Machine Learning in Computational Biology (the frequentist approach)

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

Mines ParisTech / Institut Curie / Inserm

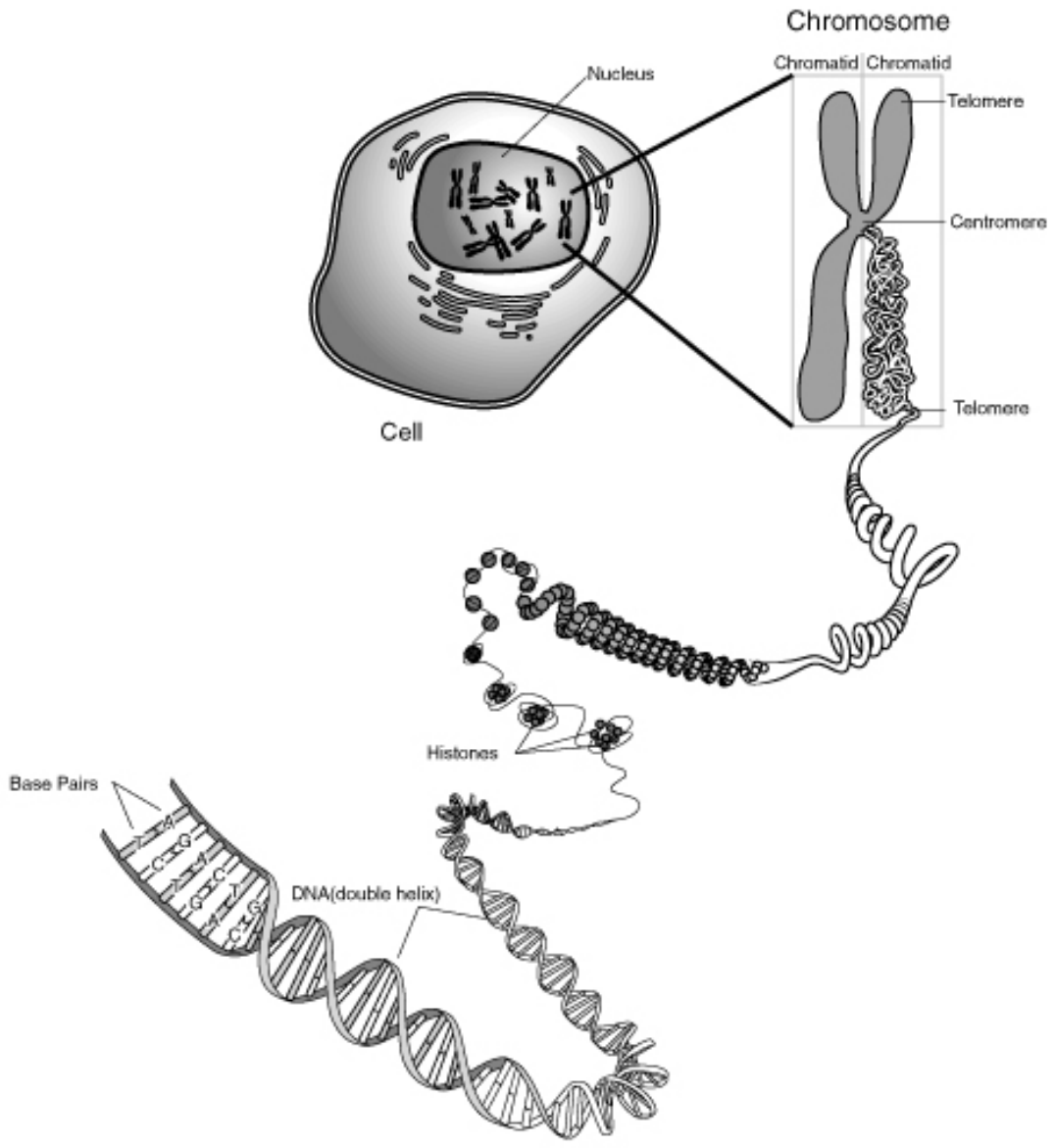
MLPM, Tübingen, September 2013.

