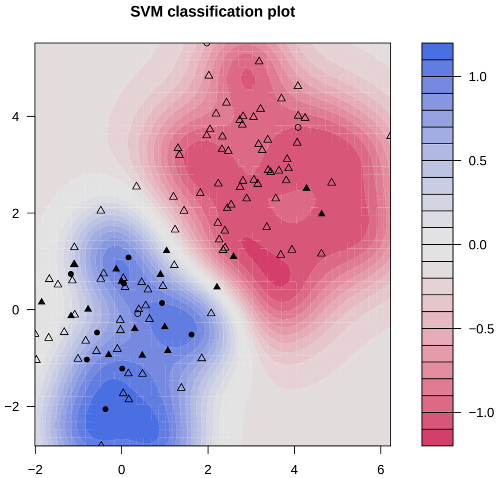
$$\text{s.t. } 0 \leq \alpha_i \leq C, \quad \text{and } \sum_{i=1}^{n} \alpha_i y_i = 0.$$

- Prediction

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

Example: SVM with a Gaussian kernel

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



How to choose or make a kernel?

- Design features
- Design a distance or similarity measure
- Design a regularizer on f

Example: Sobolev norm as regularizer

Theorem

Let $\mathcal{X} = [0, 1]$ and the kernel:

$$\forall (x, y) \in [0, 1]^2, \quad K(x, y) = \min(x, y).$$

Then the RKHS is

$$\mathcal{H} = \left\{ f : [0, 1] \mapsto \mathbb{R}, \text{absolutely continuous}, f' \in L^2([0, 1]), f(0) = 0 \right\}.$$

and the regularizer is a Sobolev norm

$$\Omega(f) = \| f \|_{\mathcal{H}}^2 = \int_0^1 f'(u)^2 \, du = \| f' \|_{L^2([0,1])}^2.$$

Proof (1/5)

Sketch

We need to show that

- $\mathcal{H}$ is a Hilbert space
- $\forall x \in [0,1], K_x \in \mathcal{H}$,
- $\forall (x, f) \in [0,1] \times \mathcal{H}, \langle f, K_x \rangle_{\mathcal{H}} = f(x)$.

Proof (2/5)

$\mathcal{H}$ is a pre-Hilbert space

- f absolutely continuous implies differentiable almost everywhere, and
$$\forall x \in [0,1], \quad f(x) = f(0) + \int_0^x f'(u)du\,.$$
- For any $f \in \mathcal{H}$, $f(0) = 0$ implies by Cauchy-Schwarz:
$$|\, f(x)\,| = \left| \int_0^x f'(u)du \right| \leq \sqrt{x} \left(\int_0^1 f'(u)^2 du \right)^{\frac{1}{2}} = \sqrt{x} \| \, f \, \|_{\mathcal{H}}\,.$$

Therefore, $\| \, f \, \|_{\mathcal{H}} = 0 \implies f = 0$, showing that $\langle ., . \rangle_{\mathcal{H}}$ is an inner product. $\mathcal{H}$ is thus a pre-Hilbert space.

Proof (3/5)

$\mathcal{H}$ is a Hilbert space

- To show that $\mathcal{H}$ is complete, let $(f_n)_{n\in\mathbb{N}}$ a Cauchy sequence in $\mathcal{H}$
- $(f'_n)_{n\in\mathbb{N}}$ is a Cauchy sequence in $L^2[0,1]$, thus converges to $g \in L^2[0,1]$
- By the previous inequality, $(f_n(x))_{n\in\mathbb{N}}$ is a Cauchy sequence and thus converges to a real number $f(x)$, for any $x \in [0,1]$. Moreover:

$$f(x) = \lim_n f_n(x) = \lim_n \int_0^x f'_n(u)du = \int_0^x g(u)du\,,$$

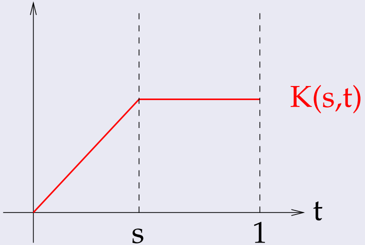
  showing that f is absolutely continuous and $f' = g$ almost everywhere; in particular, $f' \in L^2[0,1]$.
- Finally, $f(0) = \lim_n f_n(0) = 0$, therefore $f \in \mathcal{H}$ and

$$\lim_n \| f_n - f \|_{\mathcal{H}} = \| f' - g_n \|_{L^2[0,1]} = 0\,.$$

Proof (4/5)

$\forall x \in [0,1]$, $K_x \in \mathcal{H}$

Let $K_x(y) = K(x,y) = \min(x,y)$ sur $[0,1]^2$:



K_x is differentiable except at s, has a square integrable derivative, and $K_x(0) = 0$, therefore $K_x \in \mathcal{H}$ for all $x \in [0,1]$. $\square$

Proof (5/5)

For all x, f, $\langle f, K_x \rangle_{\mathcal{H}} = f(x)$

For any $x \in [0,1]$ and $f \in \mathcal{H}$ we have:

$$\langle f, K_x \rangle_{\mathcal{H}} = \int_0^1 f'(u) K_x'(u) du = \int_0^x f'(u) du = f(x),$$

which shows that K is the r.k. associated to $\mathcal{H}$. $\square$

Generalization

Theorem

Let $\mathcal{X} = \mathbb{R}^d$ and D a differential operator on a class of functions $\mathcal{H}$ such that, endowed with the inner product:

$$\forall (f,g) \in \mathcal{H}^2, \quad \langle f, g \rangle_{\mathcal{H}} = \langle Df, Dg \rangle_{L^2(\mathcal{X})},$$

it is a Hilbert space.
Then $\mathcal{H}$ is a RKHS that admits as r.k. the Green function of the operator D^*D, where D^* denotes the adjoint operator of D.

In case of...

Green functions

Let the differential equation on $\mathcal{H}$:

$$f = Dg\,,$$

where g is unknown. In order to solve it we can look for g of the form:

$$g(x) = \int_{\mathcal{X}} k(x, y)\, f(y)\, dy$$

for some function $k : \mathcal{X}^2 \mapsto \mathbb{R}$. k must then satisfy, for all $x \in \mathcal{X}$,

$$f(x) = Dg(x) = \langle Dk_x, f\rangle_{L^2(\mathcal{X})}\,.$$

k is called the Green function of the operator D.

Proof

Let $\mathcal{H}$ be a Hilbert space endowed with the inner product:

$$\langle f, g \rangle_{\mathcal{X}} = \langle Df, Dg \rangle_{L^2(\mathcal{X})} \,,$$

and K be the Green function of the operator D^*D. For all $x \in \mathcal{X}$, $K_x \in \mathcal{H}$ because:

$$\langle DK_x, DK_x \rangle_{L^2(\mathcal{X})} = \langle D^*DK_x, K_x \rangle_{L^2(\mathcal{X})} = K_x(x) < \infty \,.$$

Moreover, for all $f \in \mathcal{H}$ and $x \in \mathcal{X}$, we have:

$$f(x) = \langle D^*DK_x, f \rangle_{L^2(\mathcal{X})} = \langle DK_x, Df \rangle_{L^2(\mathcal{X})} = \langle K_x, f \rangle_{\mathcal{H}} \,,$$

which shows that $\mathcal{H}$ is a RKHS with K as r.k. $\square$

Translation invariant kernels

Definition

A kernel $K : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ is called translation invariant (t.i.) if it only depends on the difference between its argument, i.e.:

$$\forall (x, y) \in \mathbb{R}^{2d}, \quad K(x, y) = \kappa(x - y) \, .$$

Theorem (Bochner)

A real-valued function $\kappa(x - y)$ on $\mathbb{R}^d$ is positive definite if and only if it is the Fourier transform of a symmetric, positive, and finite Borel measure.

RKHS of translation invariant kernels

Theorem

Let K be a translation invariant p.d. kernel, such that κ is integrable on $\mathbb{R}^d$ as well as its Fourier transform $\hat{\kappa}$. The subset $\mathcal{H}_K$ of $L_2\left(\mathbb{R}^d\right)$ that consists of integrable and continuous functions f such that:

$$\| f \|_K^2 := \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \frac{\left| \hat{f}(\omega) \right|^2}{\hat{\kappa}(\omega)} d\omega < +\infty \,,$$

endowed with the inner product:

$$\langle f, g \rangle := \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \frac{\hat{f}(\omega)\hat{g}\,(\omega)^*}{\hat{\kappa}(\omega)} d\omega$$

is a RKHS with K as r.k.

Example: Gaussian RBF kernel

$$K(x,y) = e^{-\frac{(x-y)^2}{2\sigma^2}}$$

corresponds to:

$$\hat{\kappa}(\omega) = e^{-\frac{\sigma^2\omega^2}{2}}$$

and

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

In particular, all functions in $\mathcal{H}$ are infinitely differentiable with all derivatives in L^2.

Example: Laplace kernel

$$K(x,y) = \frac{1}{2} e^{-\gamma |x-y|}$$

corresponds to:

$$\hat{\kappa}(\omega) = \frac{\gamma}{\gamma^2 + \omega^2}$$

and

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

The RKHS is the set of functions L^2 differentiable with derivatives in L^2 (Sobolev space).

