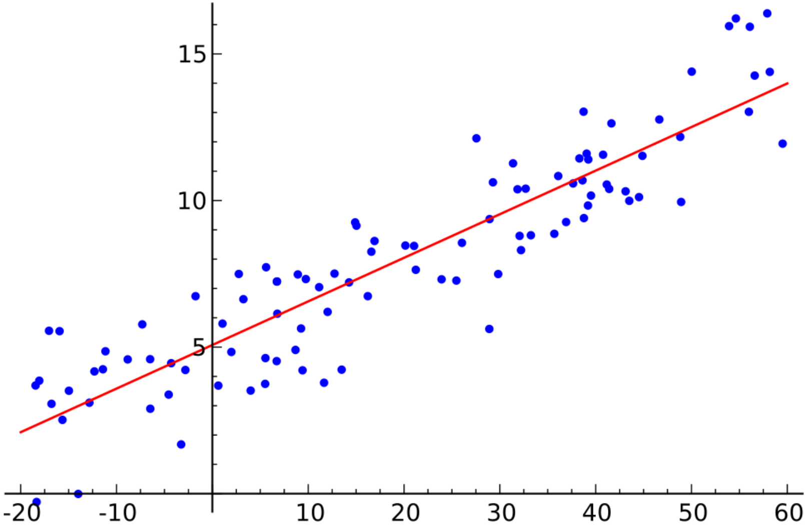
Learning with kernels : an introduction

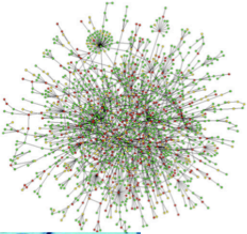
Jean-Philippe Vert
Jean-Philippe.Vert@mines.org

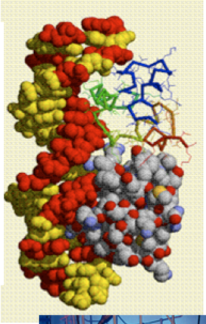
MINES ParisTech
institut Curie
Together, let's beat cancer.

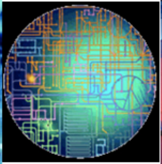
What we know how to solve



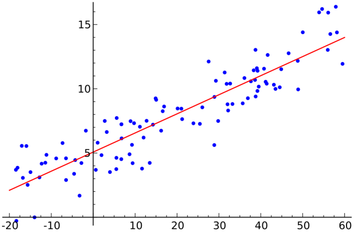
But real data are often more complicated...

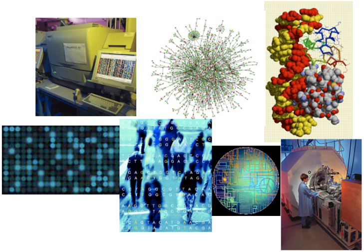






Main goal of this course





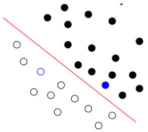
Extend well-understood, linear statistical learning techniques to nonlinear techniques for real-world, complicated, structured, high-dimensional data (images, texts, time series, graphs, distributions, permutations...)

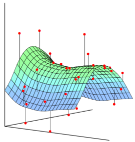
Outline

1. Penalized empirical risk minimization
2. Learning with ℓ_2 regularization
3. Kernel methods
4. Learning molecular classifiers with network information
5. Data integration with kernels

Outline

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

$$\hat{f} \in \underset{f \in \mathcal{F}}{\operatorname{argmin}} \; R_n(f)$$

Example: ordinary least squares (OLS)

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

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

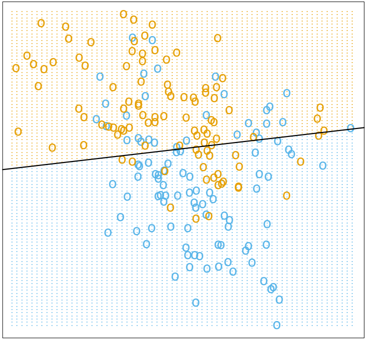
- $\ell(t, y) = (t - y)^2$ is the squared error
- The empirical risk is the mean squared error (MSE):

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

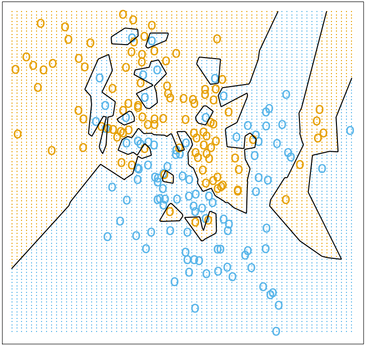
- When $\mathbf{X}^\top \mathbf{X}$ is non-singular, the ERM estimator is

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

The curse of dimensionality



Small dimension



Large dimension

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

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

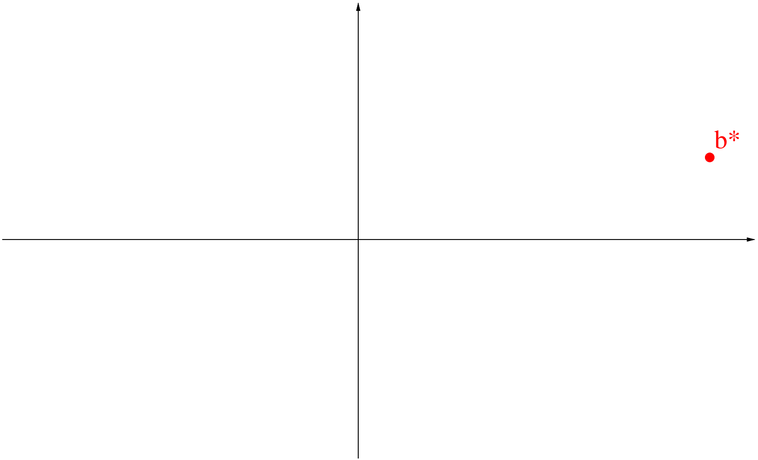
$$\hat{f} \in \operatorname*{argmin}_{f \in \mathcal{F}} R_n(f) \quad \text{such that} \quad \Omega(f) \leq C$$

or

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

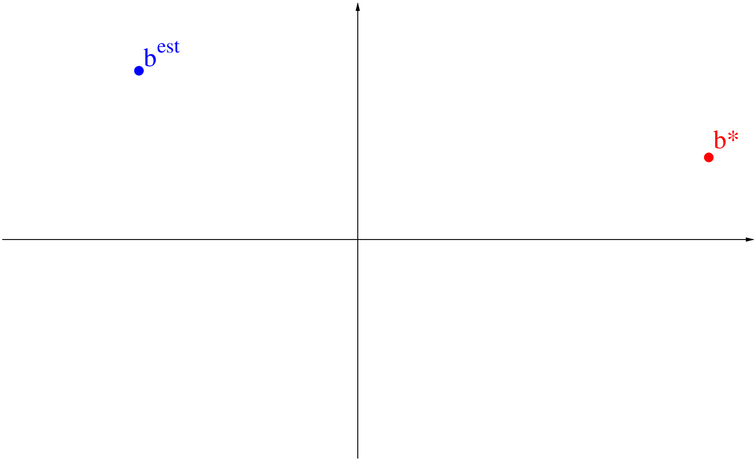
Why skrinkage classifiers?

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



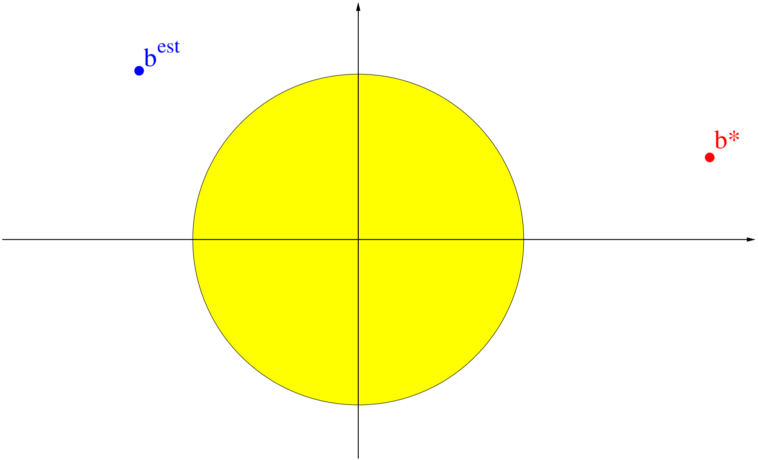
Why skrinkage classifiers?

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



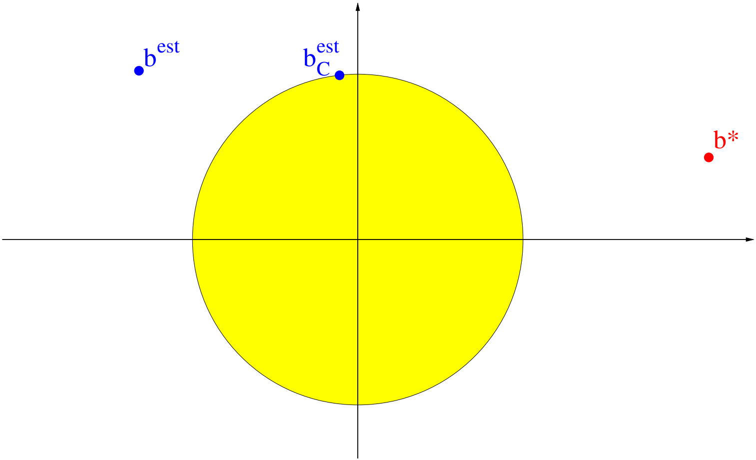
Why skrinkage classifiers?