Outline

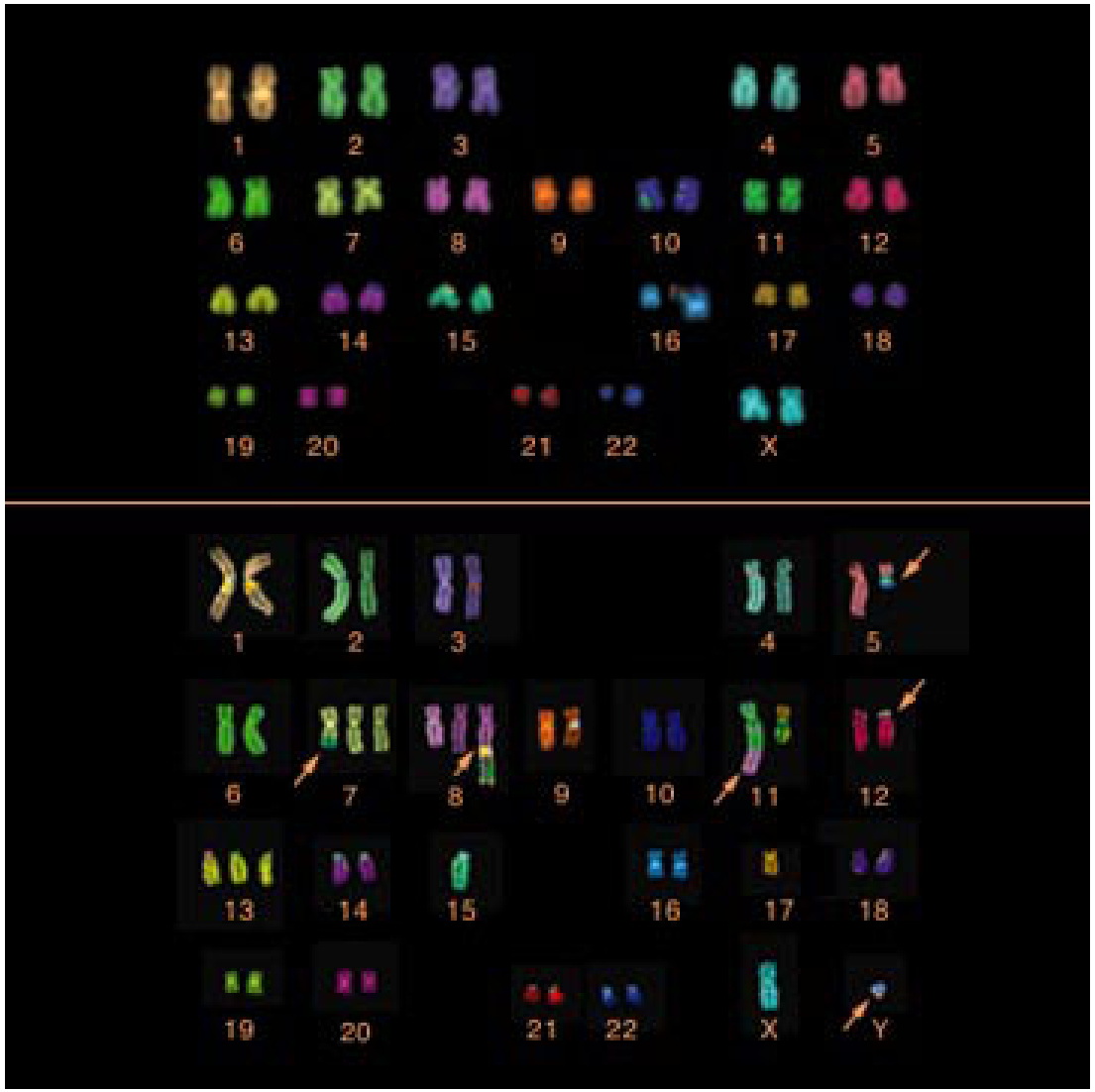
Outline

Outline

Cells, chromosomes, DNA



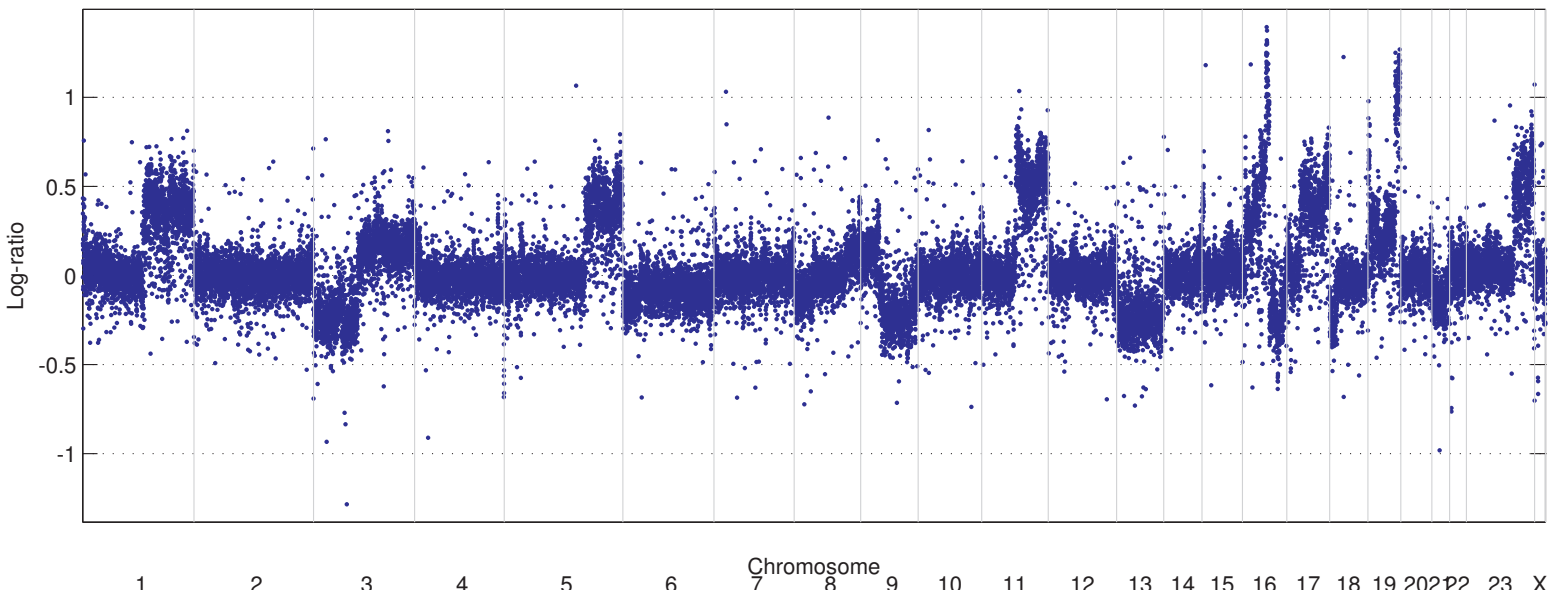
Chromosomic aberrations in cancer cells



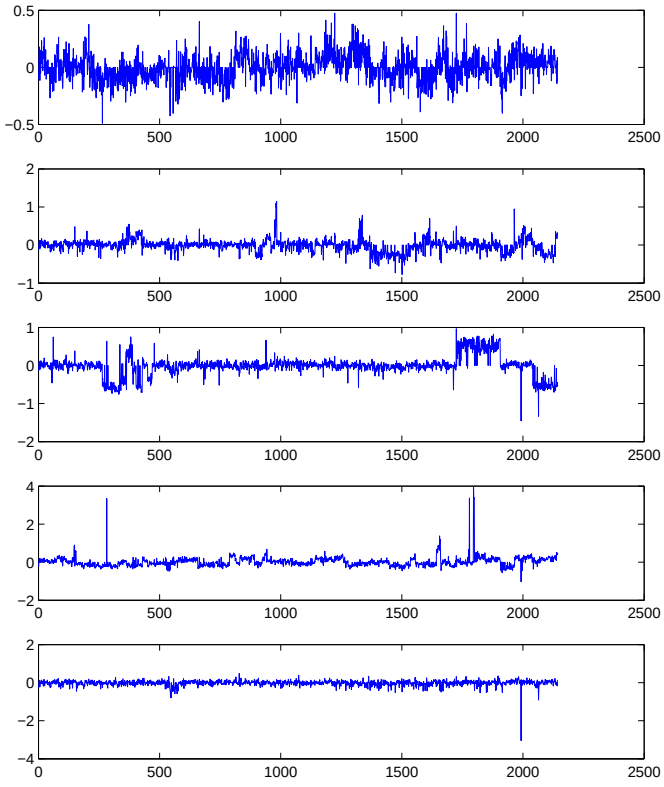
Comparative Genomic Hybridization (CGH)

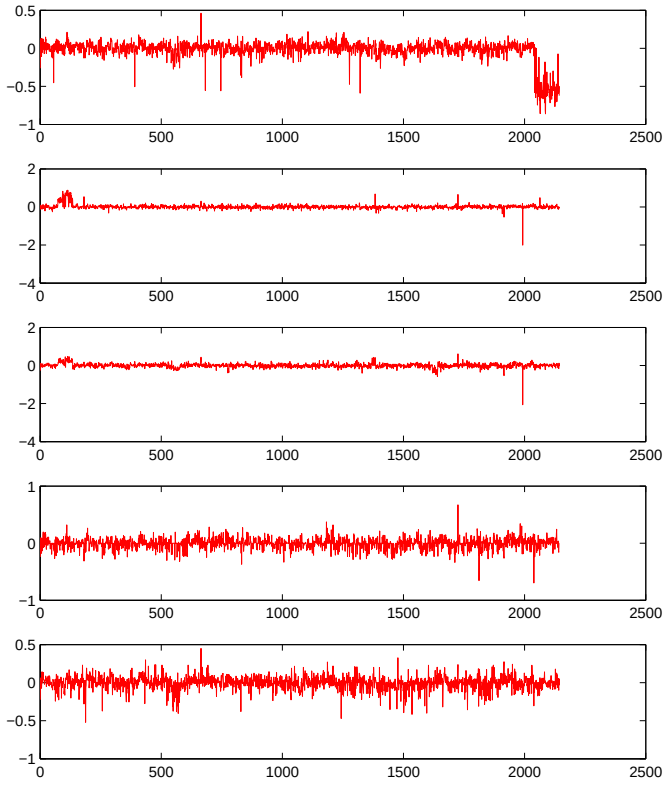
Motivation

- Comparative genomic hybridization (CGH) data measure the DNA copy number along the genome
- Very useful, in particular in cancer research to observe systematically variants in DNA content



Cancer prognosis: can we predict the future evolution?