Example: sinc kernel

$$K(x,y) = \frac{\sin(\Omega(x-y))}{\pi(x-y)}$$

corresponds to:

$$\hat{\kappa}(\omega) = \mathbf{1}(-\Omega \leq \omega \leq \Omega) \,.$$

The RKHS is the set of functions whose spectrum is included in $[-\Omega, \Omega]$:

$$\mathcal{H} = \left\{ f : \int_{|\,\omega\,| > \Omega} \left| \hat{f}(\omega) \right|^2 d\omega = 0 \right\},$$

and

$$\|\, f \,\|_{\mathcal{H}}^2 = \int_{|\,\omega\,| \leq \Omega} \left| \hat{f}(\omega) \right|^2 = \int_{\omega \in \mathbb{R}} \left| \hat{f}(\omega) \right|^2 = (2\pi)^d \int_{x \in \mathbb{R}} |\, f(x) \,|^2 \, dx \,.$$

Supervised sequence classification

Data (training)

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

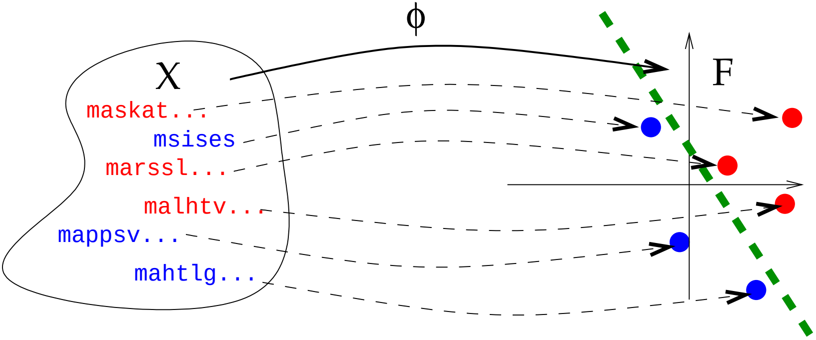
Goal

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

String kernels

The idea

- Map each string $x \in \mathcal{X}$ to a vector $\Phi(x) \in \mathcal{F}$.
- Train a classifier for vectors on the images $\Phi(x_1), \ldots, \Phi(x_n)$ of the training set (nearest neighbor, linear perceptron, logistic regression, support vector machine...)



Substring indexation

The approach

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)

Example: spectrum kernel (1/2)

Kernel definition

- The 3-spectrum of

$$\mathbf{x} = \mathtt{CGGSLIAMMWFGV}$$

  is:

$$\mathtt{(CGG,GGS,GSL,SLI,LIA,IAM,AMM,MMW,MWF,WFG,FGV)} \,.$$

- Let $\Phi_u(\mathbf{x})$ denote the number of occurrences of u in $\mathbf{x}$. The k-spectrum kernel is:

$$K(\mathbf{x}, \mathbf{x}') := \sum_{u \in \mathcal{A}^k} \Phi_u(\mathbf{x}) \Phi_u(\mathbf{x}') \,.$$

Example: spectrum kernel (2/2)

Implementation

- The computation of the kernel is formally a sum over $|\mathcal{A}|^k$ terms, but at most $|\mathbf{x}| - k + 1$ terms are non-zero in $\Phi(\mathbf{x}) \implies$ Computation in $O(|\mathbf{x}| + |\mathbf{x}'|)$ with pre-indexation of the strings.
- Fast classification of a sequence $\mathbf{x}$ in $O(|\mathbf{x}|)$:

$$f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) = \sum_u w_u \Phi_u(\mathbf{x}) = \sum_{i=1}^{|\mathbf{x}|-k+1} w_{x_i \dots x_{i+k-1}}.$$

Remarks

- Work with any string (natural language, time series...)
- Fast and scalable, a good default method for string classification.
- Variants allow matching of k-mers up to m mismatches.

Local alignmnent kernel (Saigo et al., 2004)

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

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

$$
SW_{S,g}(x,y) := \max_{\pi \in \Pi(x,y)} s_{S,g}(\pi) \quad \text{is not a kernel}
$$

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

LA kernel is p.d.: proof (1/2)

Definition: Convolution kernel (Haussler, 1999)

Let K_1 and K_2 be two p.d. kernels for strings. The convolution of K_1 and K_2, denoted $K_1 \star K_2$, is defined for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ by:

$$K_1 \star K_2(\mathbf{x}, \mathbf{y}) := \sum_{\mathbf{x}_1\mathbf{x}_2 = \mathbf{x}, \mathbf{y}_1\mathbf{y}_2 = \mathbf{y}} K_1(\mathbf{x}_1, \mathbf{y}_1) K_2(\mathbf{x}_2, \mathbf{y}_2).$$

Lemma

If K_1 and K_2 are p.d. then $K_1 \star K_2$ is p.d..

LA kernel is p.d.: proof (2/2)

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

with

- The constant kernel:

$$K_0 \left(\mathbf{x}, \mathbf{y} \right) := 1 \,.$$

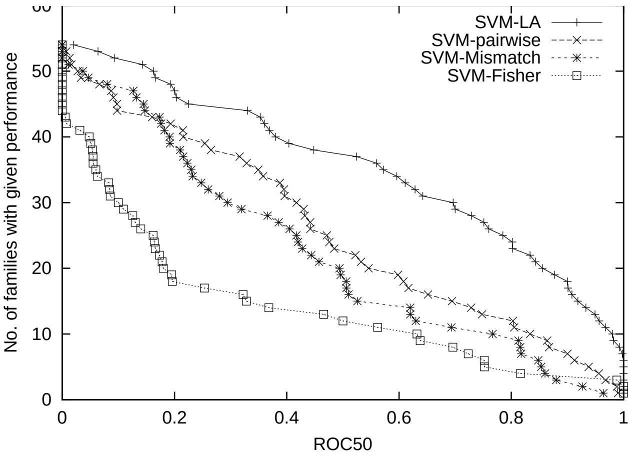
- A kernel for letters:

$$K_a^{(\beta)} \left(\mathbf{x}, \mathbf{y} \right) := \begin{cases} 0 & \text{if } \mid \mathbf{x} \mid \neq 1 \text{ where } \mid \mathbf{y} \mid \neq 1 \,, \\ \exp\left(\beta S(\mathbf{x}, \mathbf{y}) \right) & \text{otherwise} \,. \end{cases}$$

- A kernel for gaps:

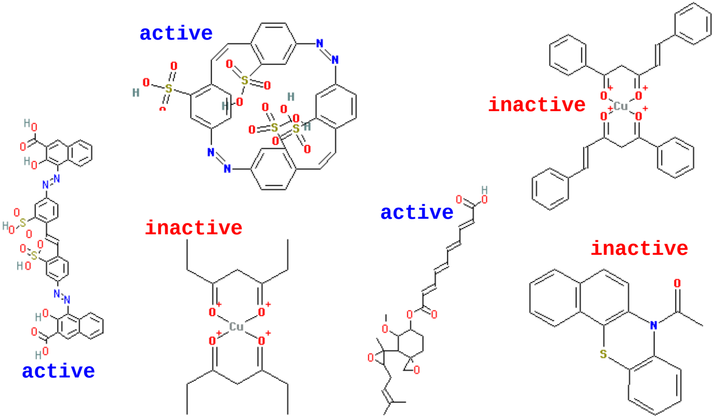
$$K_g^{(\beta)} \left(\mathbf{x}, \mathbf{y} \right) = \exp\left[\beta \left(g\left(\mid \mathbf{x} \mid \right) + g\left(\mid \mathbf{x} \mid \right) \right) \right] \,.$$

The choice of kernel matters



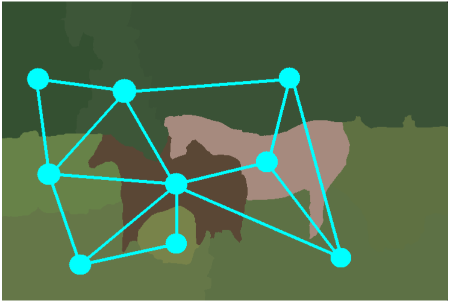
Performance on the SCOP superfamily recognition benchmark (from Saigo et al., 2004).

Virtual screening for drug discovery



NCI AIDS screen results (from http://cactus.nci.nih.gov).

Image retrieval and classification



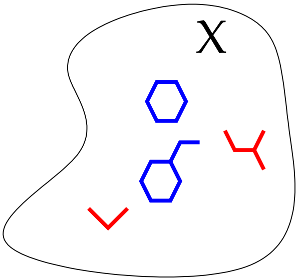
From Harchaoui and Bach (2007).

Graph kernels

1. Represent each graph x by a vector $\Phi(x) \in \mathcal{H}$, either explicitly or implicitly through the kernel

$$K(x, x') = \Phi(x)^\top \Phi(x')\,.$$

2. Use a linear method for classification in $\mathcal{H}$.

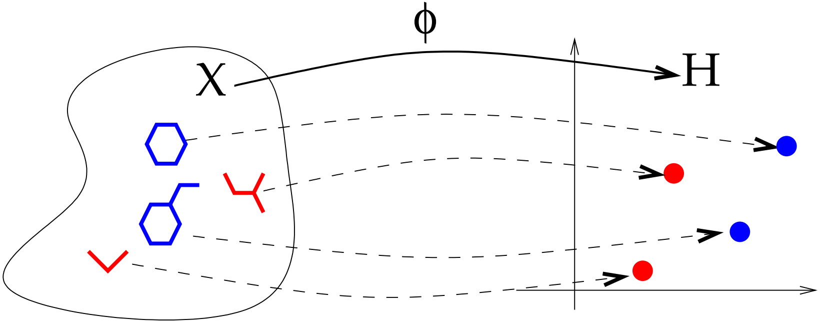


Graph kernels

1. Represent each graph x by a vector $\Phi(x) \in \mathcal{H}$, either explicitly or implicitly through the kernel

$$K(x, x') = \Phi(x)^\top \Phi(x') .$$

2. Use a linear method for classification in $\mathcal{H}$.

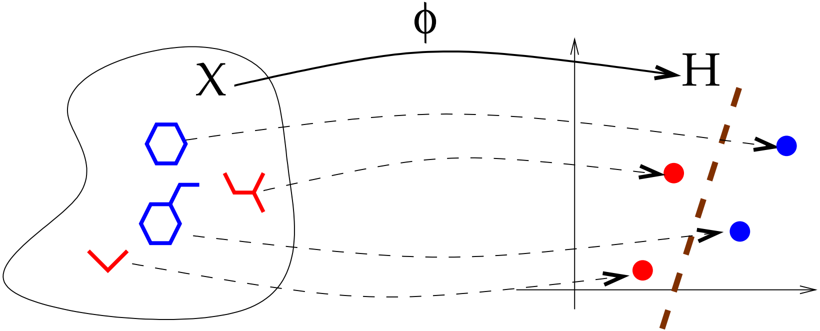


Graph kernels

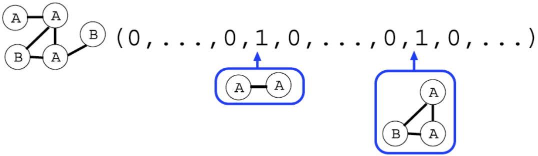
1. Represent each graph x by a vector $\Phi(x) \in \mathcal{H}$, either explicitly or implicitly through the kernel

$$K(x, x') = \Phi(x)^\top \Phi(x').$$

2. Use a linear method for classification in $\mathcal{H}$.



Indexing by all subgraphs?