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



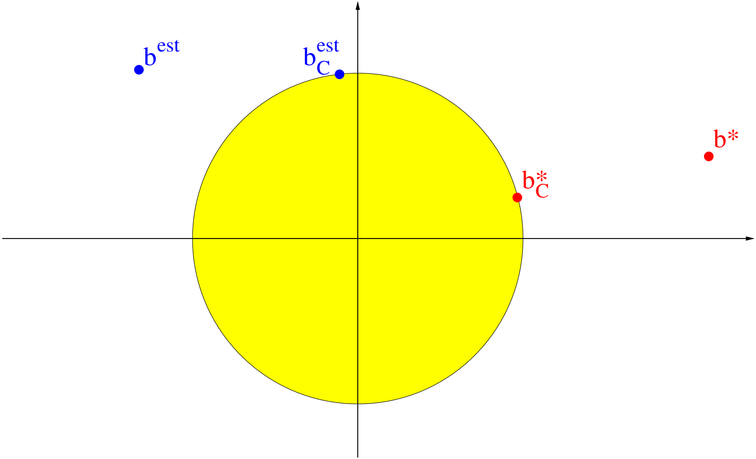
Why skrinkage classifiers?

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



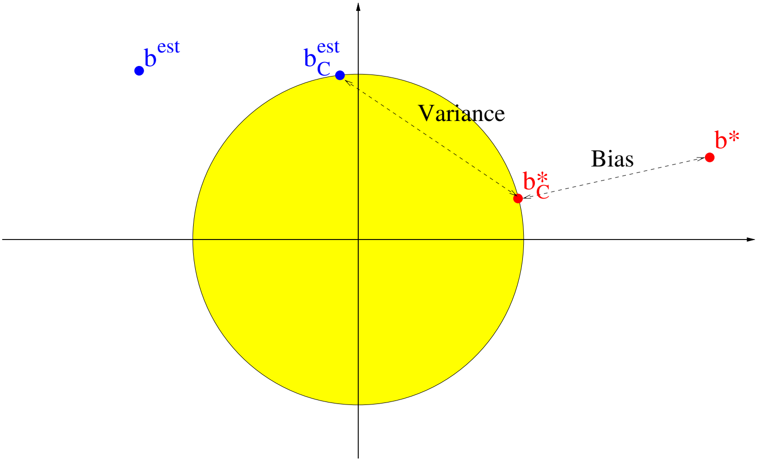
Why skrinkage classifiers?

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



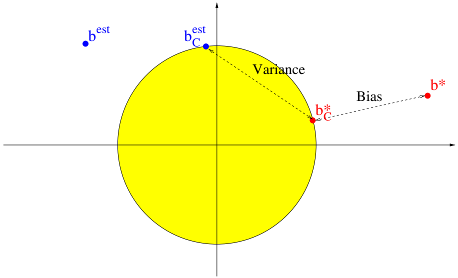
Why skrinkage classifiers?

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



Why skrinkage classifiers?

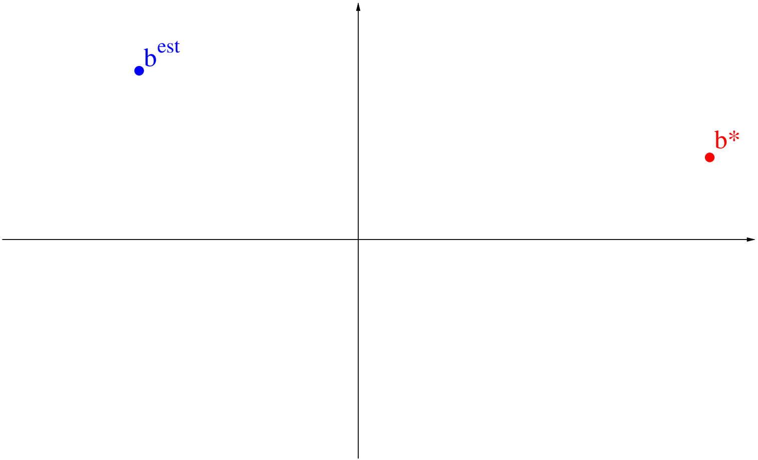
$$\min_{\beta} R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C\,.$$



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

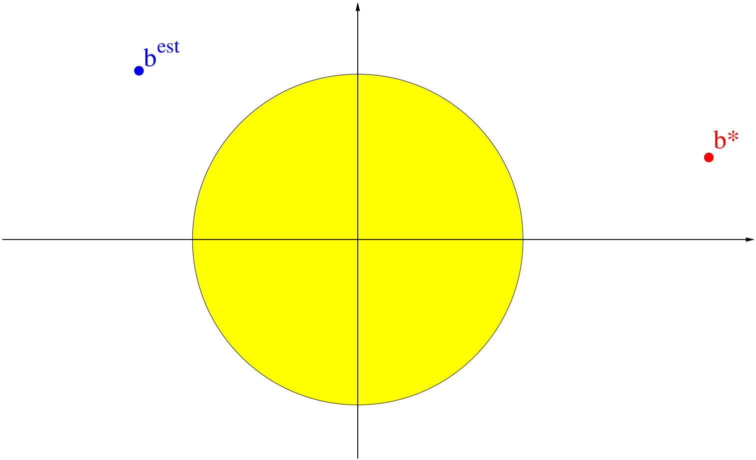
Choice of Ω can decrease the bias

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



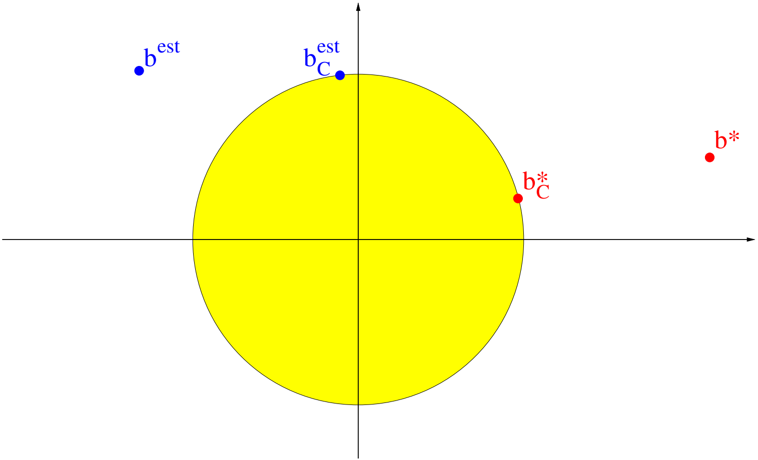
Choice of Ω can decrease the bias

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



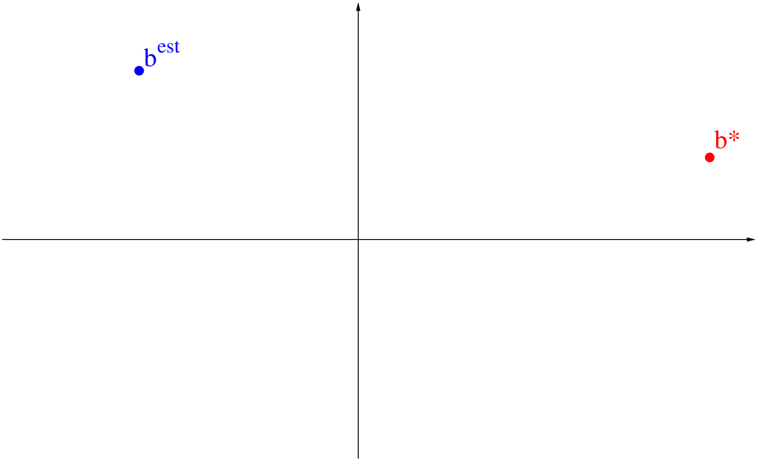
Choice of Ω can decrease the bias

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



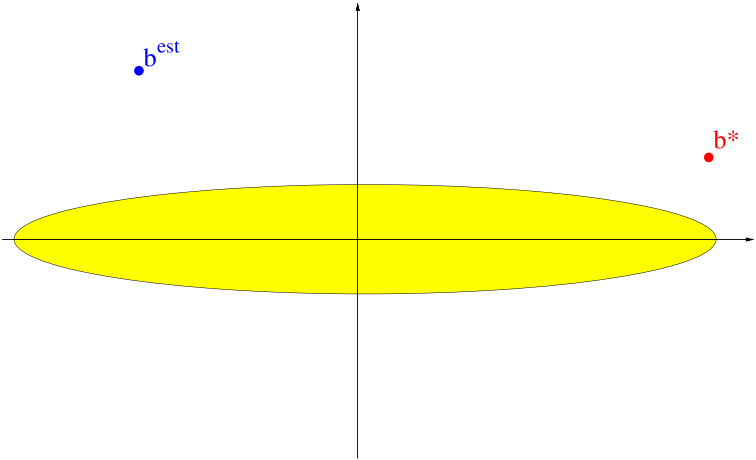
Choice of Ω can decrease the bias

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



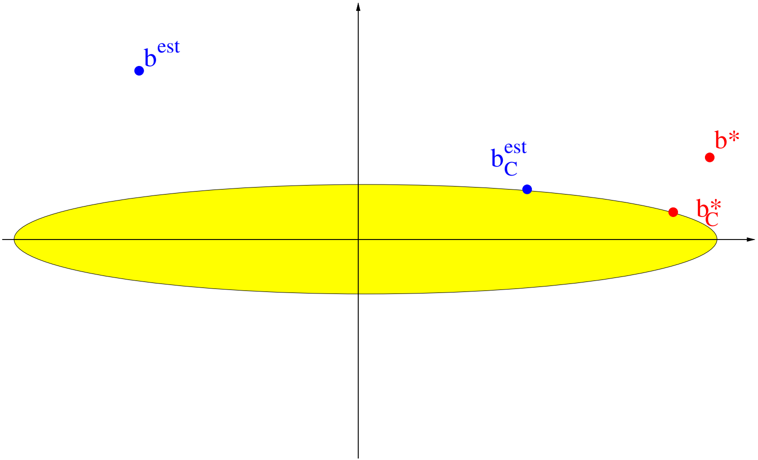
Choice of Ω can decrease the bias

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

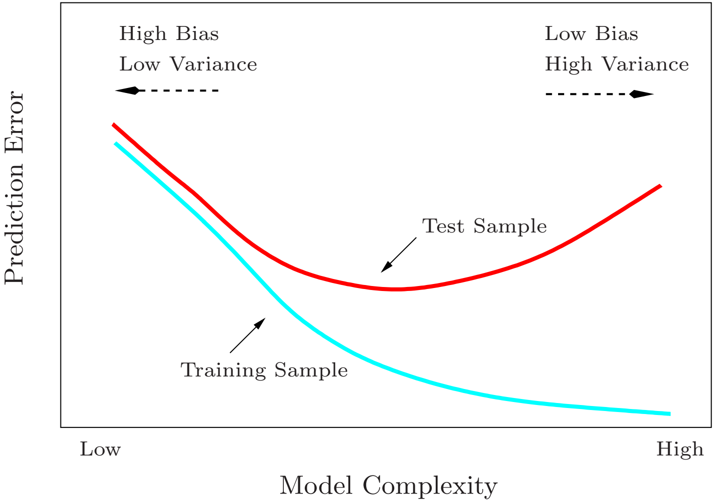


Choice of Ω can decrease the bias

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



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

1. Penalized empirical risk minimization
2. Learning with ℓ_2 regularization
3. Kernel methods
4. Learning molecular classifiers with network information
5. Data integration with kernels