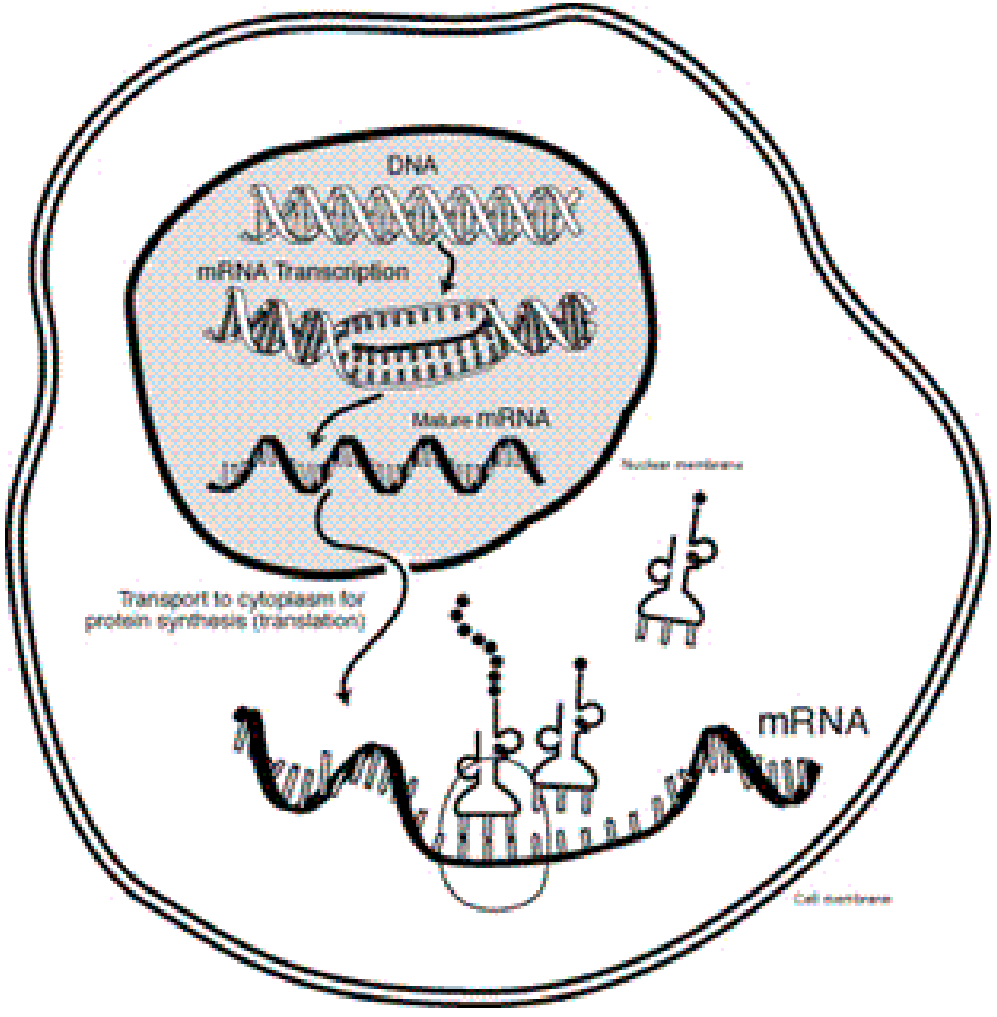


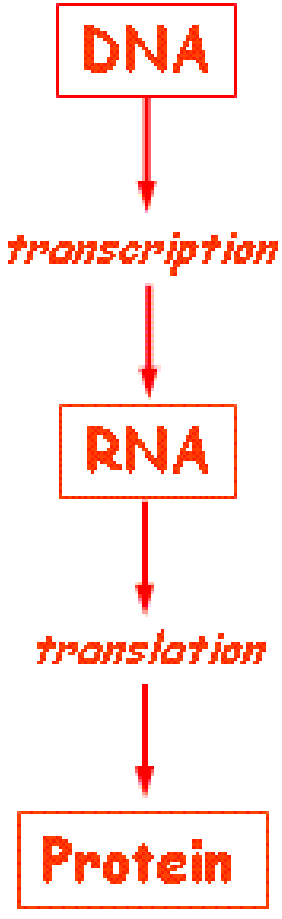


Problem 1

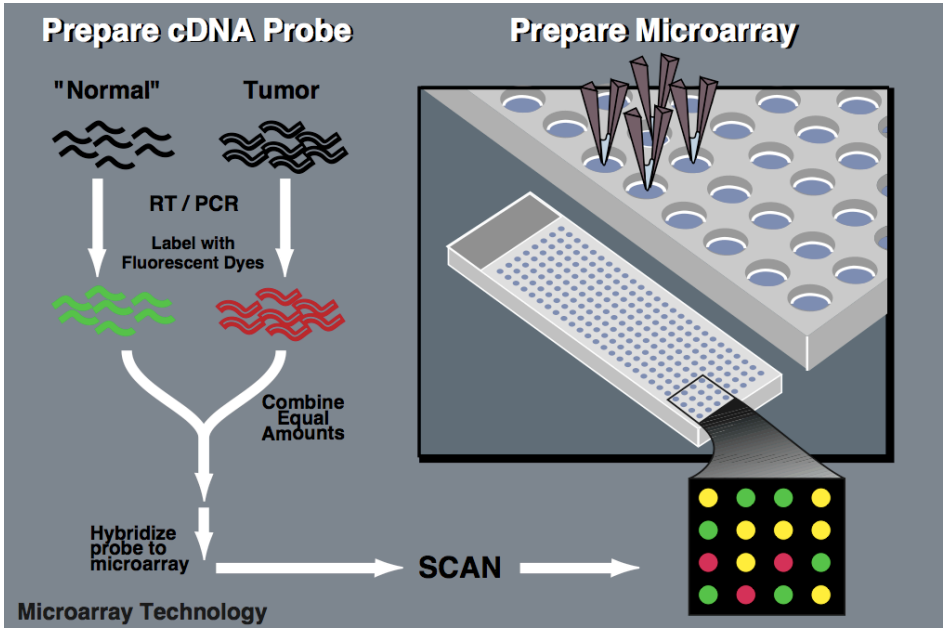
From a CGH profile, can we predict whether a melanoma will relapse (left) or not (right)?