Theorem

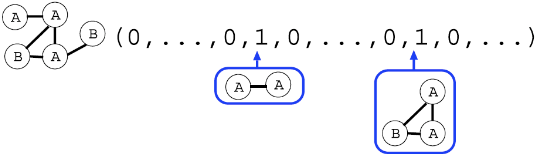
Computing all subgraph occurrences is NP-hard.

Proof.

- The linear graph of size n is a subgraph of a graph X with n vertices iff X has an Hamiltonian path
- The decision problem whether a graph has a Hamiltonian path is NP-complete.

$\square$

Indexing by all subgraphs?



Theorem

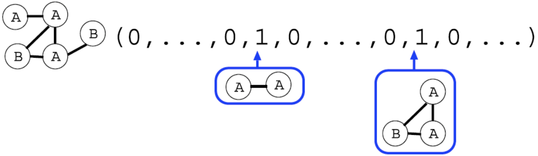
Computing all subgraph occurrences is NP-hard.

Proof.

- The linear graph of size n is a subgraph of a graph X with n vertices iff X has an Hamiltonian path
- The decision problem whether a graph has a Hamiltonian path is NP-complete.

□

Indexing by all subgraphs?



Theorem

Computing all subgraph occurrences is NP-hard.

Proof.

- The linear graph of size n is a subgraph of a graph X with n vertices iff X has an Hamiltonian path
- The decision problem whether a graph has a Hamiltonian path is NP-complete.

□

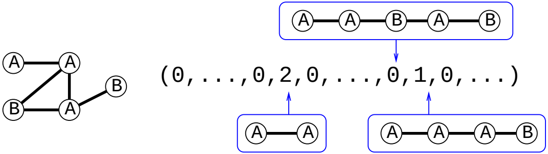
Indexing by specific subgraphs

Substructure selection

We can imagine more limited sets of substuctures that lead to more computationnally efficient indexing (non-exhaustive list)

- substructures selected by domain knowledge (MDL fingerprint)
- all path up to length k (Openeye fingerprint, Nicholls 2005)
- all shortest paths (Borgwardt and Kriegel, 2005)
- all subgraphs up to k vertices (graphlet kernel, Sherashidze et al., 2009)
- all frequent subgraphs in the database (Helma et al., 2004)

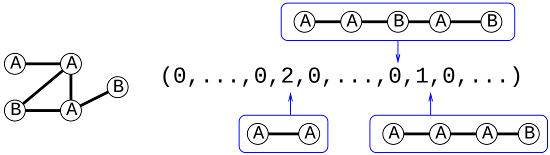
Example : Indexing by all shortest paths



Properties (Borgwardt and Kriegel, 2005)

- There are $O(n^2)$ shortest paths.
- The vector of counts can be computed in $O(n^4)$ with the Floyd-Warshall algorithm.

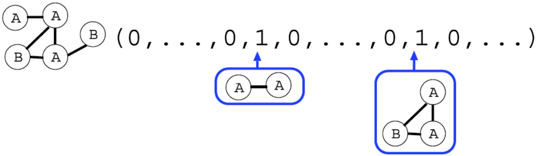
Example : Indexing by all shortest paths



Properties (Borgwardt and Kriegel, 2005)

- There are $O(n^2)$ shortest paths.
- The vector of counts can be computed in $O(n^4)$ with the Floyd-Warshall algorithm.

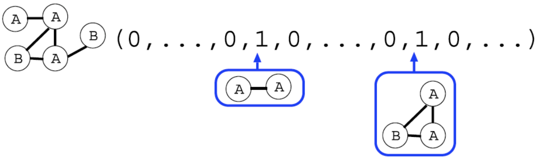
Example : Indexing by all subgraphs up to k vertices



Properties (Shervashidze et al., 2009)

- Naive enumeration scales as $O(n^k)$.
- Enumeration of connected graphlets in $O(nd^{k-1})$ for graphs with degree $\leq d$ and $k \leq 5$.
- Randomly sample subgraphs if enumeration is infeasible.

Example : Indexing by all subgraphs up to k vertices



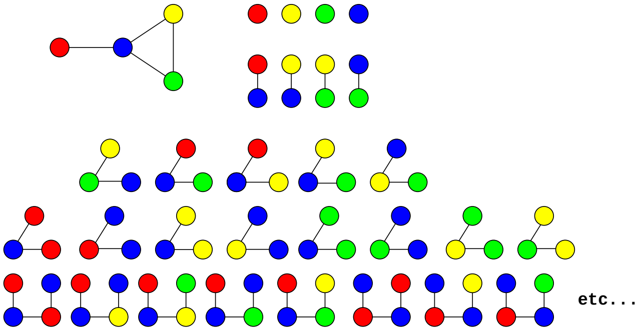
Properties (Shervashidze et al., 2009)

- Naive enumeration scales as $O(n^k)$.
- Enumeration of connected graphlets in $O(nd^{k-1})$ for graphs with degree $\leq d$ and $k \leq 5$.
- Randomly sample subgraphs if enumeration is infeasible.

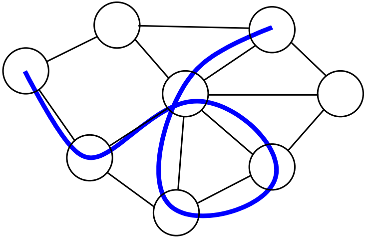
Walks

Definition

- A walk of a graph (V, E) is sequence of $v_1, \ldots, v_n \in V$ such that $(v_i, v_{i+1}) \in E$ for $i = 1, \ldots, n-1$.
- We note $\mathcal{W}_n(G)$ the set of walks with n vertices of the graph G, and $\mathcal{W}(G)$ the set of all walks.



Walks $\neq$ paths



Walk kernel

Definition

- Let $\mathcal{S}_n$ denote the set of all possible label sequences of walks of length n (including vertices and edges labels), and $\mathcal{S} = \cup_{n\geq 1}\mathcal{S}_n$.
- For any graph $\mathcal{X}$ let a weight $\lambda_G(w)$ be associated to each walk $w \in \mathcal{W}(G)$.
- Let the feature vector $\Phi(G) = (\Phi_s(G))_{s\in\mathcal{S}}$ be defined by:

$$\Phi_s(G) = \sum_{w\in\mathcal{W}(G)} \lambda_G(w)\mathbf{1}\,(s \text{ is the label sequence of } w)\ .$$

- A walk kernel is a graph kernel defined by:

$$K_{walk}(G_1, G_2) = \sum_{s\in\mathcal{S}} \Phi_s(G_1)\Phi_s(G_2)\ .$$

Walk kernel

Definition

- Let $\mathcal{S}_n$ denote the set of all possible label sequences of walks of length n (including vertices and edges labels), and $\mathcal{S} = \cup_{n\geq 1}\mathcal{S}_n$.
- For any graph $\mathcal{X}$ let a weight $\lambda_G(w)$ be associated to each walk $w \in \mathcal{W}(G)$.
- Let the feature vector $\Phi(G) = (\Phi_s(G))_{s\in\mathcal{S}}$ be defined by:

$$\Phi_s(G) = \sum_{w\in\mathcal{W}(G)} \lambda_G(w)\mathbf{1}\left(s \text{ is the label sequence of } w\right).$$

- A walk kernel is a graph kernel defined by:

$$K_{walk}(G_1, G_2) = \sum_{s\in\mathcal{S}} \Phi_s(G_1)\Phi_s(G_2).$$

Walk kernel examples

- The *n*th-order walk kernel is the walk kernel with $\lambda_G(w) = 1$ if the length of w is n, 0 otherwise. It compares two graphs through their common walks of length n.
- The random walk kernel is obtained with $\lambda_G(w) = P_G(w)$, where P_G is a Markov random walk on G. In that case we have:

$$K(G_1, G_2) = P(label(W_1) = label(W_2)),$$

where W_1 and W_2 are two independant random walks on G_1 and G_2, respectively (Kashima et al., 2003).
- The geometric walk kernel is obtained (when it converges) with $\lambda_G(w) = \beta^{length(w)}$, for $\beta > 0$. In that case the feature space is of infinite dimension (Gärtner et al., 2003).