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

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

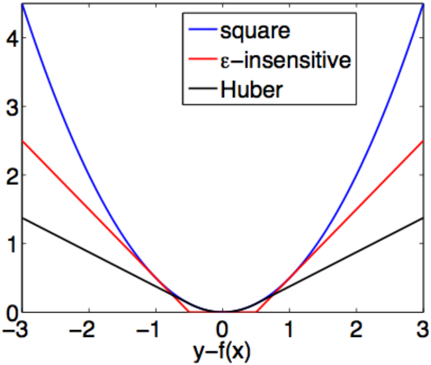
where

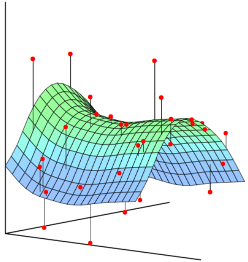
$$R(\beta) = \frac{1}{n} \sum_{i=1}^{n} \ell(f_\beta(x_i), y_i)$$

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:

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

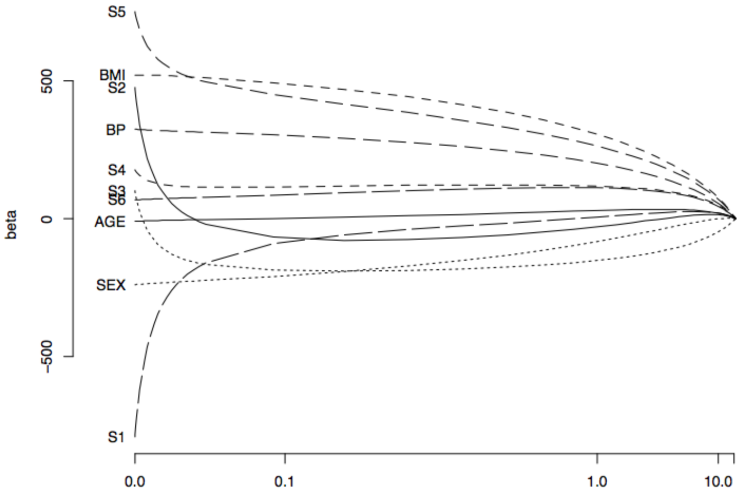
Limit cases

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

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

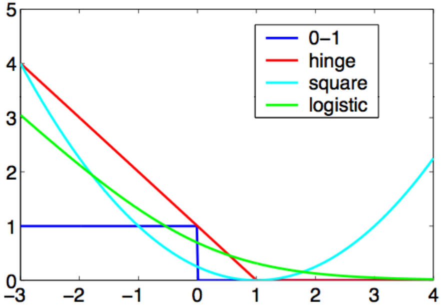
Ridge regression is particularly useful in the presence of correlated features:

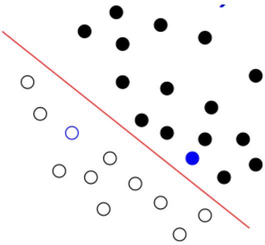
```
> library(MASS) # for the lm.ridge command
> x1 <- rnorm(20)
> x2 <- rnorm(20,mean=x1,sd=.01)
> y <- rnorm(20,mean=3+x1+x2)
> lm(y~x1+x2)$coef
(Intercept)          x1          x2
   3.070699   25.797872  -23.748019
> lm.ridge(y~x1+x2,lambda=1)
                  x1        x2
3.066027 1.015862 0.956560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

Loss for pattern recognition

Large margin classifiers

- For pattern recognition $\mathcal{Y} = \{-1, 1\}$
- Estimate a function $f : \mathcal{X} \rightarrow \mathbb{R}$.
- 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:

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

for the following model:

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

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.)

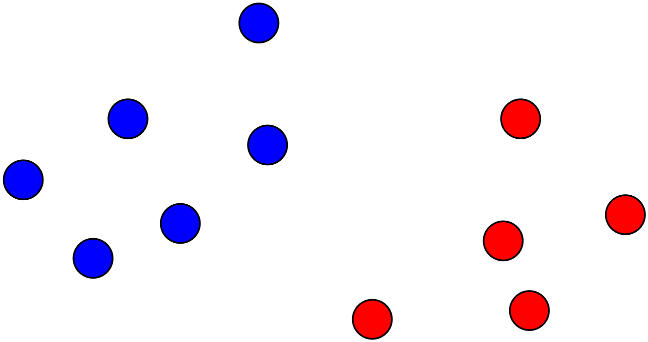
$$\min_{\beta} J(\beta) = \frac{1}{n} \sum_{i=1}^{n} \ln \left(1 + e^{-y_i \beta^\top x_i}\right) + \lambda \|\beta\|_2^2$$

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

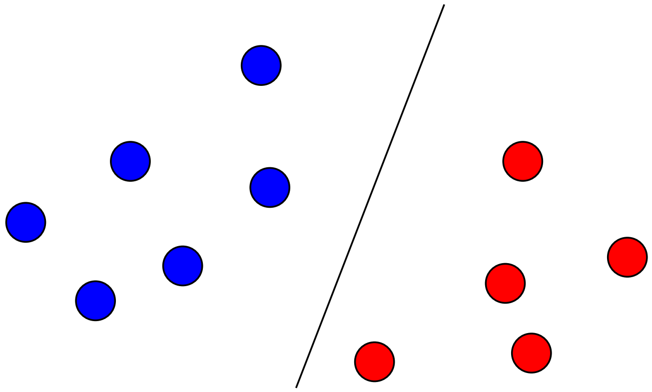
$$\beta^{new} \leftarrow \beta^{old} - \left[\nabla_\beta^2 J\left(\beta^{old}\right)\right]^{-1} \nabla_\beta J\left(\beta^{old}\right) .$$

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

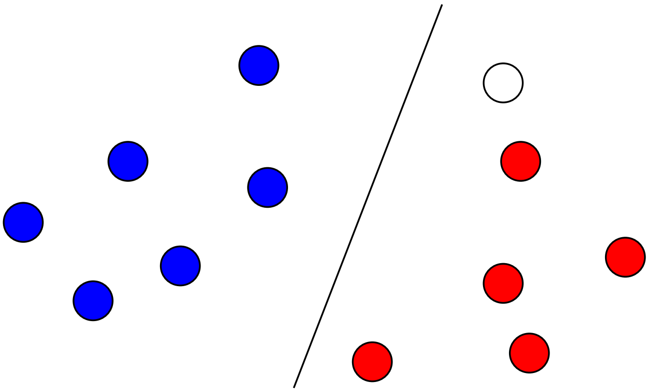
Example: hard-margin SVM



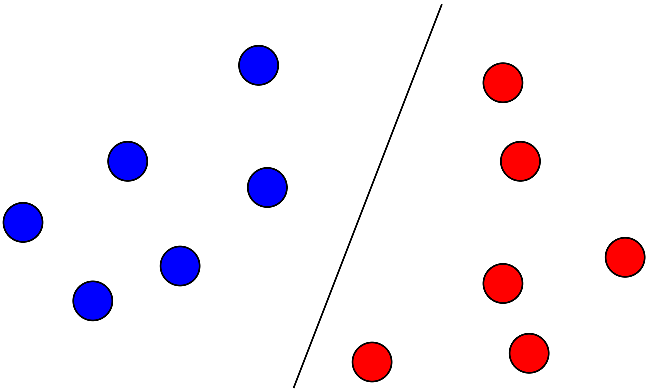
Example: hard-margin SVM



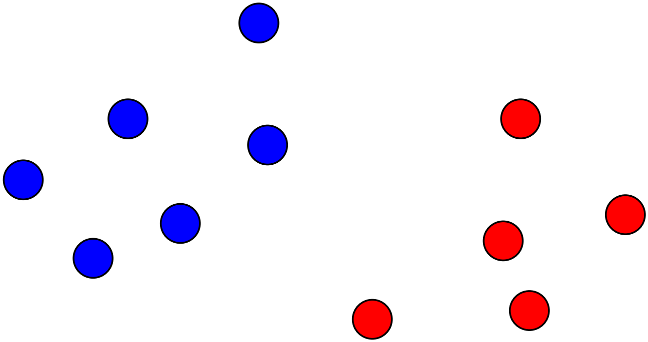
Example: hard-margin SVM



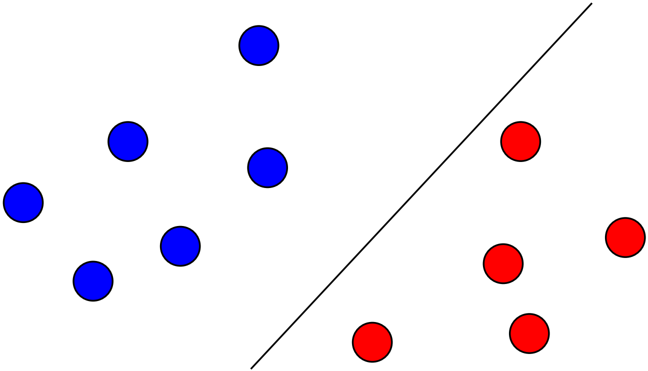
Example: hard-margin SVM



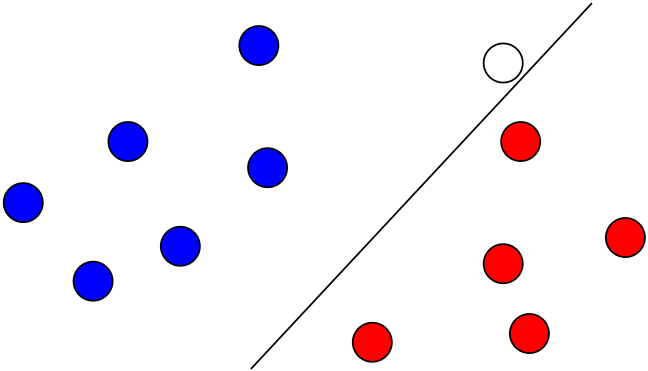
Example: hard-margin SVM



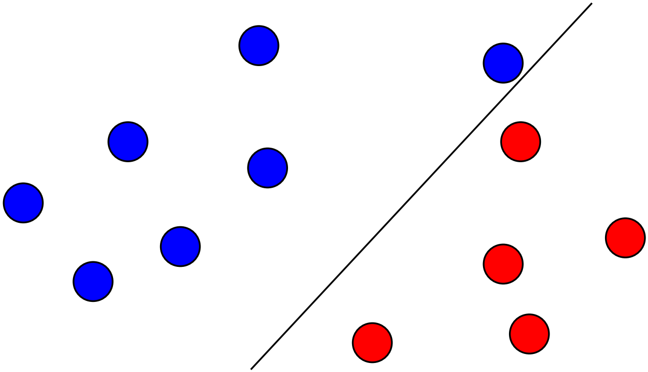
Example: hard-margin SVM



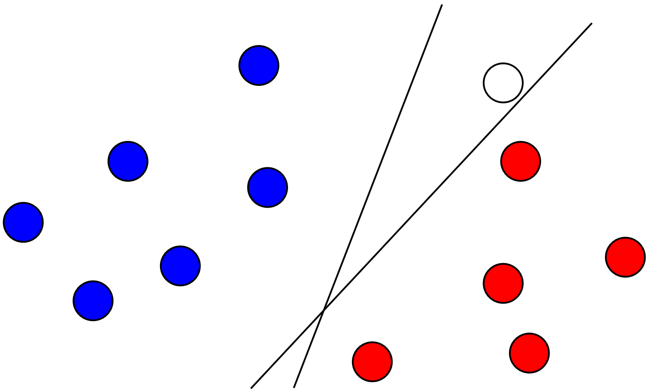
Example: hard-margin SVM



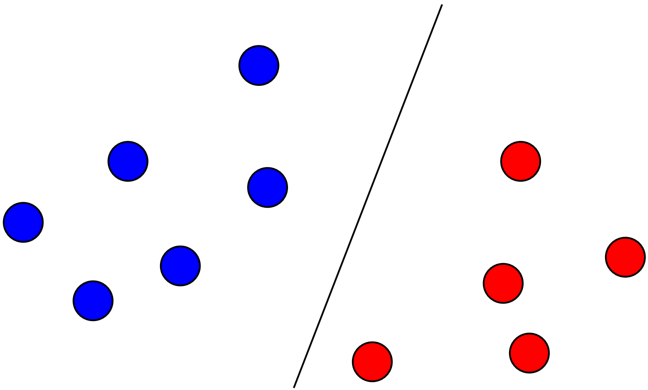
Example: hard-margin SVM



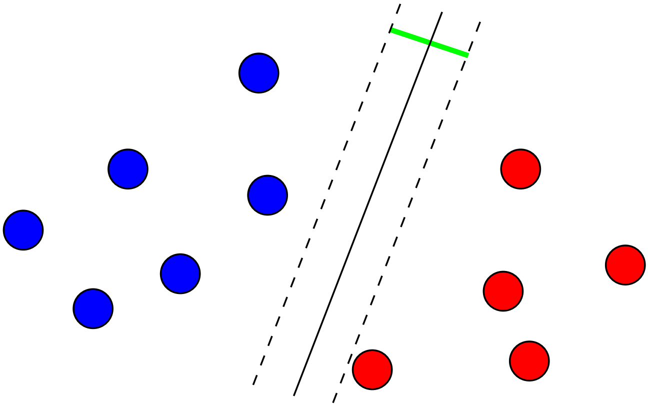
Example: hard-margin SVM



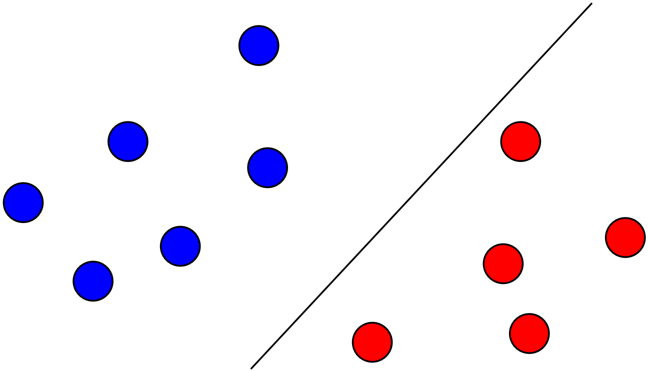
Example: hard-margin SVM



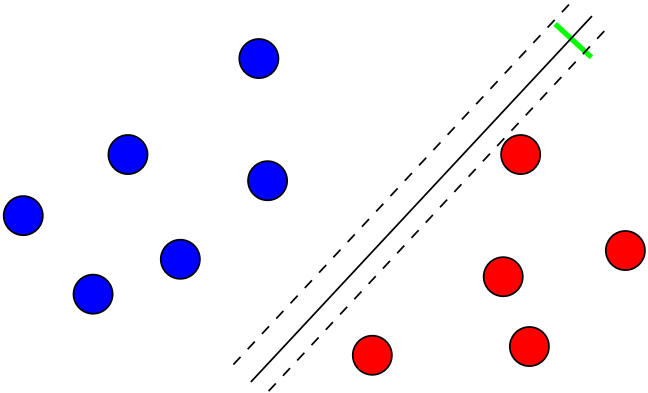
Example: hard-margin SVM



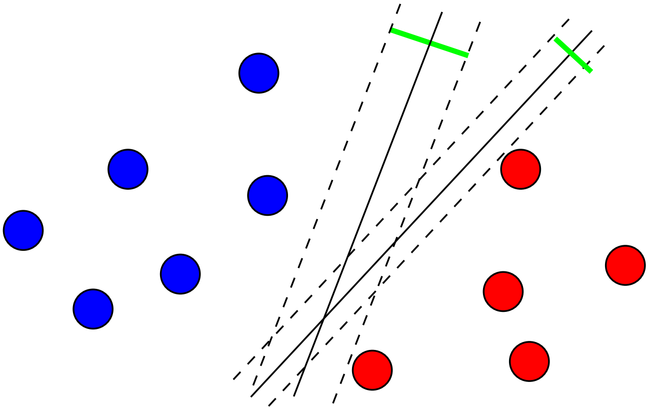
Example: hard-margin SVM



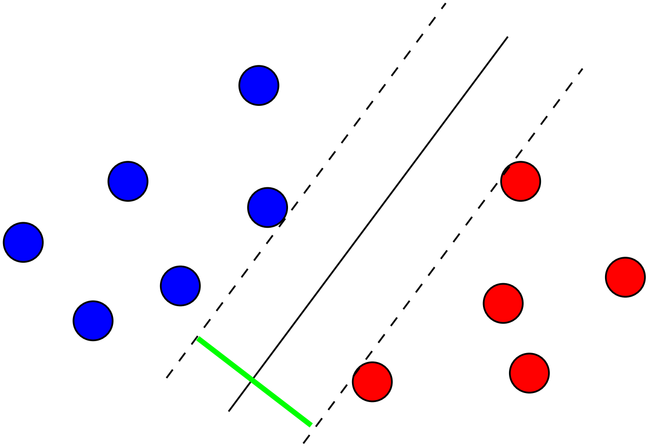
Example: hard-margin SVM



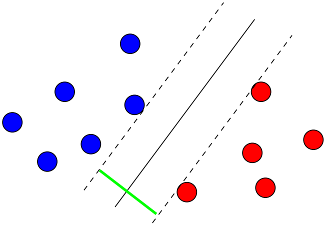
Example: hard-margin SVM



Example: hard-margin SVM



Hard-margin SVM is an ℓ_2-regularized method



Exercice

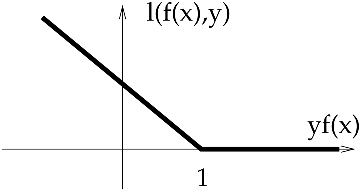
Show that hard-margin SVM solve a problem of the form

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

What is ϕ_{hard}?

Example: (soft-margin) SVM

- The hinge loss

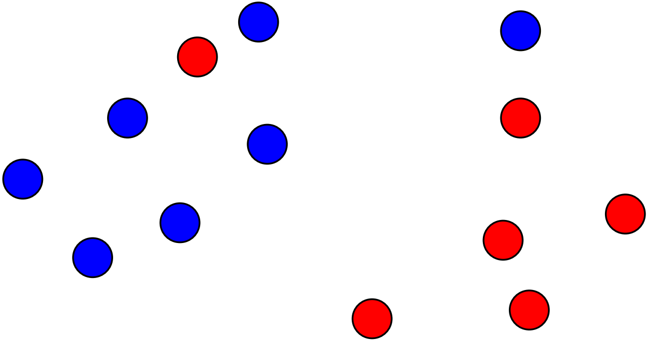


$$\phi_{\text{hinge}}(u) = \max(1-u, 0) = \begin{cases} 0 & \text{if } u \geq 1, \\ 1-u & \text{otherwise.} \end{cases}$$

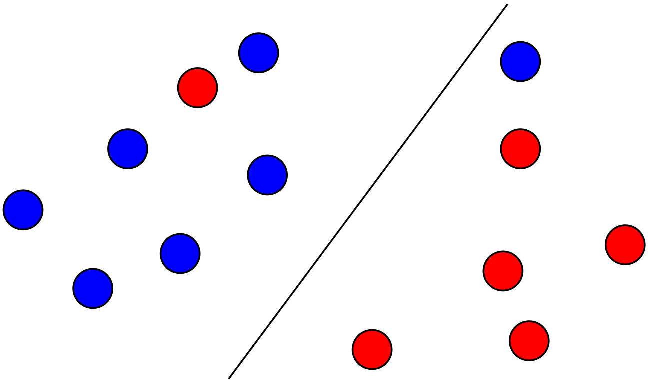
- SVM solves:

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

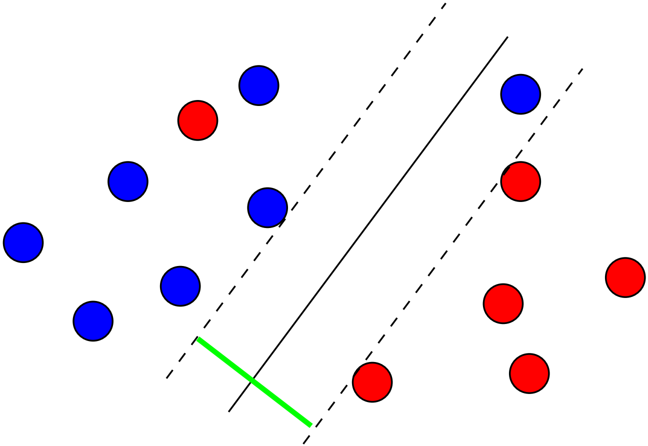
SVM: graphical interpretation



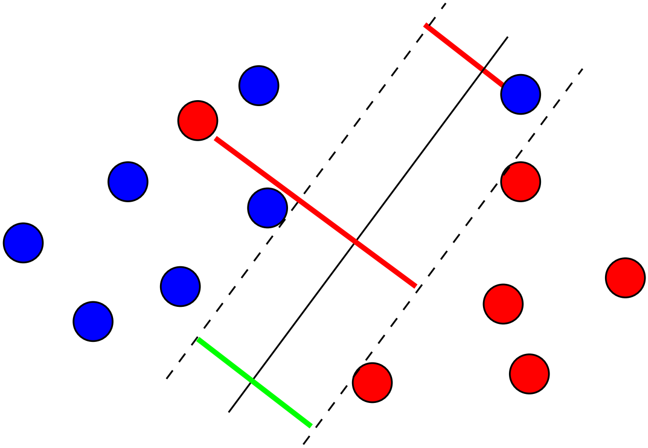
SVM: graphical interpretation



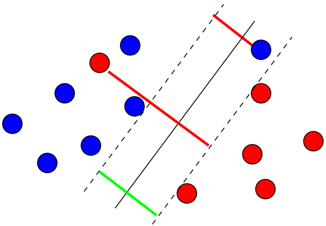
SVM: graphical interpretation



SVM: graphical interpretation



SVM: graphical interpretation



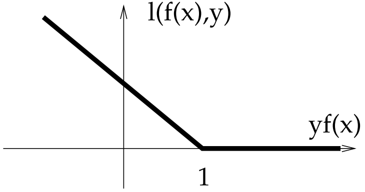
Exercice

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):

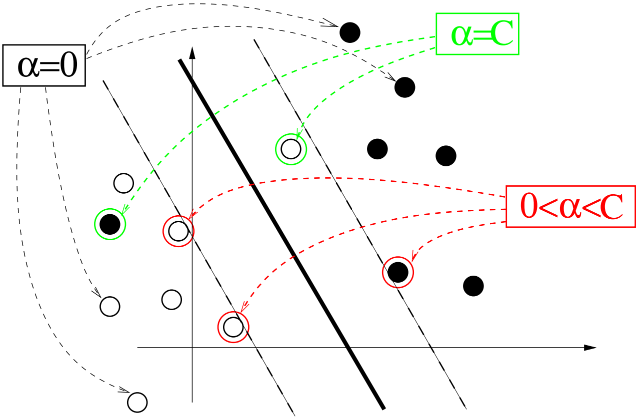
$$\nabla_\beta L = \beta - \sum_{i=1}^{n} \alpha_i y_i x_i \quad \Longrightarrow \quad \beta = \sum_{i=1}^{n} \alpha_i y_i x_i$$

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