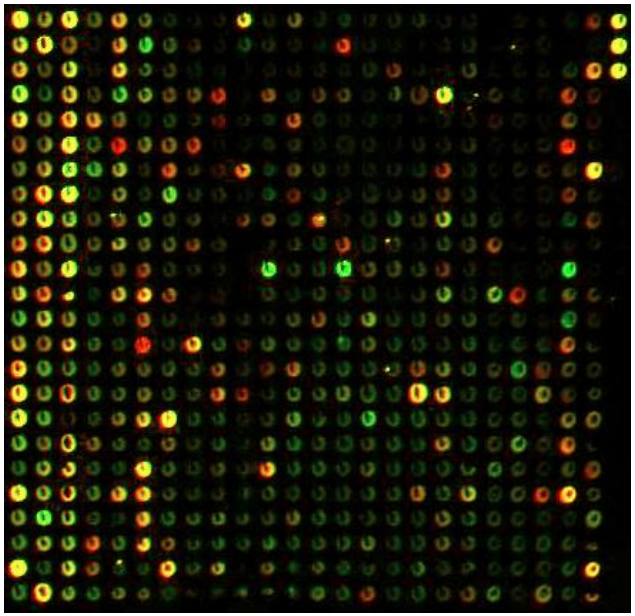
DNA → RNA → protein

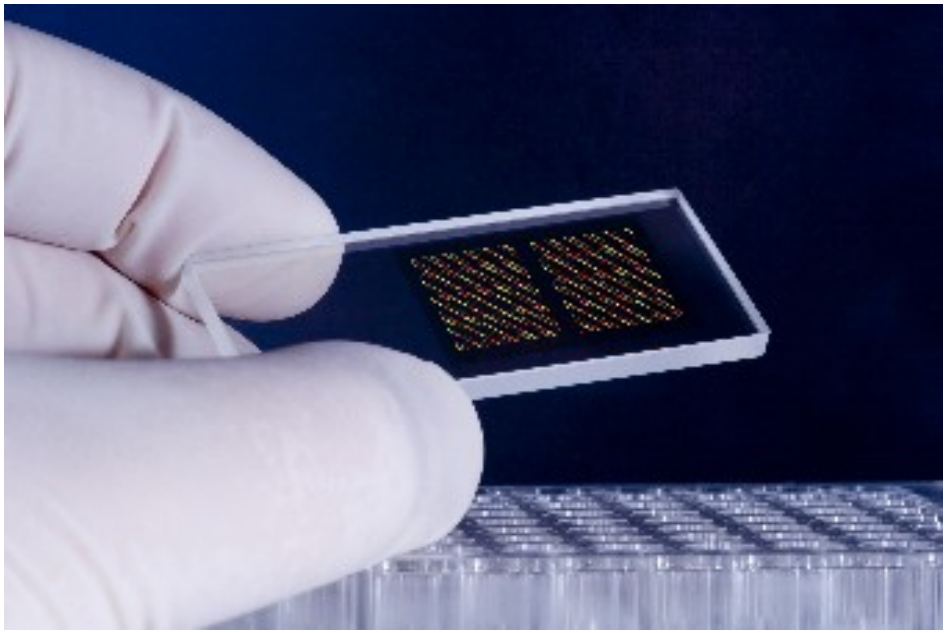




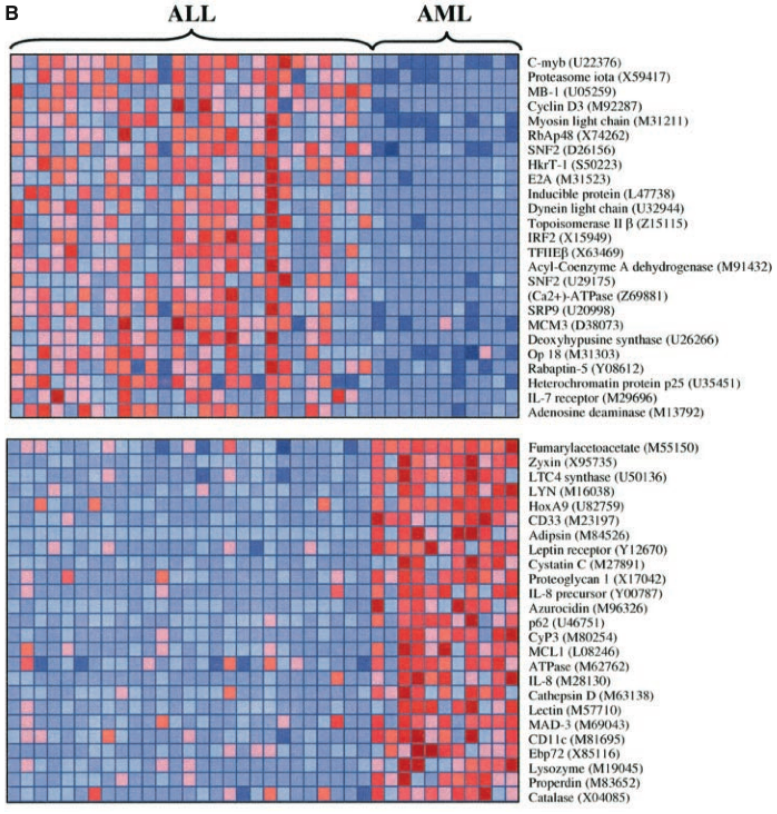
Tissue profiling with DNA chips







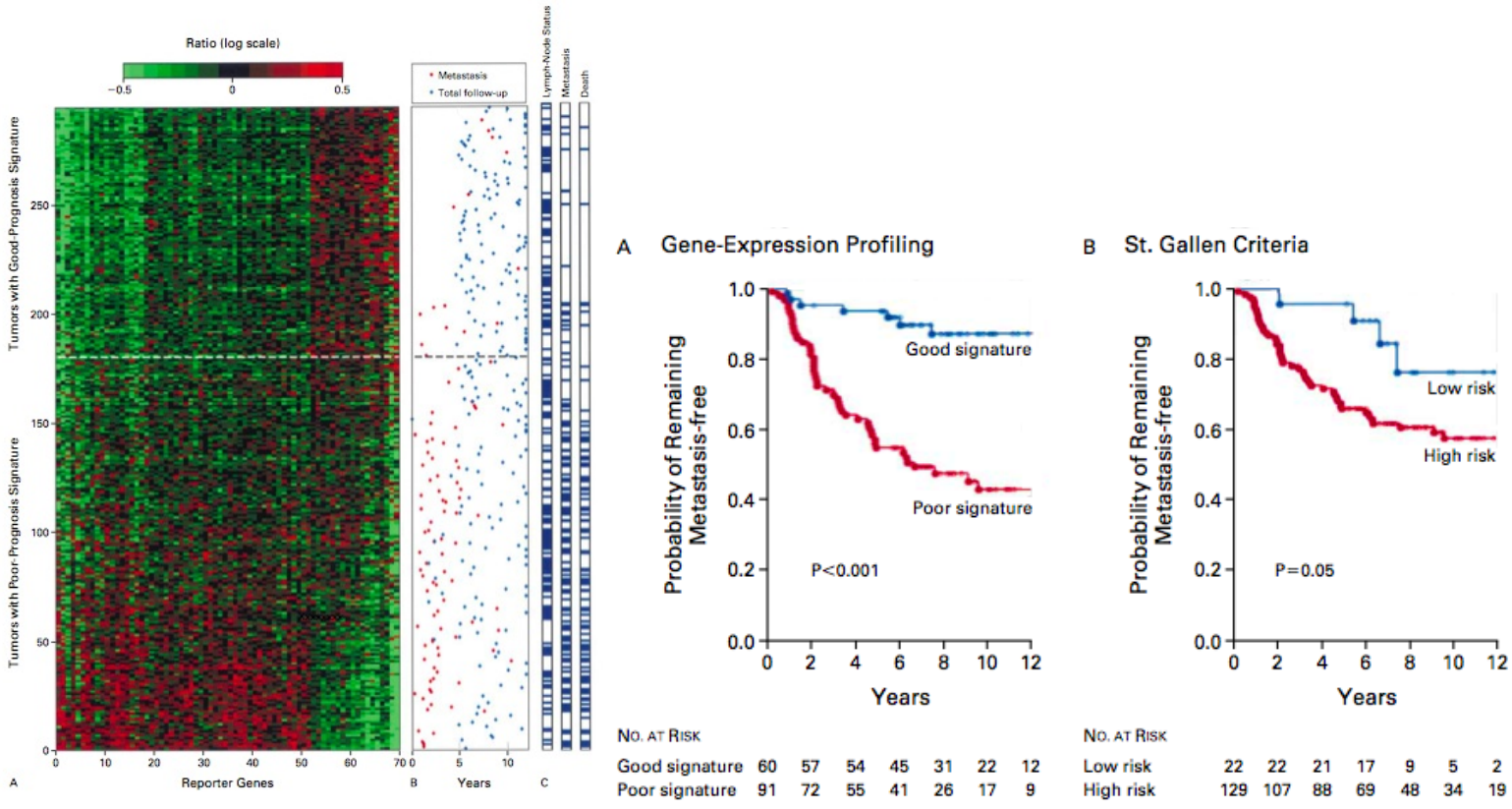
Use in diagnosis



Problem 2

Given the expression profile of a leukemia, is it an acute lymphocytic or myeloid leukemia (ALL or AML)?

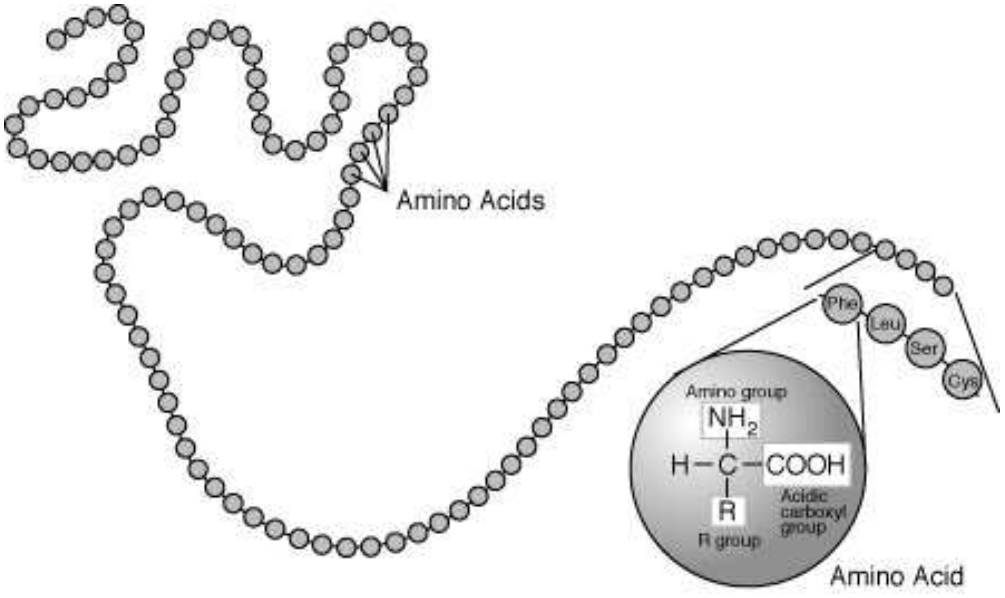
Use in prognosis



Problem 3

Given the expression profile of a breast cancer, is the risk of relapse within 5 years high?

Proteins



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

Protein annotation

Data available

- Secreted proteins:

  MASKATLLLAFTLLFATCIARHQQRQQQQNQCQLQNIEA...

  MARSSLFTFLCLAVFINGCLSQIEQQSPWEFQGSEVW...

  MALHTVLIMLSLLPMLEAQNPEHANITIGEPITNETLGWL...

  ...

- Non-secreted proteins:

  MAPPSVFAEVPQAQPVLVFKLIADFREDPDPRKVNLGVG...

  MAHTLGLTQPNSTEPHKISFTAKEIDVIEWKGDILVVG...

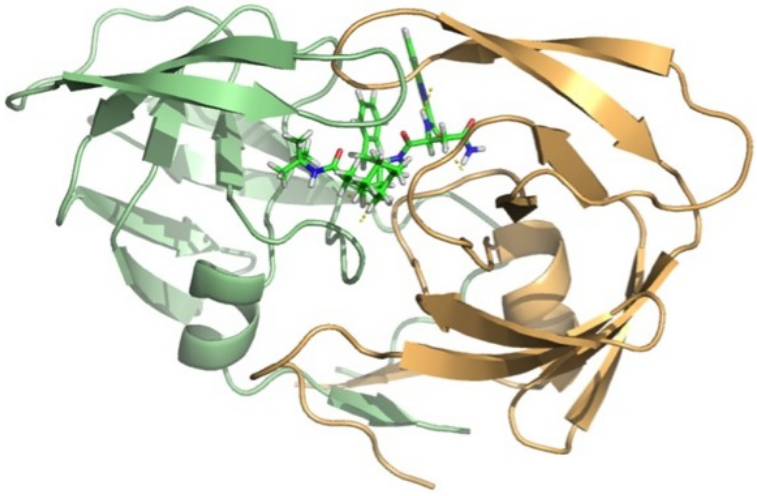
  MSISESYAKEIKTAFRQFTDFPIEGEQFEDFLPIIGNP..