Walk kernel examples

- The *n*th-order walk kernel is the walk kernel with $\lambda_G(w) = 1$ if the length of w is n, 0 otherwise. It compares two graphs through their common walks of length n.
- The random walk kernel is obtained with $\lambda_G(w) = P_G(w)$, where P_G is a Markov random walk on G. In that case we have:

$$K(G_1, G_2) = P(label(W_1) = label(W_2)),$$

  where W_1 and W_2 are two independant random walks on G_1 and G_2, respectively (Kashima et al., 2003).
- The geometric walk kernel is obtained (when it converges) with $\lambda_G(w) = \beta^{length(w)}$, for $\beta > 0$. In that case the feature space is of infinite dimension (Gärtner et al., 2003).

Walk kernel examples

- The nth-order walk kernel is the walk kernel with $\lambda_G(w) = 1$ if the length of w is n, 0 otherwise. It compares two graphs through their common walks of length n.
- The random walk kernel is obtained with $\lambda_G(w) = P_G(w)$, where P_G is a Markov random walk on G. In that case we have:

$$K(G_1, G_2) = P(label(W_1) = label(W_2)) \,,$$

  where W_1 and W_2 are two independant random walks on G_1 and G_2, respectively (Kashima et al., 2003).
- The geometric walk kernel is obtained (when it converges) with $\lambda_G(w) = \beta^{length(w)}$, for $\beta > 0$. In that case the feature space is of infinite dimension (Gärtner et al., 2003).

Computation of walk kernels

Proposition

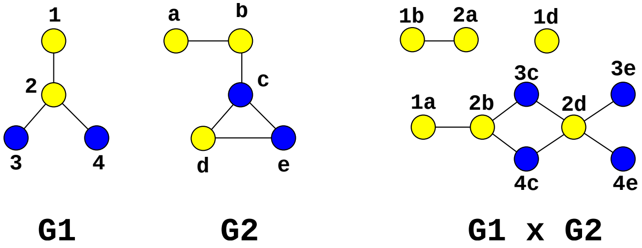
These three kernels (nth-order, random and geometric walk kernels) can be computed efficiently in polynomial time.

Product graph

Definition

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs with labeled vertices. The product graph $G = G_1 \times G_2$ is the graph $G = (V, E)$ with:

1. $V = \{(v_1, v_2) \in V_1 \times V_2 \ : \ v_1 \text{ and } v_2 \text{ have the same label}\}$,
2. $E = \left\{((v_1, v_2), (v_1', v_2')) \in V \times V \ : \ (v_1, v_1') \in E_1 \text{ and } (v_2, v_2') \in E_2\right\}$.



Walk kernel and product graph

Lemma

There is a bijection between:

1. The pairs of walks $w_1 \in \mathcal{W}_n(G_1)$ and $w_2 \in \mathcal{W}_n(G_2)$ with the same label sequences,
2. The walks on the product graph $w \in \mathcal{W}_n(G_1 \times G_2)$.

Corollary

$$
\begin{aligned}
K_{walk}(G_1, G_2) &= \sum_{s \in \mathcal{S}} \Phi_s(G_1)\Phi_s(G_2) \\
&= \sum_{(w_1,w_2) \in \mathcal{W}(G_1) \times \mathcal{W}(G_1)} \lambda_{G_1}(w_1)\lambda_{G_2}(w_2)\mathbf{1}(l(w_1) = l(w_2)) \\
&= \sum_{w \in \mathcal{W}(G_1 \times G_2)} \lambda_{G_1 \times G_2}(w) .
\end{aligned}
$$

Walk kernel and product graph

Lemma

There is a bijection between:

1. The pairs of walks $w_1 \in \mathcal{W}_n(G_1)$ and $w_2 \in \mathcal{W}_n(G_2)$ with the same label sequences,
2. The walks on the product graph $w \in \mathcal{W}_n(G_1 \times G_2)$.

Corollary

$$
\begin{aligned}
K_{walk}(G_1, G_2) &= \sum_{s \in \mathcal{S}} \Phi_s(G_1)\Phi_s(G_2) \\
&= \sum_{(w_1,w_2) \in \mathcal{W}(G_1) \times \mathcal{W}(G_1)} \lambda_{G_1}(w_1)\lambda_{G_2}(w_2)\mathbf{1}(l(w_1) = l(w_2)) \\
&= \sum_{w \in \mathcal{W}(G_1 \times G_2)} \lambda_{G_1 \times G_2}(w) \,.
\end{aligned}
$$

Computation of the nth-order walk kernel

- For the nth-order walk kernel we have $\lambda_{G_1 \times G_2}(w) = 1$ if the length of w is n, 0 otherwise.
- Therefore:

$$K_{nth-order}\left(G_1, G_2\right) = \sum_{w \in \mathcal{W}_n(G_1 \times G_2)} 1 \,.$$

- Let A be the adjacency matrix of $G_1 \times G_2$. Then we get:

$$K_{nth-order}\left(G_1, G_2\right) = \sum_{i,j} [A^n]_{i,j} = \mathbf{1}^\top A^n \mathbf{1} \,.$$

- Computation in $O(n|G_1||G_2|d_1 d_2)$, where d_i is the maximum degree of G_i.

Computation of random and geometric walk kernels

- In both cases $\lambda_G(w)$ for a walk $w = v_1 \dots v_n$ can be decomposed as:

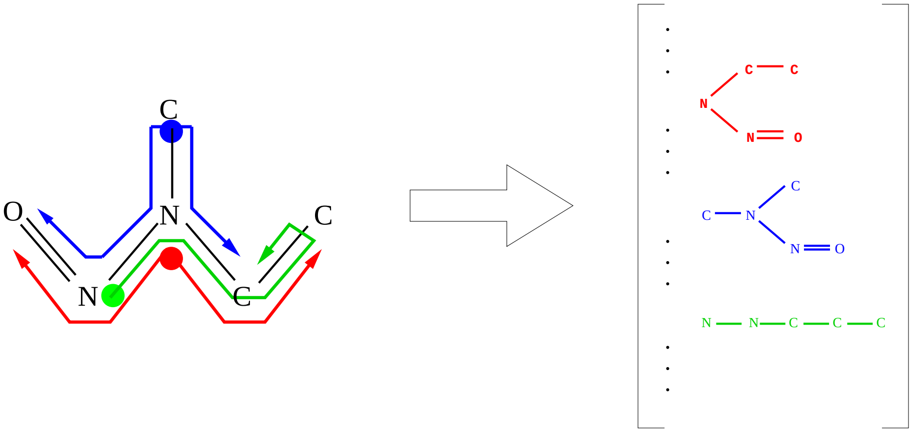
$$\lambda_G(v_1 \dots v_n) = \lambda^i(v_1) \prod_{i=2}^{n} \lambda^t(v_{i-1}, v_i) \,.$$

- Let Λ_i be the vector of $\lambda^i(v)$ and Λ_t be the matrix of $\lambda^t(v, v')$:

$$\begin{aligned} K_{walk}(G_1, G_2) &= \sum_{n=1}^{\infty} \sum_{w \in \mathcal{W}_n(G_1 \times G_2)} \lambda^i(v_1) \prod_{i=2}^{n} \lambda^t(v_{i-1}, v_i) \\ &= \sum_{n=0}^{\infty} \Lambda_i \Lambda_t^n \mathbf{1} \\ &= \Lambda_i \left(I - \Lambda_t\right)^{-1} \mathbf{1} \end{aligned}$$

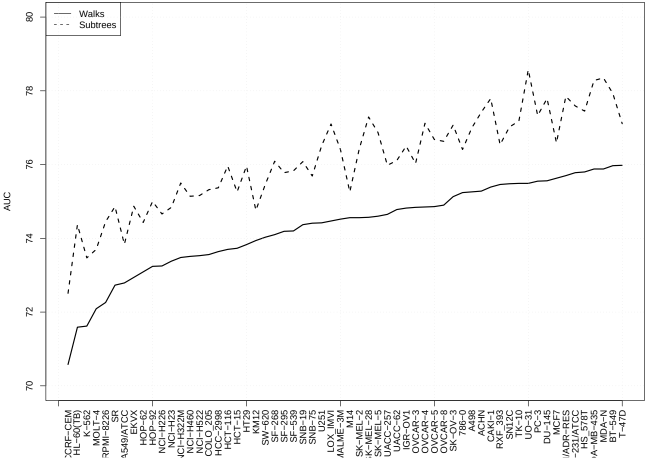
- Computation in $O(|G_1|^3 |G_2|^3)$

Extension: branching walks (Ramon and Gärtner, 2003; Mahé and Vert, 2009)



$$\mathcal{T}(v, n+1) = \sum_{R \subset \mathcal{N}(v)} \prod_{v' \in R} \lambda_t(v, v')\mathcal{T}(v', n)\,,$$

2D Subtree vs walk kernels

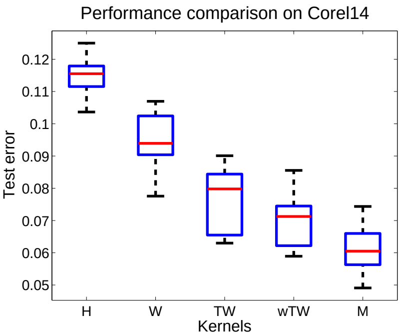


Screening of inhibitors for 60 cancer cell lines.