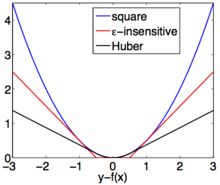
Dual problem

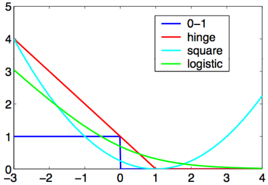
$$\max_{0 \leq \alpha \leq \frac{1}{2n\lambda}} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^\top x_j \right\}$$

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



Summary: ℓ_2-regularized linear methods



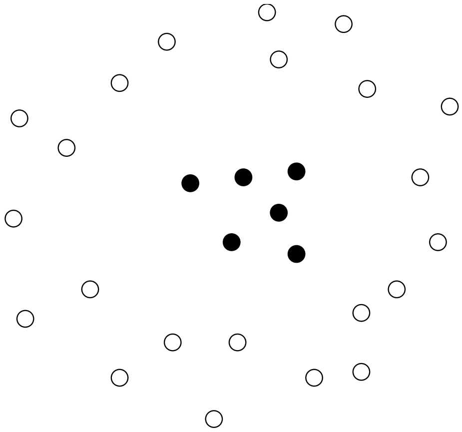


$$f_\beta(x) = \beta^\top x\,, \quad \min_\beta \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(f_\beta(x_i), y_i) + \lambda \|\beta\|_2^2 \right\}$$

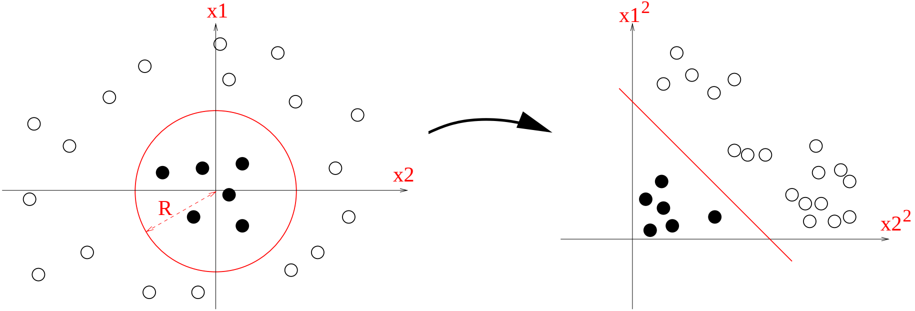
- Many popular methods for regression and classification are obtained by changing the loss function: ridge regression, logistic regression, SVM...
- Needs to solve numerically a convex optimization problem, well adapted to large datasets (stochastic gradient...)
- In practice, very similar performance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

Outline

Sometimes linear methods are not interesting



Solution: non-linear mapping to a feature space



Let $\vec{\Phi}(\vec{x}) = (x_1^2, x_2^2)'$, $\vec{w} = (1, 1)'$ and $b = 1$. Then the decision function is:

$$f(\vec{x}) = x_1^2 + x_2^2 - R^2 = \vec{w}.\vec{\Phi}(\vec{x}) + b,$$

Kernels

Definition

For a given mapping Φ from the space of objects $\mathcal{X}$ to some feature space, 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: if $\vec{\Phi}(\vec{x}) = (x_1^2, x_2^2)'$, *then*

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

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

$$L(\vec{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \vec{x}_i^\top \vec{x}_j,$$

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

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

Trick 1 illustration: SVM in the feature space

- Train the SVM by maximizing

$$L(\vec{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \Phi\left(\vec{x}_i\right)^\top \Phi\left(\vec{x}_j\right) ,$$

  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

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

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

- Train the SVM by maximizing

$$L(\vec{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K\left(\vec{x}_i, \vec{x}_j\right) ,$$

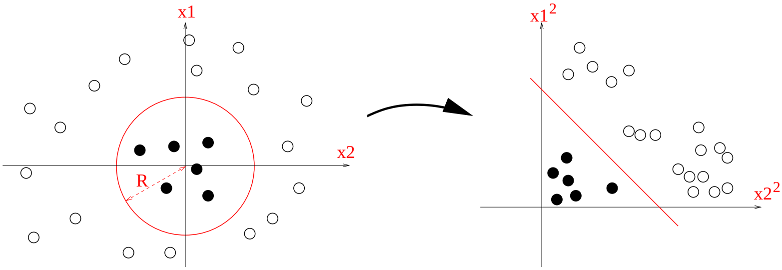
under the constraints:

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

- Predict with the decision function

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

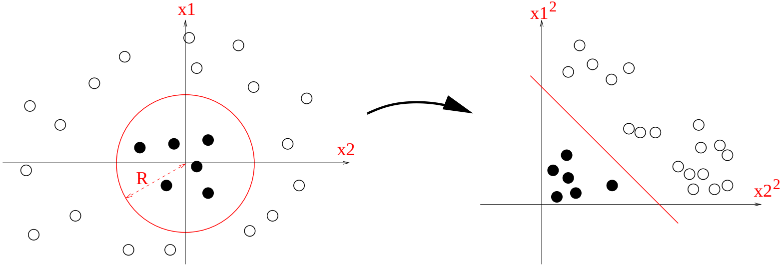
Trick 2 illustration: polynomial kernel



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

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

Trick 2 illustration: polynomial kernel



More generally,

$$K(\vec{x},\vec{x}') = \left(\vec{x}.\vec{x}' + 1\right)^d$$

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

$$L(\vec{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\vec{x}_i^\top \vec{x}_j + 1 \right)^d ,$$

  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

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

Illustration: toy nonlinear problem

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

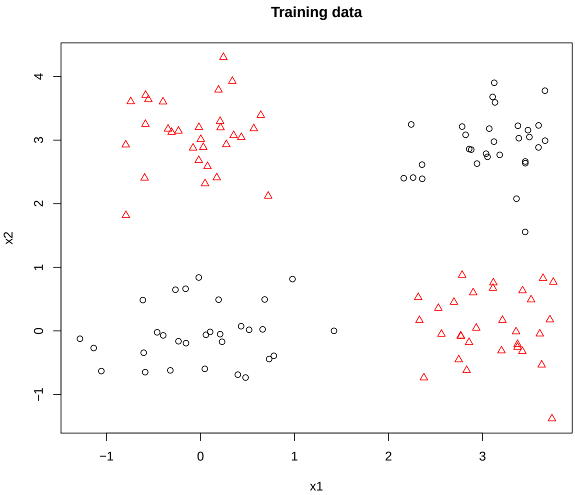


Illustration: toy nonlinear problem, linear SVM

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

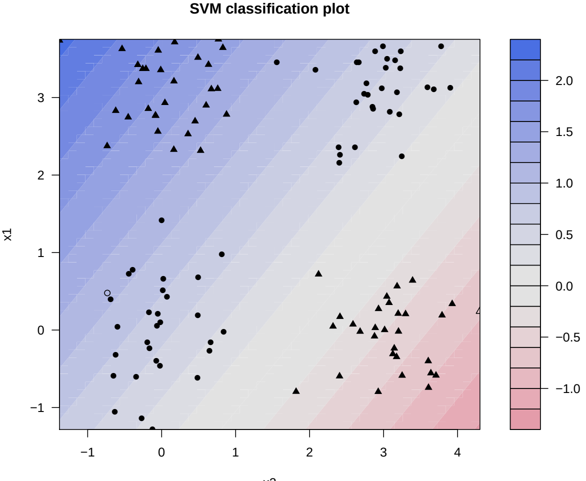
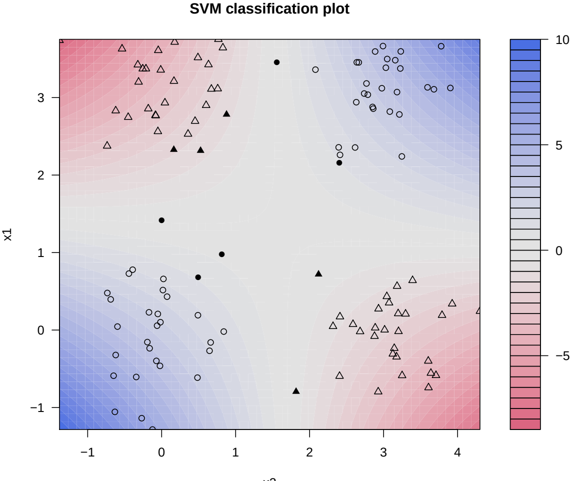


Illustration: toy nonlinear problem, polynomial SVM