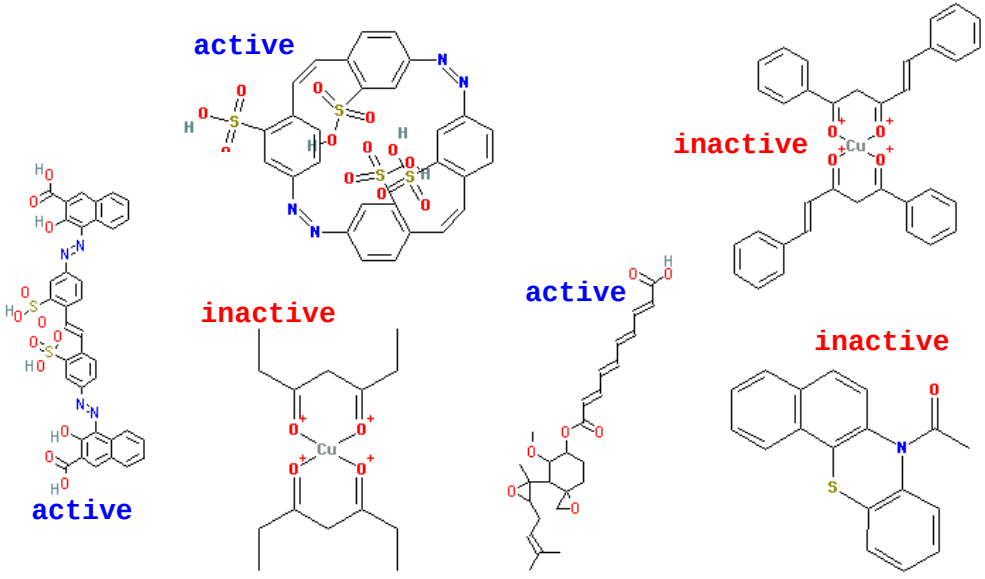
  ...

Problem 4

Given a newly sequenced protein, is it secreted or not?

Drug discovery

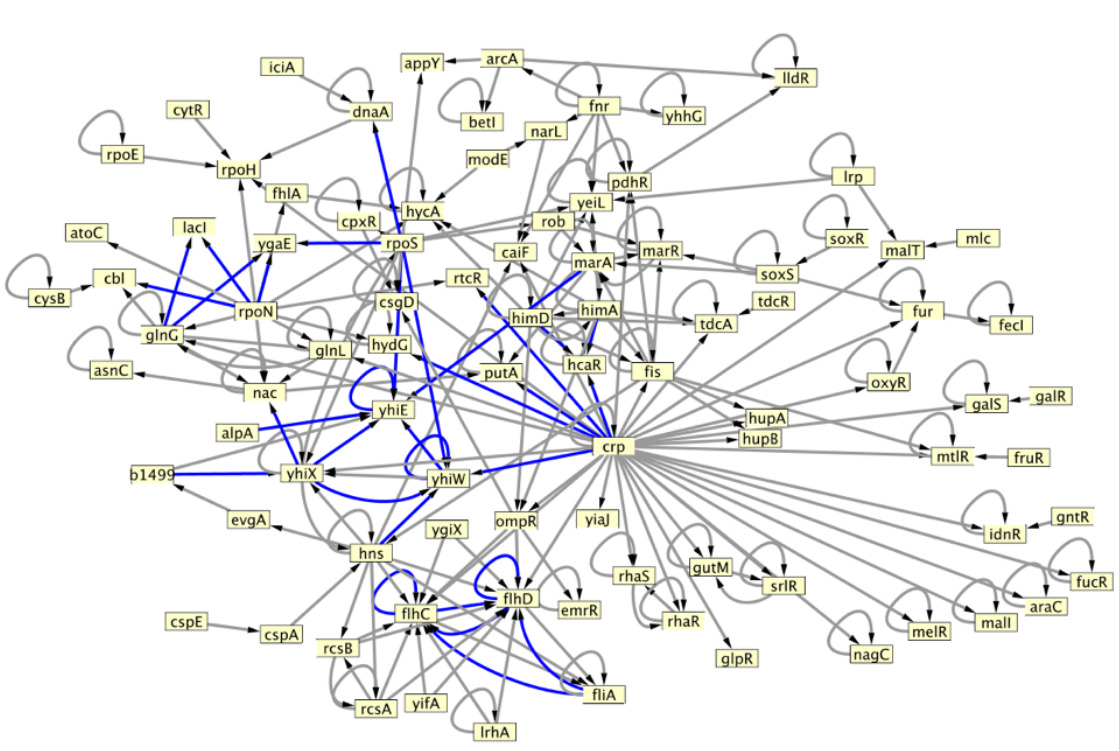


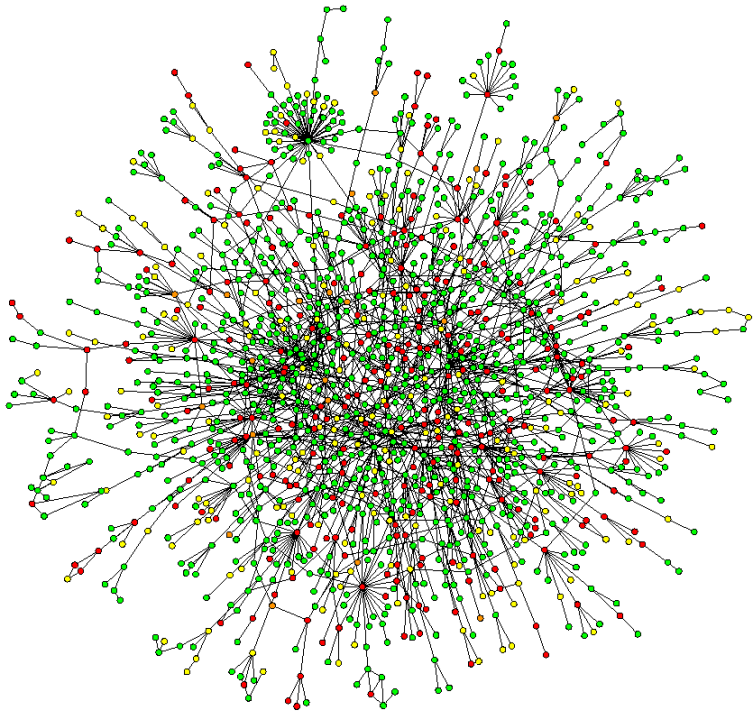


Problem 5

Given a new candidate molecule, is it likely to be active?

Gene network inference

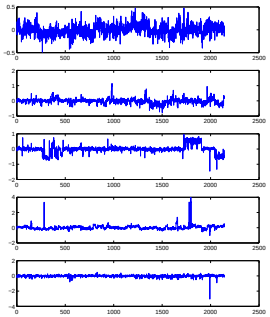


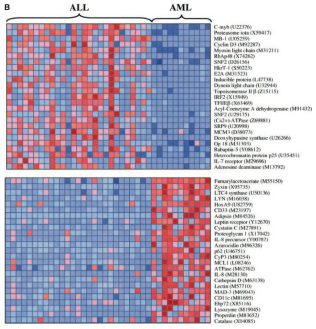


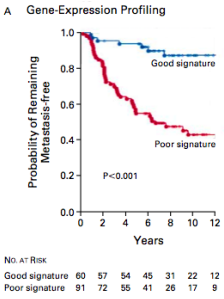
Problem 6

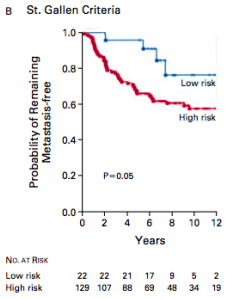
Given known interactions, can we infer new ones?

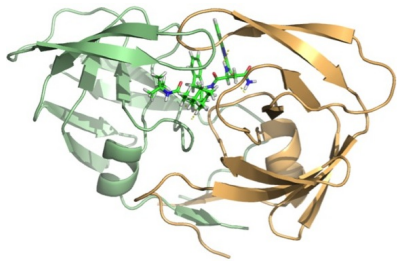
A common topic...