Image classification (Harchaoui and Bach, 2007)

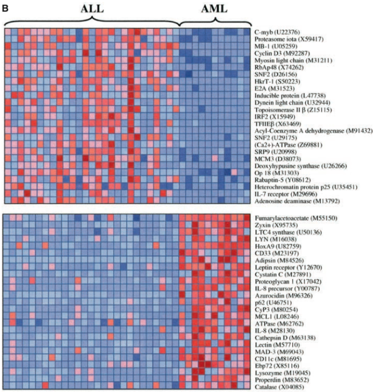
COREL14 dataset

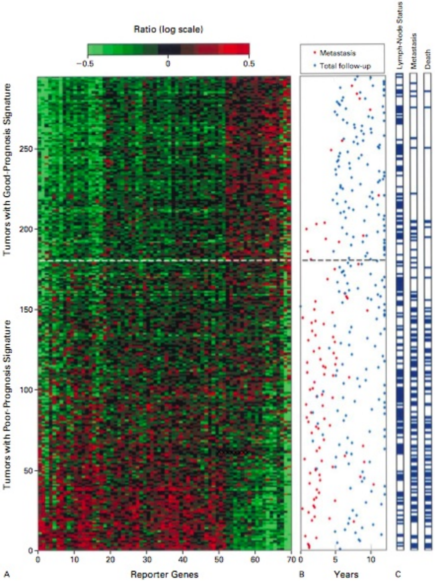
- 1400 natural images in 14 classes
- Compare kernel between histograms (H), walk kernel (W), subtree kernel (TW), weighted subtree kernel (wTW), and a combination (M).



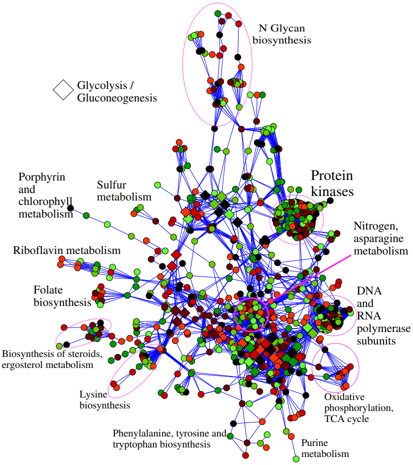
Outline

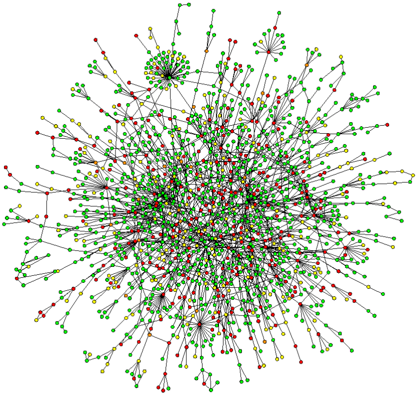
Molecular diagnosis / prognosis / theragnosis





Gene networks

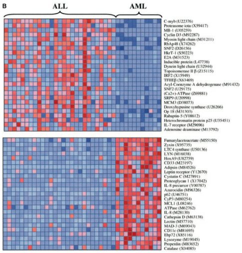


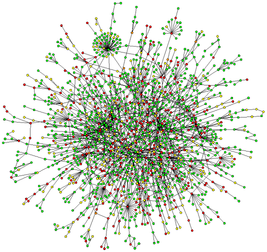


Gene networks and expression data

Motivation

- Basic biological functions usually involve the coordinated action of several proteins:
  - Formation of protein complexes
  - Activation of metabolic, signalling or regulatory pathways
- Many pathways and protein-protein interactions are already known
- Hypothesis: the weights of the classifier should be “coherent” with respect to this prior knowledge





Graph based penalty

$$f_\beta(x) = \beta^\top x \qquad \min_\beta R(f_\beta) + \lambda \Omega(\beta)$$

Prior hypothesis

Genes near each other on the graph should have similar weigths.

An idea (Rapaport et al., 2007)

$$\Omega(\beta) = \sum_{i \sim j} (\beta_i - \beta_j)^2 ,$$

$$\min_{\beta \in \mathbb{R}^p} R(f_\beta) + \lambda \sum_{i \sim j} (\beta_i - \beta_j)^2 .$$

Graph based penalty

$$f_\beta(x) = \beta^\top x \qquad \min_\beta R(f_\beta) + \lambda\Omega(\beta)$$

Prior hypothesis

Genes near each other on the graph should have similar weigths.

An idea (Rapaport et al., 2007)

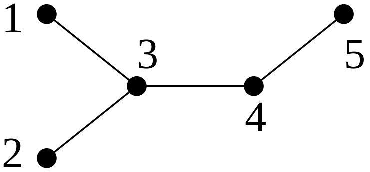
$$\Omega(\beta) = \sum_{i \sim j} (\beta_i - \beta_j)^2 ,$$

$$\min_{\beta \in \mathbb{R}^p} R(f_\beta) + \lambda \sum_{i \sim j} (\beta_i - \beta_j)^2 .$$

Graph Laplacian

Definition

The Laplacian of the graph is the matrix $L = D - A$.



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

Spectral penalty as a kernel

Theorem

The function $f(x) = \beta^\top x$ where β is solution of

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \ell\left(\beta^\top x_i, y_i\right) + \lambda \sum_{i \sim j} \left(\beta_i - \beta_j\right)^2$$

is equal to $g(x) = \gamma^\top \Phi(x)$ where γ is solution of

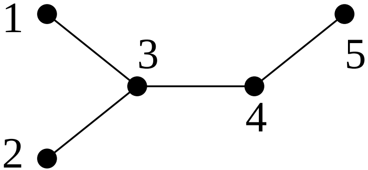
$$\min_{\gamma \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \ell\left(\gamma^\top \Phi(x_i), y_i\right) + \lambda \gamma^\top \gamma \,,$$

and where

$$\Phi(x)^\top \Phi(x') = x^\top K_G x'$$

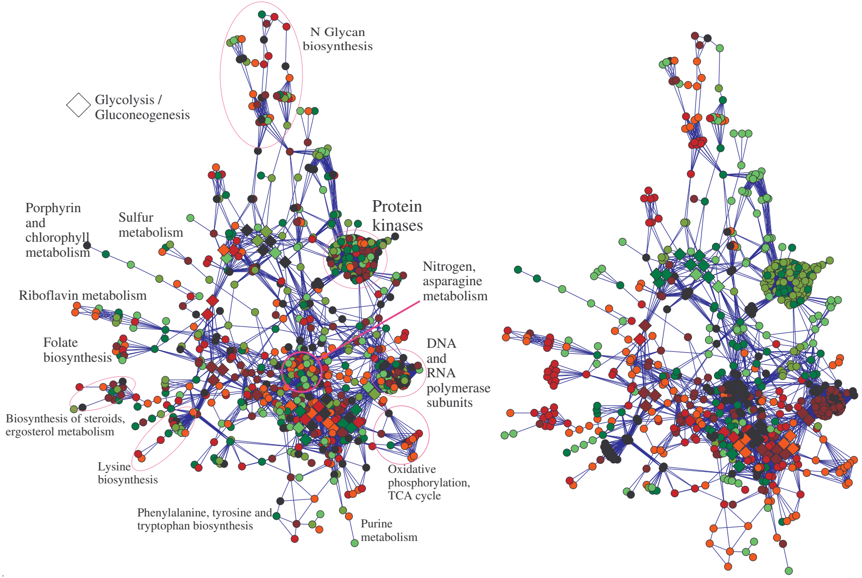
for $K_G = L^*$, the pseudo-inverse of the graph Laplacian.

Example

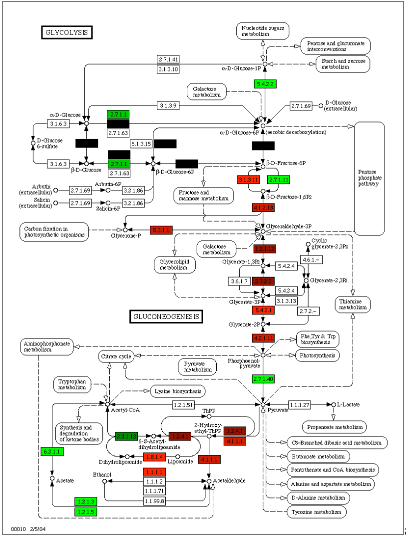


$$L^* = \begin{pmatrix} 0.88 & -0.12 & 0.08 & -0.32 & -0.52 \\ -0.12 & 0.88 & 0.08 & -0.32 & -0.52 \\ 0.08 & 0.08 & 0.28 & -0.12 & -0.32 \\ -0.32 & -0.32 & -0.12 & 0.48 & 0.28 \\ -0.52 & -0.52 & -0.32 & 0.28 & 1.08 \end{pmatrix}$$

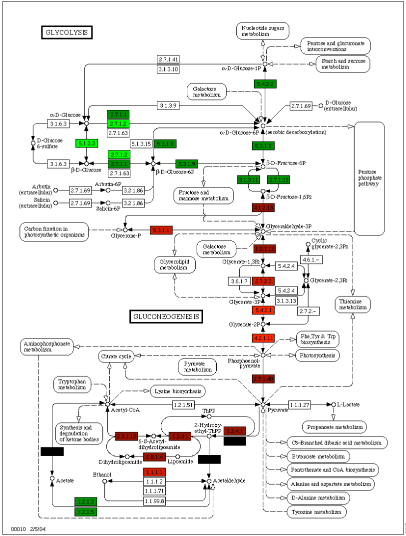
Classifiers



Classifier



a)



b)

Other penalties with kernels

$$\Phi(x)^\top \Phi(x') = x^\top K_G x'$$

with:

- $K_G = (c + L)^-1$ leads to

$$\Omega(\beta) = c \sum_{i=1}^{p} \beta_i^2 + \sum_{i \sim j} \left(\beta_i - \beta_j\right)^2 .$$

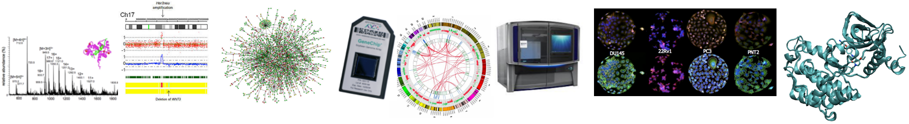
- The diffusion kernel:

$$K_G = \exp_M(-2tL) .$$

penalizes high frequencies of β in the Fourier domain.