```
> 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 $\beta_\mathcal{S} \in \mathit{span}(\Phi(x_1), \ldots, \Phi(x_n))$ 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:

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

can be expanded as:

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

Example: kernel ridge regression

- Let $Y = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$ the vector of response variables.
- Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{R}^n$ 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:

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

- 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 α. Its minimum can therefore be found by setting the gradient in α to zero:

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

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:

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

- By Lagrange duality we saw that this is equivalent to

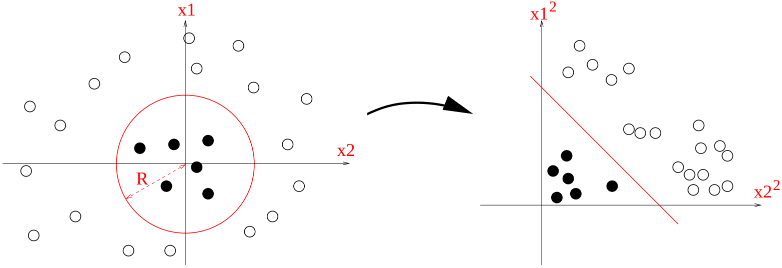
$$\max_{\alpha\in\mathbb{R}^n} L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) ,$$

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*).

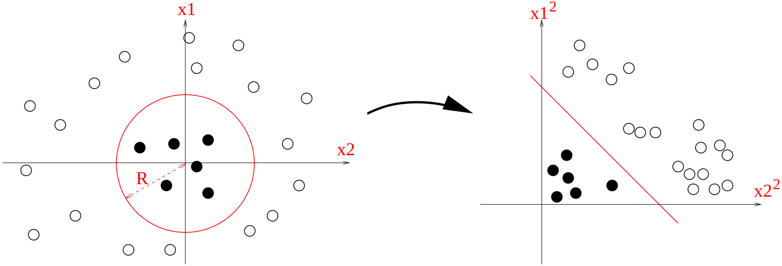
Kernel example: polynomial kernel



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

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

Kernel example: polynomial kernel



More generally,

$$K(\vec{x}, \vec{x}') = \left(\vec{x}.\vec{x}' + 1\right)^d$$

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

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

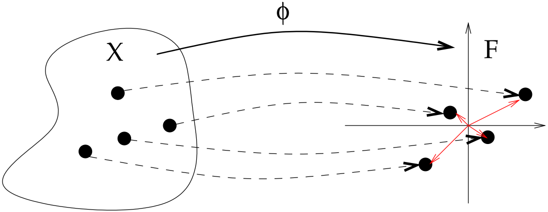
Definition

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:

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

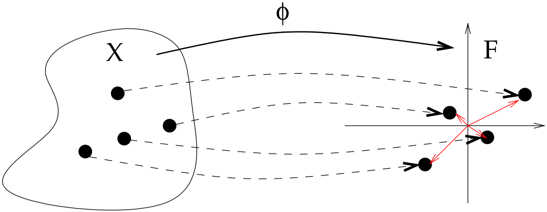
and which satisfies, for all $N \in \mathbb{N}$, $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \in \mathcal{X}^N$ et $(a_1, a_2, \ldots, a_N) \in \mathbb{R}^N$:

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

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.

- $\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\mathbb{R}^d} = \langle \Phi(\mathbf{x}'), \Phi(\mathbf{x})_{\mathbb{R}^d} \rangle$,
- $\sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j \left\langle \Phi(\mathbf{x}_i), \Phi\left(\mathbf{x}_j\right) \right\rangle_{\mathbb{R}^d} = \| \sum_{i=1}^{N} a_i \Phi(\mathbf{x}_i) \|_{\mathbb{R}^d}^2 \geq 0$.

Proof: p.d. $\implies$ kernel (1/5)

- Assume $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is p.d.
- For any $\mathbf{x} \in \mathcal{X}$, let $K_{\mathbf{x}} : \mathcal{X} \mapsto \mathbb{R}$ defined by:

$$K_{\mathbf{x}} : \mathbf{t} \mapsto K(\mathbf{x}, \mathbf{t}) \ .$$

- Let $\mathcal{H}_0$ be the vector subspace of $\mathbb{R}^{\mathcal{X}}$ spanned by the functions $\{K_{\mathbf{x}}\}_{\mathbf{x} \in \mathcal{X}}$, i.e. the functions $f : \mathcal{X} \mapsto \mathbb{R}$ for the form:

$$f = \sum_{i=1}^{m} a_i K_{\mathbf{x}_i}$$

for some $m \in \mathbb{N}$ and $(a_1, \ldots, a_m) \in \mathbb{R}^m$.

Proof: p.d. $\implies$ kernel (2/5)

- For any $f, g \in \mathcal{H}_0$, given by:

$$f = \sum_{i=1}^{m} a_i K_{\mathbf{x}_i}, \quad g = \sum_{j=1}^{n} b_j K_{\mathbf{y}_j},$$

  let:

$$\langle f, g \rangle_{\mathcal{H}_0} := \sum_{i,j} a_i b_j K\left(\mathbf{x}_i, \mathbf{y}_j\right).$$

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

$$\langle f, g \rangle_{\mathcal{H}_0} = \sum_{i=1}^{m} a_i g\left(\mathbf{x}_i\right) = \sum_{j=1}^{n} b_j f\left(\mathbf{y}_j\right).$$

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

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

Proof: p.d. $\implies$ kernel (3/5)

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

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

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

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

  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. $\implies$ kernel (4/5)

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

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

  Therefore for any $\mathbf{x}$ the sequence $(f_n(\mathbf{x}))_{n\geq 0}$ 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 (up to a few technicalities, left as exercice). $\square$

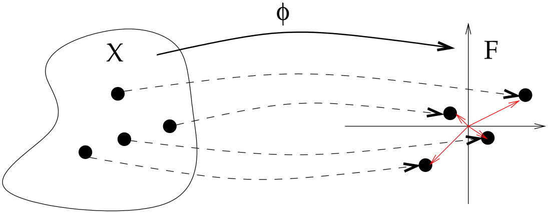
Proof: p.d. $\implies$ kernel (5/5)

- Let now the mapping $\Phi : \mathcal{X} \to \mathcal{H}$ defined by:

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

- By the reproducing property we have:

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



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:

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

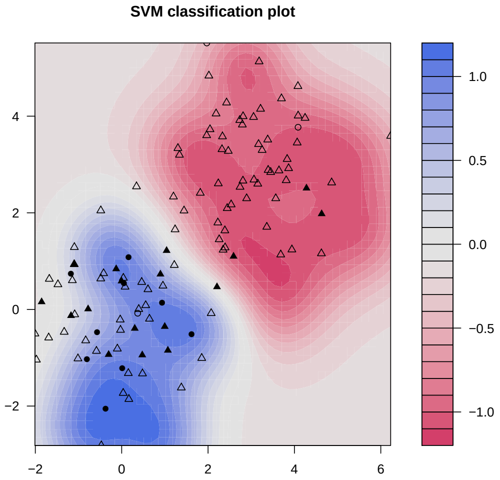
$$\text{s.t. } 0 \leq \alpha_i \leq C, \quad \text{and } \sum_{i=1}^{n} \alpha_i y_i = 0.$$

- 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

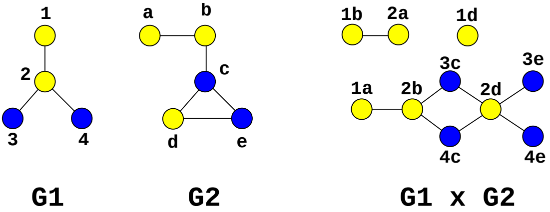
$$f(\vec{x}) = \sum_{i=1}^{n} \alpha_i \exp\left(-\frac{||\vec{x} - \vec{x}_i||^2}{2\sigma^2}\right)$$



How to choose or make a kernel?