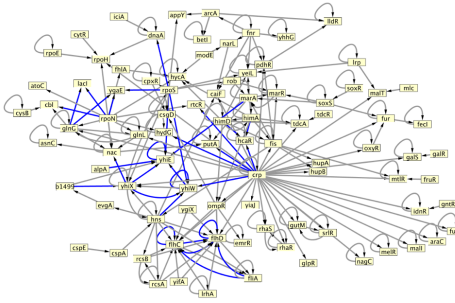


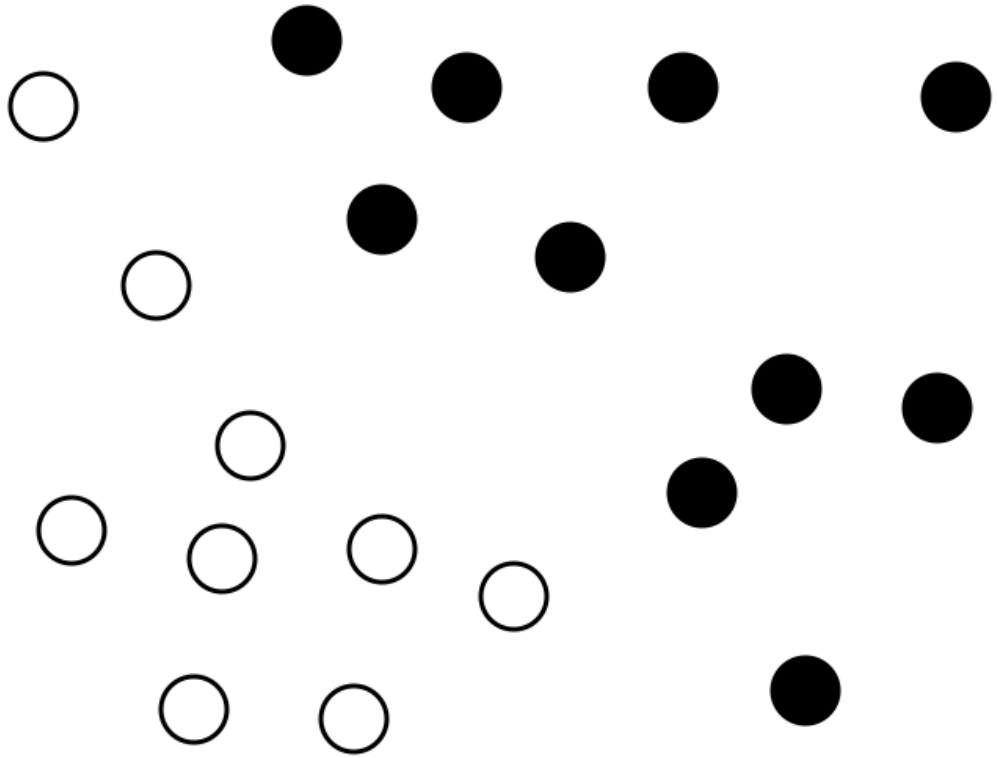




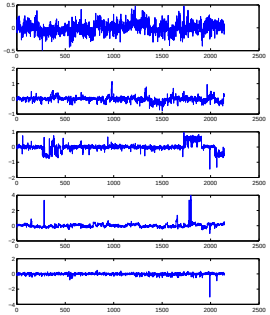


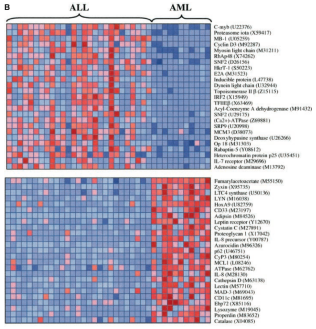


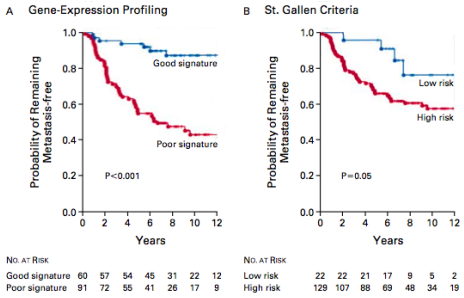


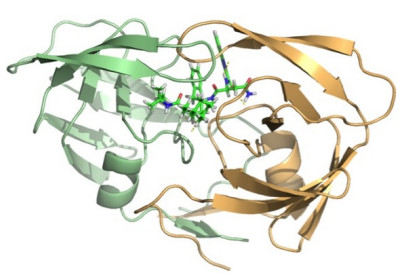


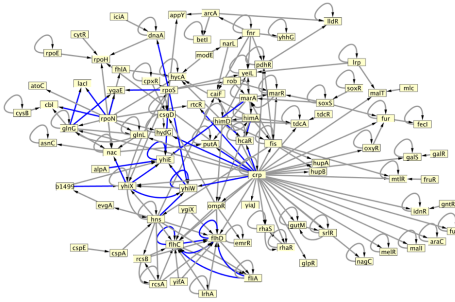
A common topic...

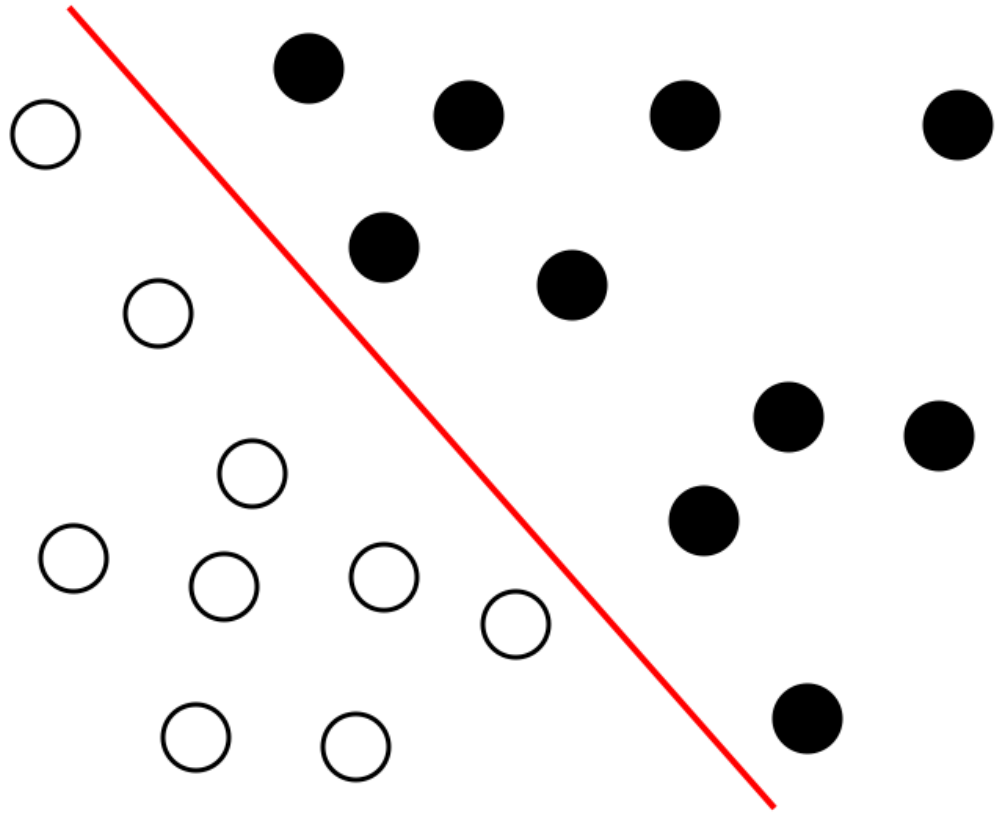




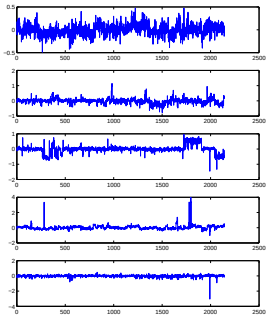


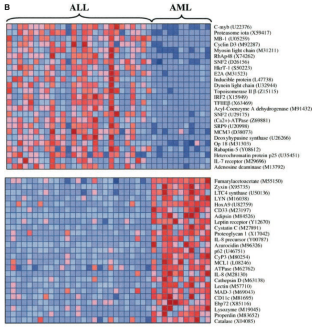


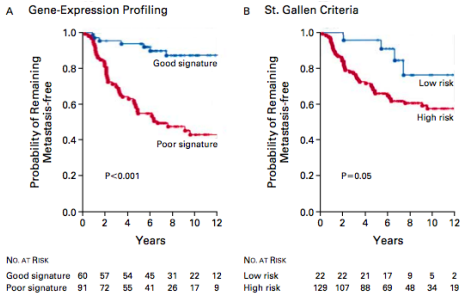


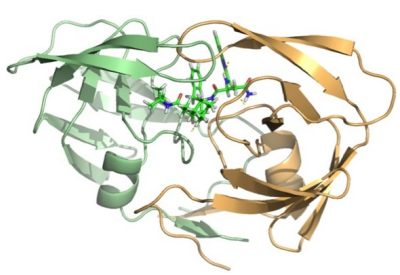


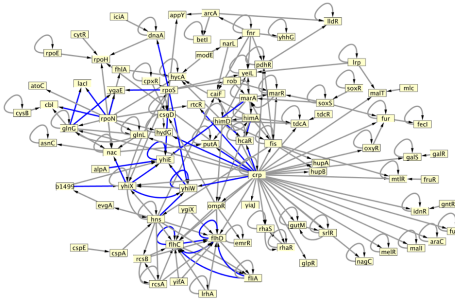
A common topic...