Outline

Motivation



- Assume we observe K types of data and would like to learn a joint model (e.g., predict susceptibility from SNP and expression data).
- We saw in the previous part how to make kernels for each type of data, and learn with kernels
- Kernels are also well suited for data integration!

Setting

- For a kernel K with RKHS $\mathcal{H}$, we learn a function $f \in \mathcal{H}$ by solving:

$$\min_{f \in \mathcal{H}} R(f^n) + \lambda \|f\|_{\mathcal{H}}^2 ,$$

where $f^n = (f(x_1), \ldots, f(x_n)) \in \mathbb{R}^n$

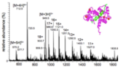
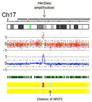
- By the representer theorem, we know that the solution is

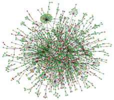
$$f(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i) ,$$

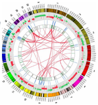
where $\boldsymbol{\alpha} \in \mathbb{R}^n$ is the solution of another optimization problem:

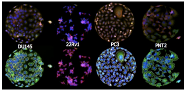
$$\min_{\boldsymbol{\alpha}} R(K\boldsymbol{\alpha}) + \lambda \boldsymbol{\alpha}^\top K \boldsymbol{\alpha} = \min_{\boldsymbol{\alpha}} J_K(\boldsymbol{\alpha}) .$$

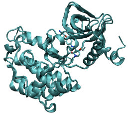
Sum kernel







Definition

Let $K_1, \ldots, K_M$ be M kernels on $\mathcal{X}$. The sum kernel K_S is the kernel on $\mathcal{X}$ defined as

$$\forall x, x' \in \mathcal{X}, \quad K_S(x, x') = \sum_{i=1}^{M} K_i(x, x') \,.$$

Sum kernel and vector concatenation

Theorem

For $i = 1, \ldots, M$, let $\Phi_i : \mathcal{X} \to \mathcal{H}_i$ be a feature map such that

$$K_i(x, x') = \left\langle \Phi_i\left(x\right), \Phi_i\left(x'\right) \right\rangle_{\mathcal{H}_i} .$$

Then $K_S = \sum_{i=1}^{M} K_i$ can be written as:

$$K_S(x, x') = \left\langle \Phi_S\left(x\right), \Phi_S\left(x'\right) \right\rangle_{\mathcal{H}_S} ,$$

where $\Phi_S : \mathcal{X} \to \mathcal{H}_S = \mathcal{H}_1 \oplus \ldots \oplus \mathcal{H}_M$ is the concatenation of the feature maps Φ_i:

$$\Phi_S\left(x\right) = \left(\Phi_1\left(x\right), \ldots, \Phi_M\left(x\right)\right)^\top .$$

Therefore, summing kernels amounts to concatenating their feature space representations, which is a quite natural way to integrate different features.

Proof

For $\Phi_S(x) = (\Phi_1(x), \ldots, \Phi_M(x))^\top$, we easily compute:

$$\begin{aligned}\left\langle \Phi_S(x), \Phi_S(x') \right\rangle_{\mathcal{H}_s} &= \sum_{i=1}^{M} \left\langle \Phi_i(x), \Phi_i(x') \right\rangle_{\mathcal{H}_i} \\ &= \sum_{i=1}^{M} K_i(x, x') \\ &= K_S(x, x') .\end{aligned}$$

Example: data integration with the sum kernel

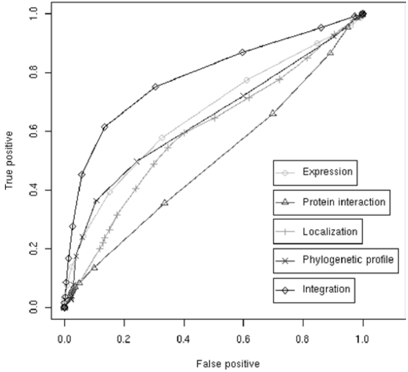
BIOINFORMATICS

Vol. 20 Suppl. 1 2004, pages i363–i370
DOI: 10.1093/bioinformatics/bth910

Protein network inference from multiple genomic data: a supervised approach

Y. Yamanishi[1,], J.-P. Vert[2] and M. Kanehisa[1]*

[1]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto 611-0011, Japan and [2]Computational Biology group, Ecole des Mines de Paris, 35 rue Saint-Honoré, 77305 Fontainebleau cedex, France



K_{exp} (Expression)
K_{ppi} (Protein interaction)
K_{loc} (Localization)
K_{phy} (Phylogenetic profile)
$K_{\text{exp}} + K_{\text{ppi}} + K_{\text{loc}} + K_{\text{phy}}$ (Integration)

The sum kernel: functional point of view

Theorem

The solution $f^* \in \mathcal{H}_{K_S}$ when we learn with $K_S = \sum_{i=1}^M K_i$ is equal to:

$$f^* = \sum_{i=1}^M f_i^* \,,$$

where $(f_1^*, \ldots, f_M^*) \in \mathcal{H}_{K_1} \times \ldots \times \mathcal{H}_{K_M}$ is the solution of:

$$\min_{f_1,\ldots,f_M} R\left(\sum_{i=1}^M f_i^n\right) + \lambda \sum_{i=1}^M \| f_i \|_{\mathcal{H}_i}^2 \,.$$

Generalization: The weighted sum kernel

Theorem

The solution f^* when we learn with $K_\eta = \sum_{i=1}^M \eta_i K_i$, with $\eta_1, \ldots, \eta_M \geq 0$, is equal to:

$$f^* = \sum_{i=1}^M f_i^* \,,$$

where $(f_1^*, \ldots, f_M^*) \in \mathcal{H}_{K_1} \times \ldots \times \mathcal{H}_{K_M}$ is the solution of:

$$\min_{f_1,\ldots,f_M} R\left(\sum_{i=1}^M f_i^n\right) + \lambda \sum_{i=1}^M \frac{\| f_i \|_{\mathcal{H}_i}^2}{\eta_i} \,.$$

Proof (1/4)

$$\min_{f_1,\ldots,f_M} R\left(\sum_{i=1}^{M} f_i^n\right) + \lambda \sum_{i=1}^{M} \frac{\| f_i \|_{\mathcal{H}_i}^2}{\eta_i}\,.$$

- R being convex, the problem is strictly convex and has a unique solution $(f_1^*, \ldots, f_M^*) \in \mathcal{H}_{K_1} \times \ldots \times \mathcal{H}_{K_M}$.
- By the representer theorem, there exists $\alpha_1^*, \ldots, \alpha_M^* \in \mathbb{R}^n$ such that

$$f_i^*(x) = \sum_{j=1}^{n} \alpha_{ij}^* K_i(x_j, x)\,.$$

- $(\alpha_1^*, \ldots, \alpha_M^*)$ is the solution of

$$\min_{\alpha_1,\ldots,\alpha_M \in \mathbb{R}^n} R\left(\sum_{i=1}^{M} K_i \alpha_i\right) + \lambda \sum_{i=1}^{M} \frac{\alpha_i^\top K_i \alpha_i}{\eta_i}\,.$$

Proof (2/4)

- This is equivalent to

$$\min_{u,\alpha_1,\ldots,\alpha_M \in \mathbb{R}^n} R(u) + \lambda \sum_{i=1}^{M} \frac{\alpha_i^\top K_i \alpha_i}{\eta_i} \quad \text{s.t.} \quad u = \sum_{i=1}^{M} K_i \alpha_i \,.$$

- This is equivalent to the saddle point problem:

$$\min_{u,\alpha_1,\ldots,\alpha_M \in \mathbb{R}^n} \max_{\gamma \in \mathbb{R}^n} R(u) + \lambda \sum_{i=1}^{M} \frac{\alpha_i^\top K_i \alpha_i}{\eta_i} + 2\lambda \gamma^\top (u - \sum_{i=1}^{M} K_i \alpha_i) \,.$$

- By Slater's condition, strong duality holds, meaning we can invert min and max:

$$\max_{\gamma \in \mathbb{R}^n} \min_{u,\alpha_1,\ldots,\alpha_M \in \mathbb{R}^n} R(u) + \lambda \sum_{i=1}^{M} \frac{\alpha_i^\top K_i \alpha_i}{\eta_i} + 2\lambda \gamma^\top (u - \sum_{i=1}^{M} K_i \alpha_i) \,.$$

Proof (3/4)

- Minimization in u:

$$\min_u R(u) + 2\lambda\gamma^\top u = -\max_u \left\{-2\lambda\gamma^\top u - R(u)\right\} = -R^*(-2\lambda\gamma)\,,$$

where R^* is the Fenchel dual of R:

$$\forall v \in \mathbb{R}^n \quad R^*(v) = \sup_{u\in\mathbb{R}^n} u^\top v - R(u)\,.$$

- Minimization in α_i for $i = 1, \ldots, M$:

$$\min_{\alpha_i} \left\{\lambda \frac{\alpha_i^\top K_i \alpha_i}{\eta_i} - 2\lambda\gamma^\top K_i\alpha_i\right\} = -\lambda\eta_i\gamma^\top K_i\gamma\,,$$

where the minimum in α_i is reached for $\alpha_i^* = \eta_i\gamma$.