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

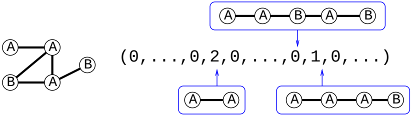
Example: design features (Gärtner et al., 2003)



$$K(G_1, G_2) = \mathbf{1}^\top A^n_{G_1 \times G_2} \mathbf{1}$$

Exercice

Show that the features are the counts of labeled walks of length n in the graph.



Example: adapt a similarity measure (Saigo et al., 2004)

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

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

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

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

Example: design a regularizer

- To any space $\mathcal{X}$ and positive definite kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that corresponds to an inner product

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

with $\Phi : \mathcal{X} \to \mathcal{H}$ is associated a class of functions

$$\mathcal{H}_K = \left\{ f_\beta(x) = \beta^\top \Phi(x) \ : \ \beta \in \mathcal{H} \right\}$$

and a regularizer

$$\Omega(f_\beta) = \|\beta\|^2$$

- In fact, $\mathcal{H}_K$ is itself a Hilbert space called the reproducing kernel Hilbert space (RKHS) of K.
- We can choose a kernel to define a particular class of function and a particular regularizer.

Example: Sobolev norm

- 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

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

Example: 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 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

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

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

Example

Laplace kernel

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

corresponds to:

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

and

$$\mathcal{H}=\left\{f:\int\left|\,\hat{f}(\omega)\,\right|^{2}\frac{\left(\gamma^{2}+\omega^{2}\right)}{\gamma}d\omega<\infty\right\},$$

the set of functions L^2 differentiable with derivatives in L^2 (Sobolev norm).

Example

Low-frequency filter

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

corresponds to:

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

and

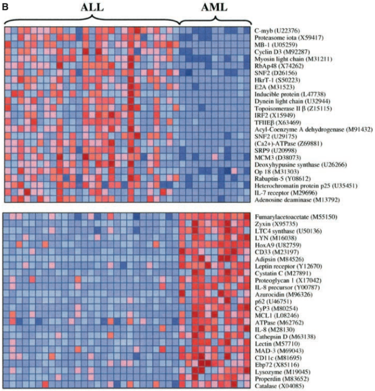
$$\mathcal{H} = \left\{ f : \int_{|\omega| > \Omega} \left| \hat{f}(\omega) \right|^2 d\omega = 0 \right\},$$

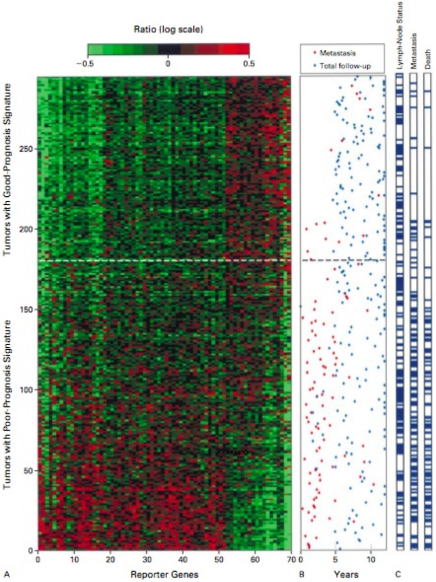
the set of functions whose spectrum is included in $[-\Omega, \Omega]$.

Outline

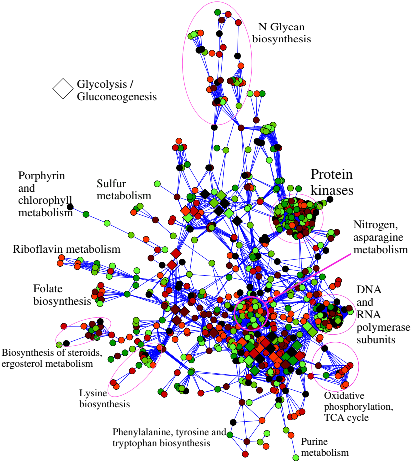
1. Penalized empirical risk minimization
2. Learning with ℓ_2 regularization
3. Kernel methods
4. Learning molecular classifiers with network information
5. Data integration with kernels

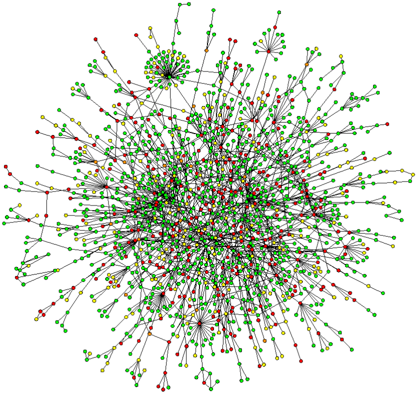
Molecular diagnosis / prognosis / theragnosis





Gene networks

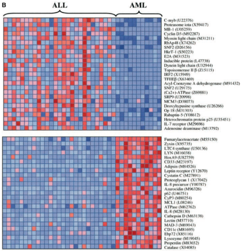


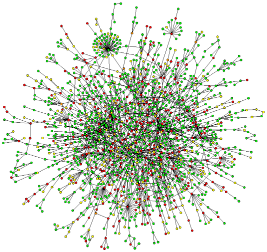


Gene networks and expression data

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)

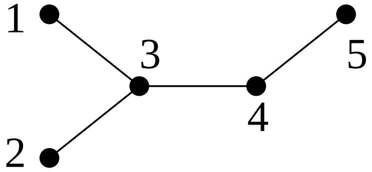
$$\Omega(\beta) = \sum_{i\sim j}(\beta_i - \beta_j)^2 \,,$$

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

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

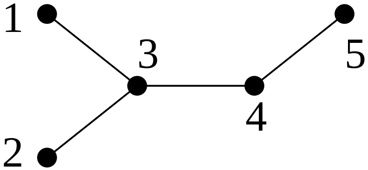
$$\min_{\gamma \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \ell\left(\gamma^\top \Phi(x_i), y_i\right) + \lambda \gamma^\top \gamma \,,$$

and where

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

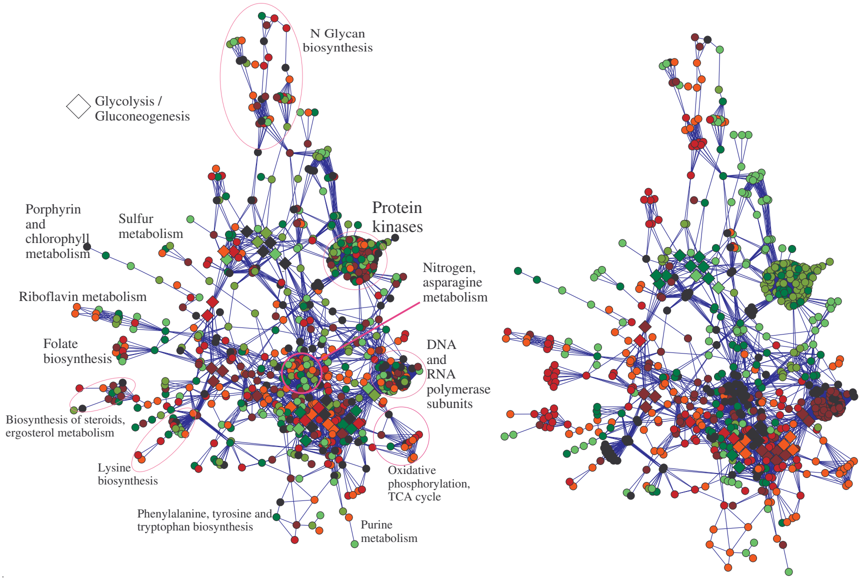
for $K_G = L^*$, the pseudo-inverse of the graph Laplacian.

Example

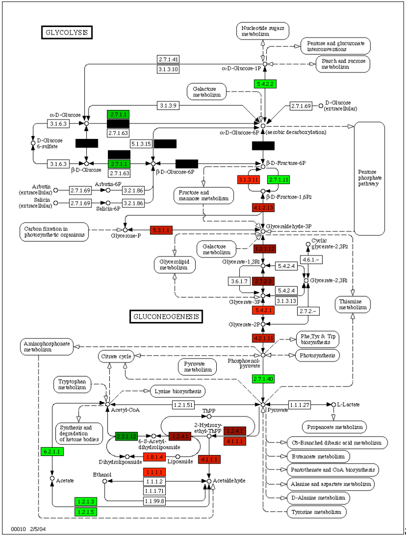


$$L^* = \begin{pmatrix} 0.88 & -0.12 & 0.08 & -0.32 & -0.52 \\ -0.12 & 0.88 & 0.08 & -0.32 & -0.52 \\ 0.08 & 0.08 & 0.28 & -0.12 & -0.32 \\ -0.32 & -0.32 & -0.12 & 0.48 & 0.28 \\ -0.52 & -0.52 & -0.32 & 0.28 & 1.08 \end{pmatrix}$$

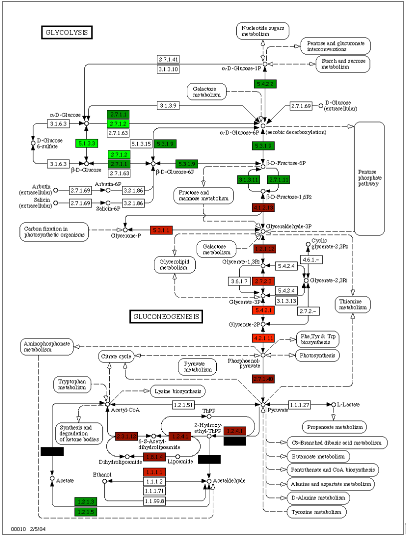
Classifiers



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:

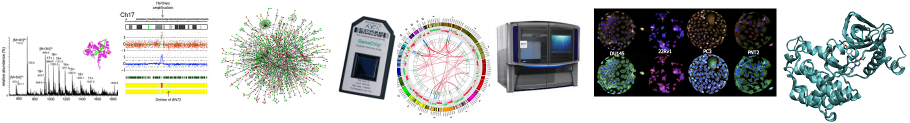
$$K_G = \exp_M(-2tL) .$$

penalizes high frequencies of β in the Fourier domain.