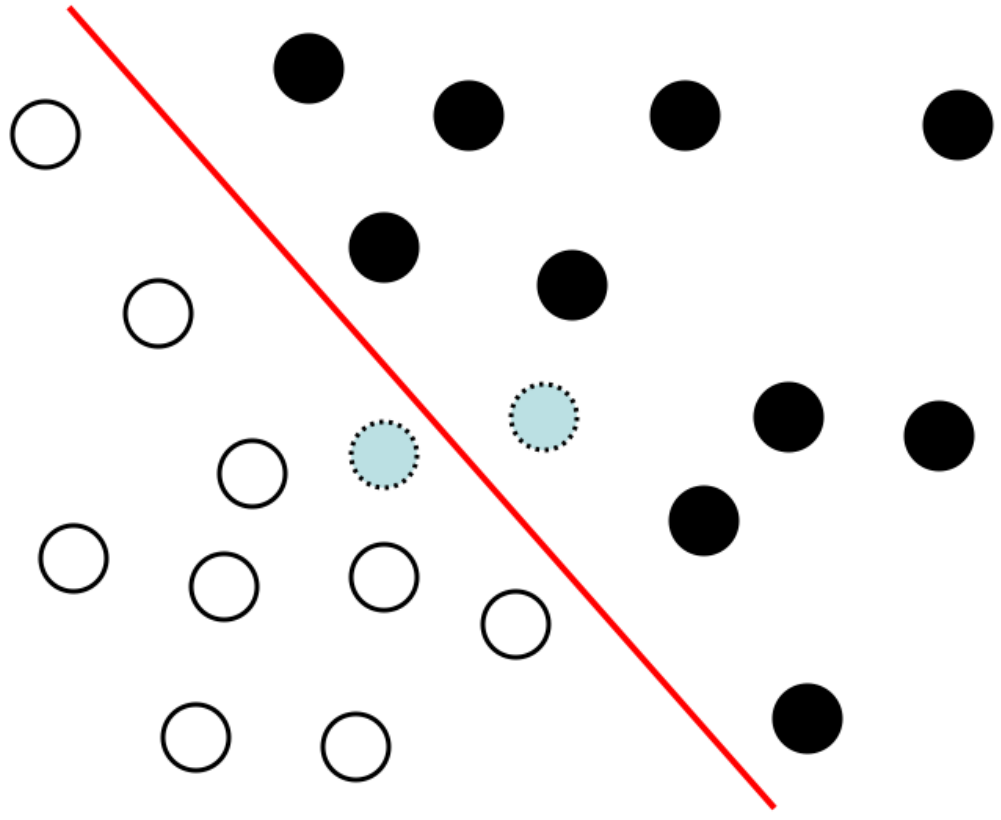




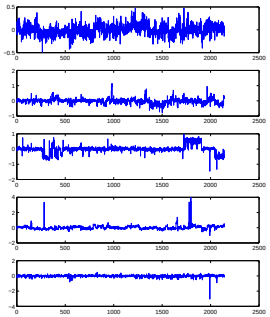


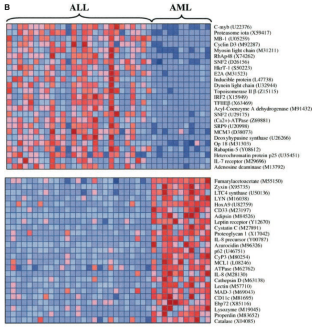


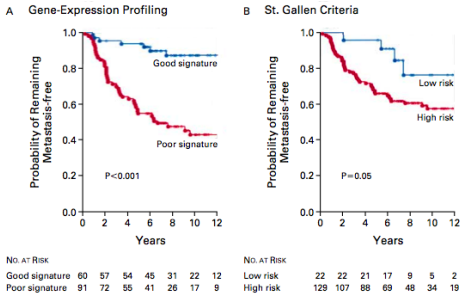


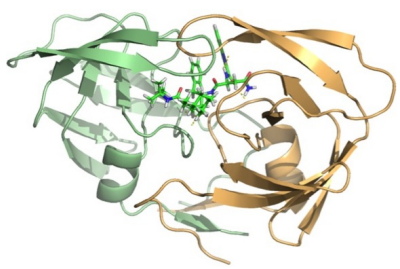


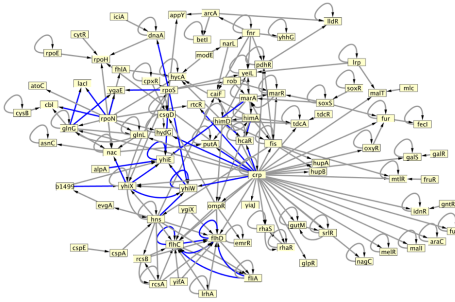
A common topic...

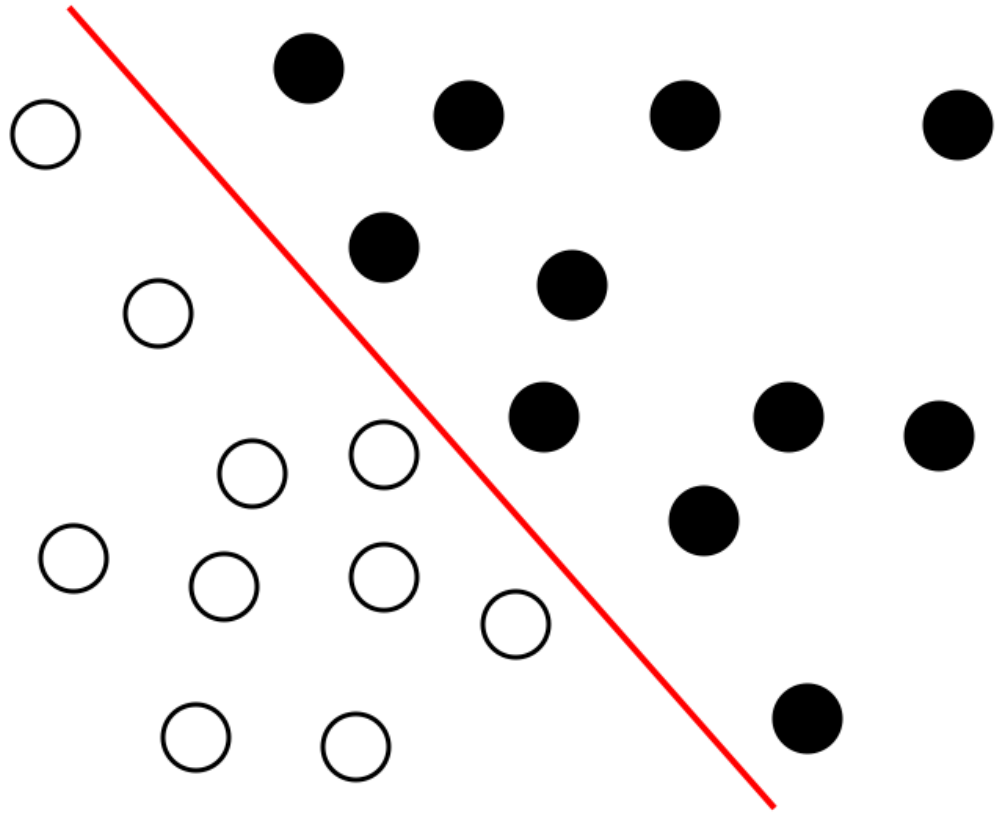




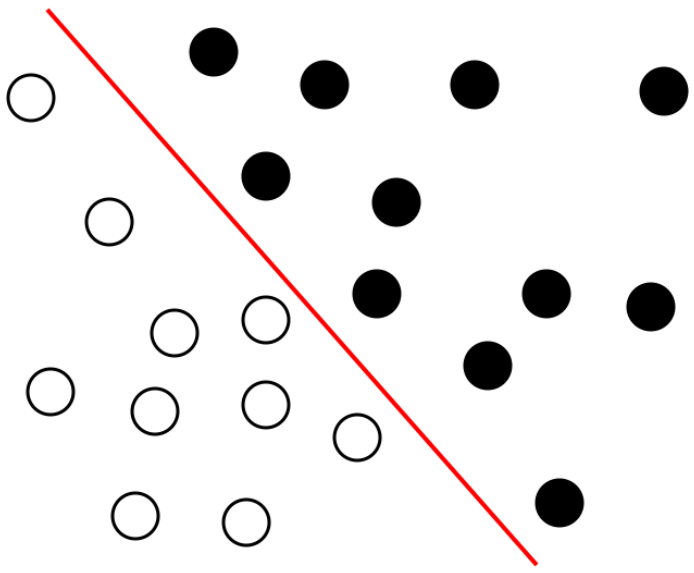








Pattern recognition, *aka* supervised classification

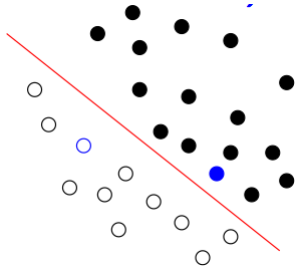


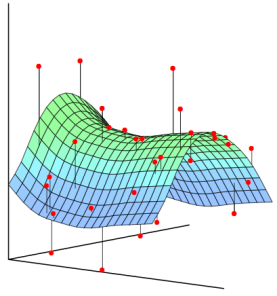
Challenges

- High dimension
- Few samples
- Structured data
- Heterogeneous data
- Prior knowledge
- Fast and scalable implementations
- Interpretable models