Proof (4/4)

- The dual problem is therefore

$$\max_{\gamma\in\mathbb{R}^n}\left\{-R^*(-2\lambda\gamma)-\lambda\gamma^\top\left(\sum_{i=1}^M\eta_iK_i\right)\gamma\right\}.$$

- Note that if learn from a single kernel K_η, we get the same dual problem

$$\max_{\gamma\in\mathbb{R}^n}\left\{-R^*(-2\lambda\gamma)-\lambda\gamma^\top K_\eta\gamma\right\}.$$

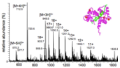
- If γ^* is a solution of the dual problem, then $\alpha_i^*=\eta_i\gamma^*$ leading to:

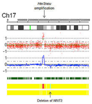
$$\forall x\in\mathcal{X},\quad f_i^*(x)=\sum_{j=1}^n\alpha_{ij}^*K_i\left(x_j,x\right)=\sum_{j=1}^n\eta_i\gamma_j^*K_i\left(x_j,x\right)$$

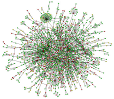
- Therefore, $f^*=\sum_{i=1}^M f_i^*$ satisfies

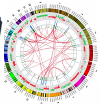
$$f^*(x)=\sum_{i=1}^M\sum_{j=1}^n\eta_i\gamma_j^*K_i\left(x_j,x\right)=\sum_{j=1}^n\gamma_j^*K_\eta\left(x_j,x\right).\quad\square$$

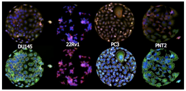
Learning the kernel

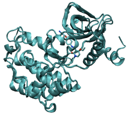












Motivation

- If we know how to weight each kernel, then we can learn with the weighted kernel

$$K_\eta = \sum_{i=1}^{M} \eta_i K_i$$

- However, usually we don't know...
- Perhaps we can optimize the weights η_i during learning?

An objective function for K

Theorem

For any p.d. kernel K on $\mathcal{X}$, let

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

The function $K \mapsto J(K)$ is convex.

This suggests a principled way to "learn" a kernel: define a convex set of candidate kernels, and minimize $J(K)$ by convex optimization.

Proof

- We have shown by strong duality that

$$J(K) = \max_{\gamma \in \mathbb{R}^n} \left\{ -R^*(-2\lambda\gamma) - \lambda\gamma^\top K\gamma \right\} .$$

- For each γ fixed, this is an affine function of K, hence convex
- A supremum of convex functions is convex. $\square$

MKL (Lanckriet et al., 2004)

- We consider the set of convex combinations

$$K_\eta = \sum_{i=1}^{M} \eta_i K_i \quad \text{with} \quad \eta \in \Sigma_M = \left\{ \eta_i \geq 0 \, , \, \sum_{i=1}^{M} \eta_i = 1 \right\}$$

- We optimize both η and f^* by solving:

$$\min_{\eta \in \Sigma_M} J(K_\eta) = \min_{\eta \in \Sigma_M} \min_{f \in \mathcal{H}_{K_\eta}} \left\{ R(f^n) + \lambda \| f \|^2_{\mathcal{H}_{K_\eta}} \right\}$$

- The problem is jointly convex in $(\eta, \boldsymbol{\alpha})$ and can be solved efficiently
- The output is both a set of weights η, and a predictor corresponding to the kernel method trained with kernel K_η.
- This method is usually called Multiple Kernel Learning (MKL).

Example: protein annotation

BIOINFORMATICS

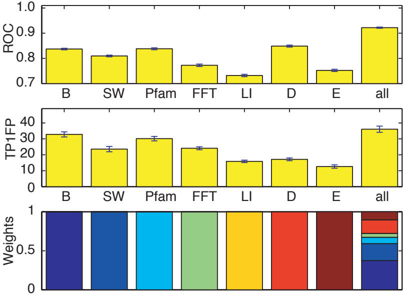
Vol. 20 no. 16 2004, pages 2626–2635
doi:10.1093/bioinformatics/bth294

A statistical framework for genomic data fusion

Gert R. G. Lanckriet[1], Tijl De Bie[3], Nello Cristianini[4], Michael I. Jordan[2] and William Stafford Noble[5,]*

[1]Department of Electrical Engineering and Computer Science, [2]Division of Computer Science, Department of Statistics, University of California, Berkeley 94720, USA, [3]Department of Electrical Engineering, ESAT-SCD, Katholieke Universiteit Leuven 3001, Belgium, [4]Department of Statistics, University of California, Davis 95618, USA and [5]Department of Genome Sciences, University of Washington, Seattle 98195, USA

Kernel	Data	Similarity measure
K_{SW}	protein sequences	Smith-Waterman
K_{B}	protein sequences	BLAST
K_{Pfam}	protein sequences	Pfam HMM
K_{FFT}	hydropathy profile	FFT
K_{LI}	protein interactions	linear kernel
K_{D}	protein interactions	diffusion kernel
K_{E}	gene expression	radial basis kernel
K_{RND}	random numbers	linear kernel

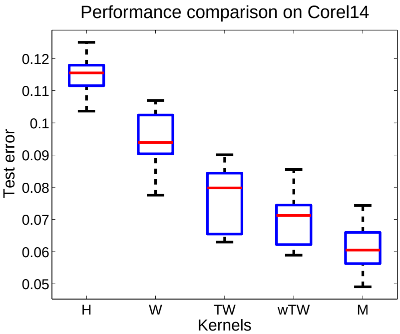


(B) Membrane proteins

Example: Image classification (Harchaoui and Bach, 2007)

COREL14 dataset

- 1400 natural images in 14 classes
- Compare kernel between histograms (H), walk kernel (W), subtree kernel (TW), weighted subtree kernel (wTW), and a combination by MKL (M).



MKL revisited (Bach et al., 2004)

$$K_\eta = \sum_{i=1}^{M} \eta_i K_i \quad \text{with} \quad \eta \in \Sigma_M = \left\{ \eta_i \geq 0 \, , \, \sum_{i=1}^{M} \eta_i = 1 \right\}$$

Theorem

The solution f^* of

$$\min_{\eta \in \Sigma_M} \min_{f \in \mathcal{H}_{K_\eta}} \left\{ R(f^n) + \lambda \| \, f \, \|^2_{\mathcal{H}_{K_\eta}} \right\}$$

is $f^* = \sum_{i=1}^{M} f_i^*$, where $\left(f_1^*, \ldots, f_M^*\right) \in \mathcal{H}_{K_1} \times \ldots \times \mathcal{H}_{K_M}$ is the solution of:

$$\min_{f_1, \ldots, f_M} \left\{ R\left(\sum_{i=1}^{M} f_i^n \right) + \lambda \left(\sum_{i=1}^{M} \| \, f_i \, \|_{\mathcal{H}_{K_i}} \right)^2 \right\} .$$

Proof (1/2)

$$
\begin{aligned}
&\min_{\eta\in\Sigma_M}\min_{f\in\mathcal{H}_{K_\eta}}\left\{R(f^n)+\lambda\| f \|^2_{\mathcal{H}_{K_\eta}}\right\}\\
&\quad=\min_{\eta\in\Sigma_M}\min_{f_1,\ldots,f_M}\left\{R\left(\sum_{i=1}^M f_i^n\right)+\lambda\sum_{i=1}^M\frac{\| f_i \|^2_{\mathcal{H}_{K_i}}}{\eta_i}\right\}\\
&\quad=\min_{f_1,\ldots,f_M}\left\{R\left(\sum_{i=1}^M f_i^n\right)+\lambda\min_{\eta\in\Sigma_M}\left\{\sum_{i=1}^M\frac{\| f_i \|^2_{\mathcal{H}_{K_i}}}{\eta_i}\right\}\right\}\\
&\quad=\min_{f_1,\ldots,f_M}\left\{R\left(\sum_{i=1}^M f_i^n\right)+\lambda\left(\sum_{i=1}^M\| f_i \|_{\mathcal{H}_{K_i}}\right)^2\right\},
\end{aligned}
$$

Proof (2/2)

where the last equality results from:

$$\forall a \in \mathbb{R}_+^M , \quad \left(\sum_{i=1}^{M} a_i \right)^2 = \inf_{\eta \in \Sigma_M} \sum_{i=1}^{M} \frac{a_i^2}{\eta_i} ,$$

which is a direct consequence of the Cauchy-Schwarz inequality:

$$\sum_{i=1}^{M} a_i = \sum_{i=1}^{M} \frac{a_i}{\sqrt{\eta_i}} \times \sqrt{\eta_i} \leq \left(\sum_{i=1}^{M} \frac{a_i^2}{\eta_i} \right)^{\frac{1}{2}} \left(\sum_{i=1}^{M} \eta_i \right)^{\frac{1}{2}} .$$

Algorithm: simpleMKL (Rakotomamonjy et al., 2008)

- We want to minimize in $\eta \in \Sigma_M$:

$$\min_{\eta \in \Sigma_M} J(K_\eta) = \min_{\eta \in \Sigma_M} \max_{\gamma \in \mathbb{R}^n} \left\{ -R^*(-2\lambda\gamma) - \lambda\gamma^\top K_\eta \gamma \right\} .$$

- For a fixed $\eta \in \Sigma_M$, we can compute $f(\eta) = J(K_\eta)$ by using a standard solver for a single kernel to find γ^*:

$$J(K_\eta) = -R^*(-2\lambda\gamma^*) - \lambda{\gamma^*}^\top K_\eta \gamma^* .$$

- From γ^* we can also compute the gradient of $J(K_\eta)$ with respect to η:

$$\frac{\partial J(K_\eta)}{\partial \eta_i} = -\lambda{\gamma^*}^\top K_i \gamma^* .$$

- $J(K_\eta)$ can then be minimized on Σ_M by a projected gradient or reduced gradient algorithm.