Outline

1. Penalized empirical risk minimization
2. Learning with ℓ_2 regularization
3. Kernel methods
4. Learning molecular classifiers with network information
5. Data integration with kernels

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 $K(x, x') = \Phi(x)^\top \Phi(x')$, we learn a function $f_\beta(x) = \beta^\top \Phi(x)$ by solving:

$$\min_\beta R(f_\beta^n) + \lambda \|\beta\|^2 ,$$

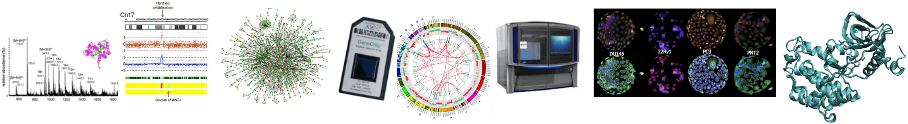
  where $f^n = (f_\beta(x_1), \ldots, f_\beta(x_n)) \in \mathbb{R}^n$
- By the representer theorem, we know that the solution is

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

  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 $i = 1, \dots, M$, 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 $\Phi_S : \mathcal{X} \to \mathcal{H}_S = \mathcal{H}_1 \oplus \dots \oplus \mathcal{H}_M$ is the concatenation of the feature maps Φ_i:

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

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

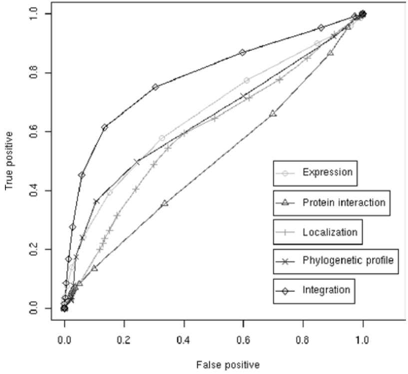
BIOINFORMATICS

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:

$$f^* = \sum_{i=1}^{M} f_i^* = \sum_{i=1}^{M} \Phi_i(x)^\top \beta_i^* \,,$$

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

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

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:

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

Proof (1/4)

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

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

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

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

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

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

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

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

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

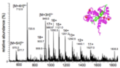
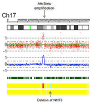
- If γ^* is a solution of the dual problem, then $\alpha_i^* = \eta_i\gamma^*$ leading to:

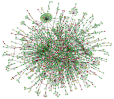
$$\forall x\in\mathcal{X}\,,\quad f_i^*(x)=\sum_{j=1}^n \alpha_{ij}^* K_i(x_j,x)=\sum_{j=1}^n \eta_i\gamma_j^* K_i(x_j,x)$$

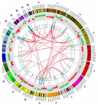
- Therefore, $f^* = \sum_{i=1}^M f_i^*$ satisfies

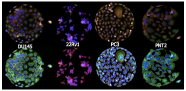
$$f^*(x)=\sum_{i=1}^M\sum_{j=1}^n \eta_i\gamma_j^* K_i(x_j,x)=\sum_{j=1}^n \gamma_j^* K_\eta(x_j,x)\,.\quad\square$$

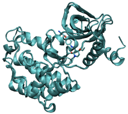
Learning the kernel









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

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

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:

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

- 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

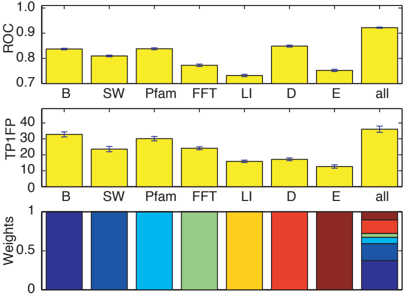
Vol. 20 no. 16 2004, pages 2626–2635
doi:10.1093/bioinformatics/bth294

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

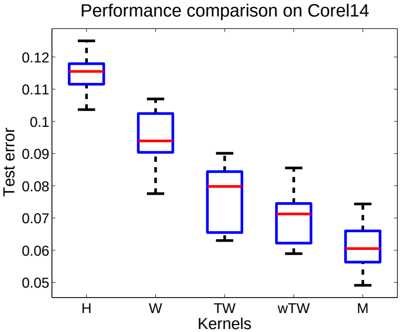


(B) Membrane proteins

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

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

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:

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

Proof (1/2)

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

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

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

- Learning with MKL (best convex combination) solves

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

- 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:

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

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:

$$\forall i = 1, \ldots, d\,, \quad K_i\left(x, x'\right) = x_i x_i'\,.$$

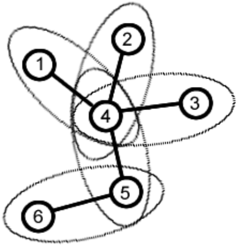
- The sum kernel is $K_S\left(x, x'\right) = \sum_{i=1}^d x_i x_i' = x^\top x$
- Learning with the sum kernel solves a ridge regression problem:

$$\min_{\beta \in \mathbb{R}^d} \left\{ R(X\beta) + \lambda \sum_{i=1}^d \beta_i^2 \right\}\,.$$

- Learning with MKL solves a LASSO regression problem:

$$\min_{\beta \in \mathbb{R}^d} \left\{ R(X\beta) + \lambda \left(\sum_{i=1}^d |\, \beta_i \,| \right)^2 \right\}\,.$$

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

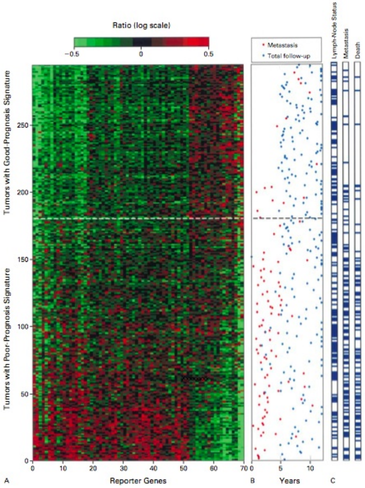


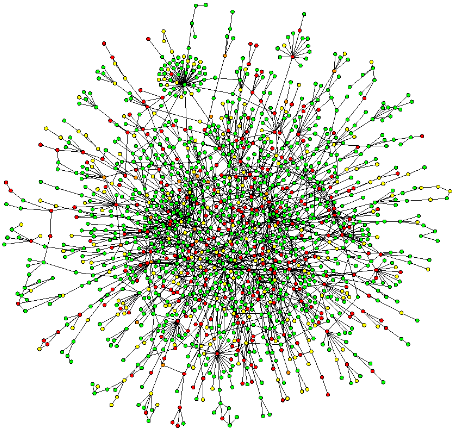
- Graph $G = (V, E)$, $\mathcal{X} = \mathbb{R}^V$
- For each edge $e = (i, j)$, define the kernel

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

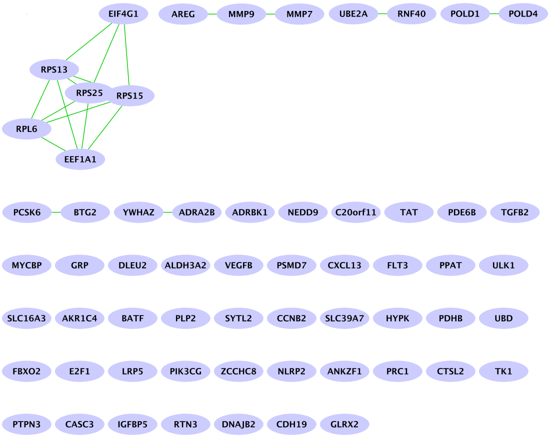
- MKL (aka latent group lasso) with the set $\{K_e : e \in E\}$ leads to a sparse linear model with connected non-zero components.

Application: breast cancer prognosis

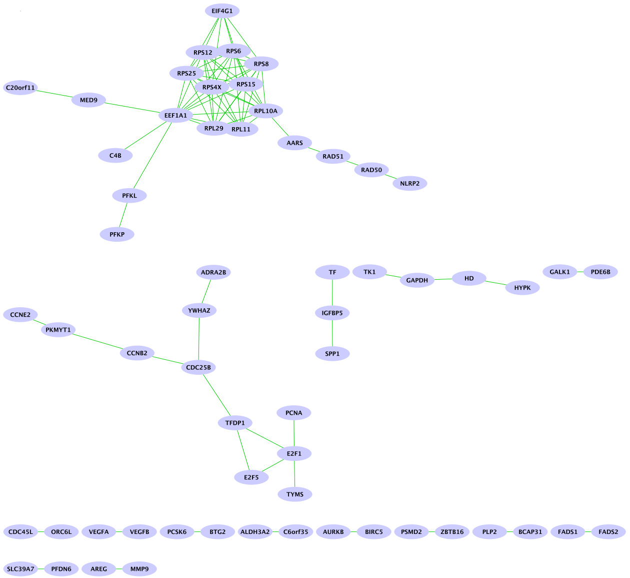




Lasso signature (accuracy 0.61



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.

T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: hardness results and efficient alternatives. In B. Schölkopf and M. Warmuth, editors, *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory and the Seventh Annual Workshop on Kernel Machines*, volume 2777 of *Lecture Notes in Computer Science*, pages 129–143, Heidelberg, 2003. Springer. doi: 10.1007/b12006. URL http://dx.doi.org/10.1007/b12006.

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.

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

References (cont.)

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`.

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`.

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`.

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`.