Sum kernel vs MKL

- Learning with the sum kernel (uniform combination) solves

$$\min_{f_1,\ldots,f_M} \left\{ R\left(\sum_{i=1}^{M} f_i^n\right) + \lambda \sum_{i=1}^{M} \| f_i \|_{\mathcal{H}_{K_i}}^2 \right\} .$$

- Learning with MKL (best convex combination) solves

$$\min_{f_1,\ldots,f_M} \left\{ R\left(\sum_{i=1}^{M} f_i^n\right) + \lambda \left(\sum_{i=1}^{M} \| f_i \|_{\mathcal{H}_{K_i}}\right)^2 \right\} .$$

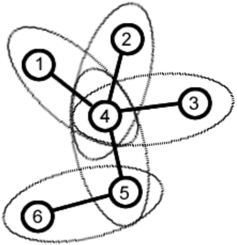
- Although MKL can be thought of as optimizing a convex combination of kernels, it is more correct to think of it as a penalized risk minimization estimator with the group lasso penalty:

$$\Omega(f) = \min_{f_1+\ldots+f_M=f} \sum_{i=1}^{M} \| f_i \|_{\mathcal{H}_{K_i}} .$$

Example: ridge vs LASSO regression

- Take $\mathcal{X} = \mathbb{R}^d$, and for $x = (x_1, \ldots, x_d)^\top$ consider the rank-1 kernels:
$$\forall i = 1, \ldots, d\,, \quad K_i\left(x, x'\right) = x_i x_i'\,.$$
- The sum kernel is $K_S\left(x, x'\right) = \sum_{i=1}^d x_i x_i' = x^\top x$
- Learning with the sum kernel solves a ridge regression problem:
$$\min_{\beta \in \mathbb{R}^d} \left\{ R(X\beta) + \lambda \sum_{i=1}^d \beta_i^2 \right\}\,.$$
- Learning with MKL solves a LASSO regression problem:
$$\min_{\beta \in \mathbb{R}^d} \left\{ R(X\beta) + \lambda \left(\sum_{i=1}^d \mid \beta_i \mid \right)^2 \right\}\,.$$

Example: Graph lasso (Jacob et al., 2009)

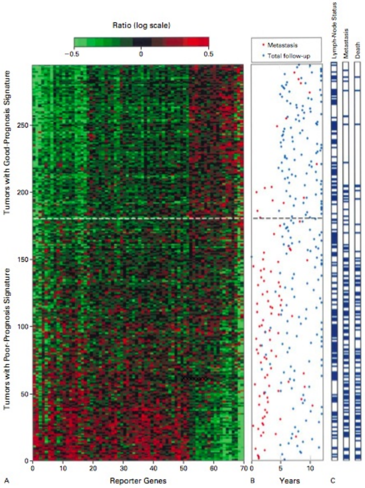


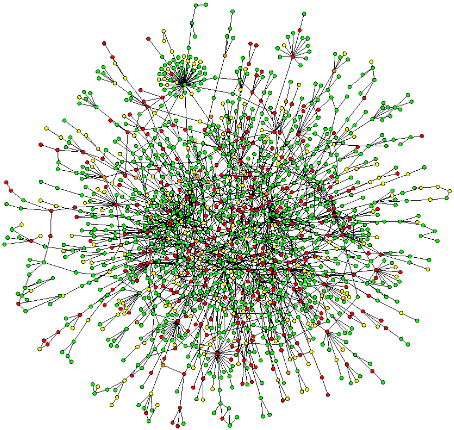
- Graph $G = (V, E)$, $\mathcal{X} = \mathbb{R}^V$
- For each edge $e = (i, j)$, define the kernel

$$K_e(x, x') = x_e^\top x'_e = x_i x'_i + x_j x'_j$$

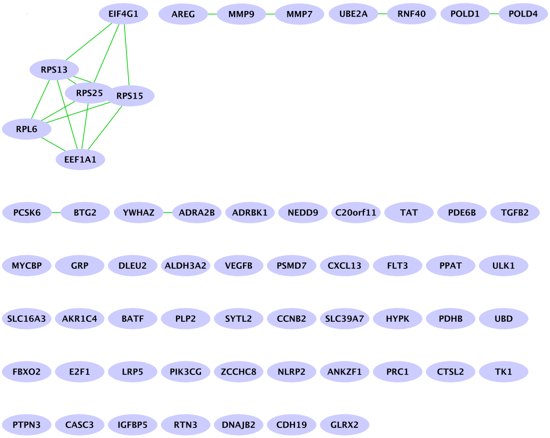
- MKL (aka latent group lasso) with the set $\{K_e \,:\, e \in E\}$ leads to a sparse linear model with connected non-zero components.

Application: breast cancer prognosis

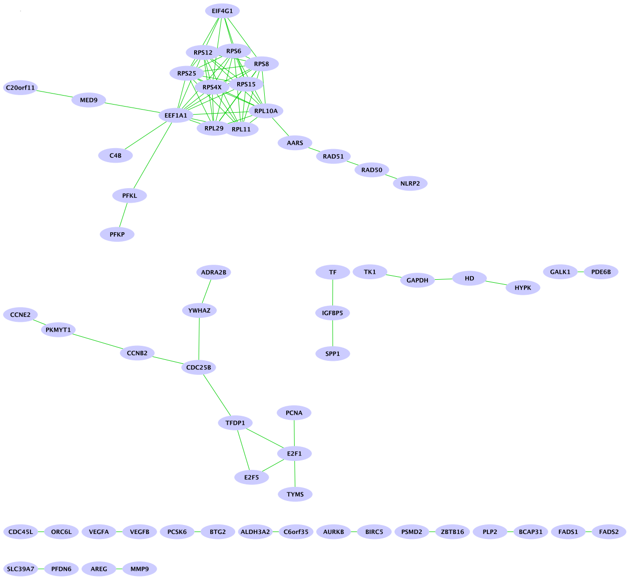




Lasso signature (accuracy 0.61)



Graph Lasso signature (accuracy 0.64)



References

N. Aronszajn. Theory of reproducing kernels. *Trans. Am. Math. Soc.*, 68:337 – 404, 1950. URL http://www.jstor.org/stable/1990404.

F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 6, New York, NY, USA, 2004. ACM. doi: http://doi.acm.org/10.1145/1015330.1015424.

K. M. Borgwardt and H.-P. Kriegel. Shortest-path kernels on graphs. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 74–81, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2278-5. doi: http://dx.doi.org/10.1109/ICDM.2005.132.

Z. Harchaoui and F. Bach. Image classification with segmentation graph kernels. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pages 1–8. IEEE Computer Society, 2007. doi: 10.1109/CVPR.2007.383049. URL http://dx.doi.org/10.1109/CVPR.2007.383049.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2001.

D. Haussler. Convolution Kernels on Discrete Structures. Technical Report UCSC-CRL-99-10, UC Santa Cruz, 1999.

C. Helma, T. Cramer, S. Kramer, and L. De Raedt. Data mining and machine learning techniques for the identification of mutagenicity inducing substructures and structure activity relationships of noncongeneric compounds. *J. Chem. Inf. Comput. Sci.*, 44(4):1402–11, 2004. doi: 10.1021/ci034254q. URL http://dx.doi.org/10.1021/ci034254q.

References (cont.)

A. E. Hoerl and R. W. Kennard. Ridge regression : biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 433–440, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: http://doi.acm.org/10.1145/1553374.1553431.

G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004a. URL http://www.jmlr.org/papers/v5/lanckriet04a.html.

G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004b. doi: 10.1093/bioinformatics/bth294. URL http://bioinformatics.oupjournals.org/cgi/content/abstract/20/16/2626.

S. Le Cessie and J. C. van Houwelingen. Ridge estimators in logistic regression. *Appl. Statist.*, 41(1):191–201, 1992. URL http://www.jstor.org/stable/2347628.

C. Leslie and R. Kuang. Fast string kernels using inexact matching for protein sequences. *J. Mach. Learn. Res.*, 5:1435–1455, 2004.

C. Leslie, E. Eskin, and W. Noble. The spectrum kernel: a string kernel for SVM protein classification. In R. B. Altman, A. K. Dunker, L. Hunter, K. Lauerdale, and T. E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing 2002*, pages 564–575, Singapore, 2002. World Scientific.

References (cont.)

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. n. p. v. d. d. r. Watkins. Text classification using string kernels. *J. Mach. Learn. Res.*, 2:419–444, 2002. URL `http://www.ai.mit.edu/projects/jmlr/papers/volume2/lodhi02a/abstract.html`.

P. Mahé and J. P. Vert. Graph kernels based on tree patterns for molecules. *Mach. Learn.*, 75(1): 3–35, 2009. doi: 10.1007/s10994-008-5086-2. URL `http://dx.doi.org/10.1007/s10994-008-5086-2`.

A. Nicholls. Oechem, version 1.3.4, openeye scientific software. website, 2005.

A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *J. Mach. Learn. Res.*, 9: 2491–2521, 2008. URL `http://jmlr.org/papers/v9/rakotomamonjy08a.html`.

J. Ramon and T. Gärtner. Expressivity versus efficiency of graph kernels. In T. Washio and L. De Raedt, editors, *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences*, pages 65–74, 2003.

F. Rapaport, A. Zynoviev, M. Dutreix, E. Barillot, and J.-P. Vert. Classification of microarray data using gene networks. *BMC Bioinformatics*, 8:35, 2007. doi: 10.1186/1471-2105-8-35. URL `http://dx.doi.org/10.1186/1471-2105-8-35`.

H. Saigo, J.-P. Vert, N. Ueda, and T. Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689, 2004. URL `http://bioinformatics.oupjournals.org/cgi/content/abstract/20/11/1682`.

N. Sherashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt. Efficient graphlet kernels for large graph comparison. In *12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 488–495, Clearwater Beach, Florida USA, 2009. Society for Artificial Intelligence and Statistics.

References (cont.)

Y. Yamanishi, J.-P. Vert, and M. Kanehisa. Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20:i363–i370, 2004. URL http://bioinformatics.oupjournals.org/cgi/reprint/19/suppl_1/i323.