Outline

More formally





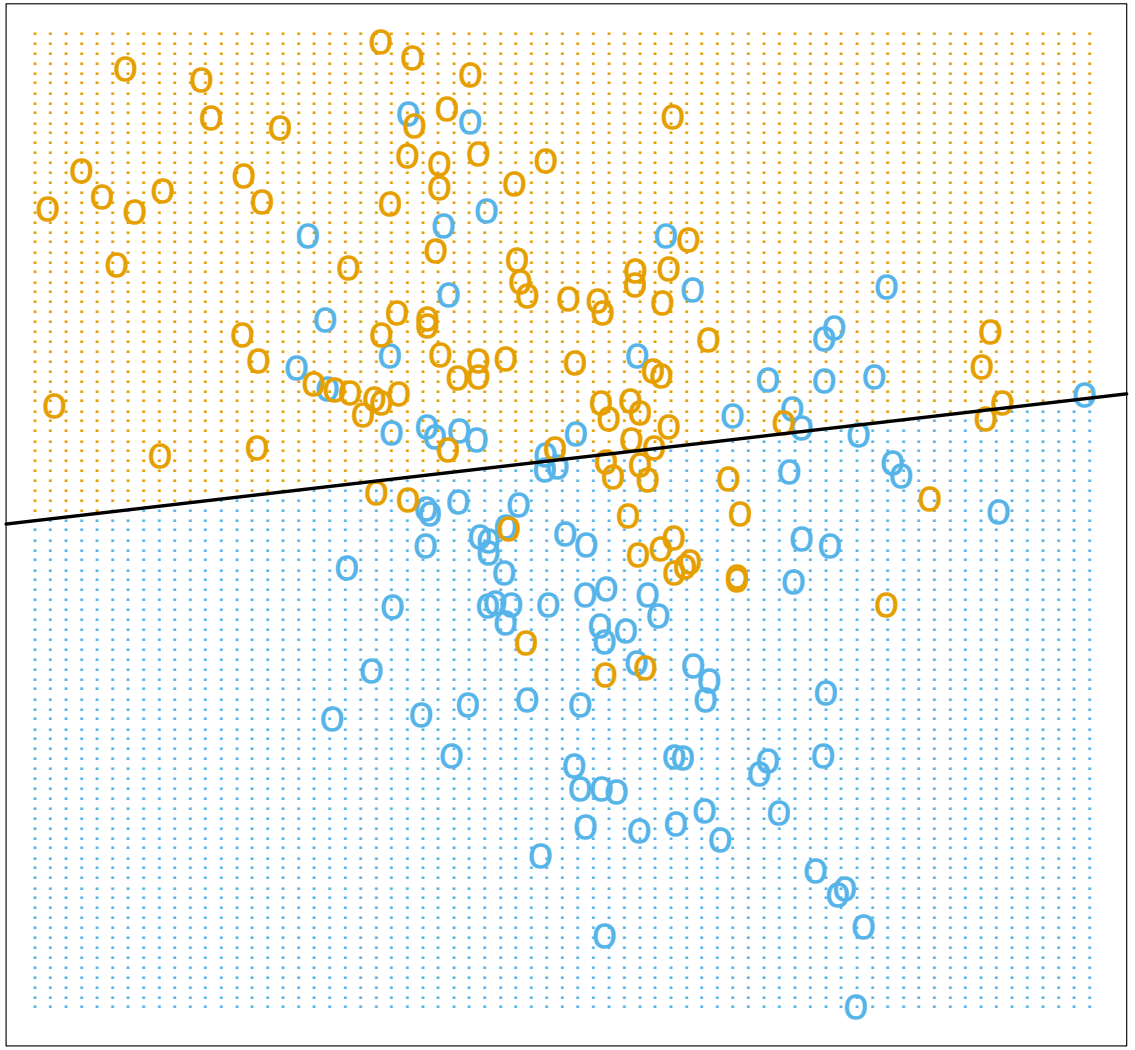
Input

- $\mathcal{X}$ the space of patterns (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

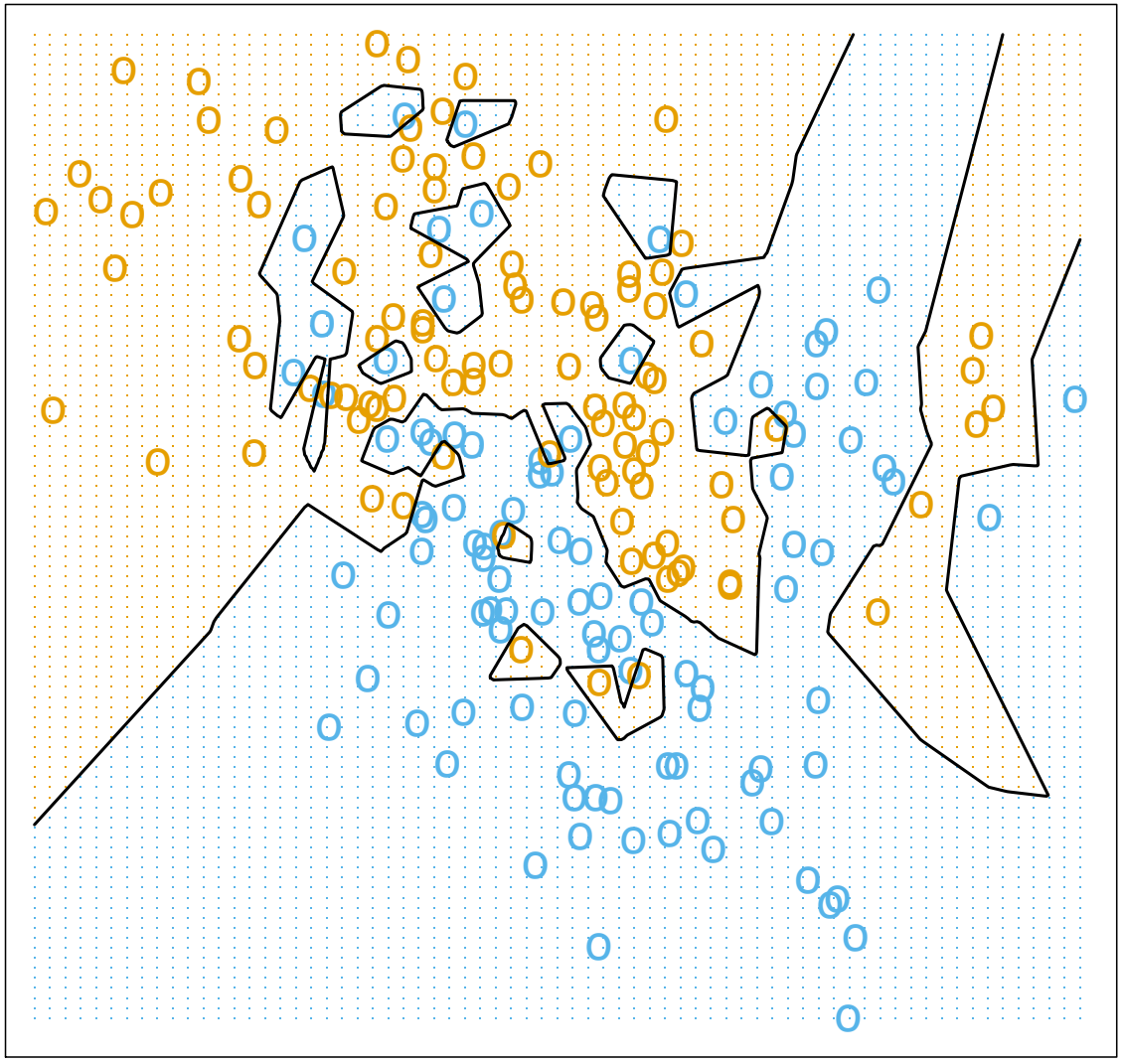
- A function $f : \mathcal{X} \to \mathcal{Y}$ to predict the output associated to any new pattern $x \in \mathcal{X}$ by $f(x)$

Simple example 1 : ordinary least squares (OLS)

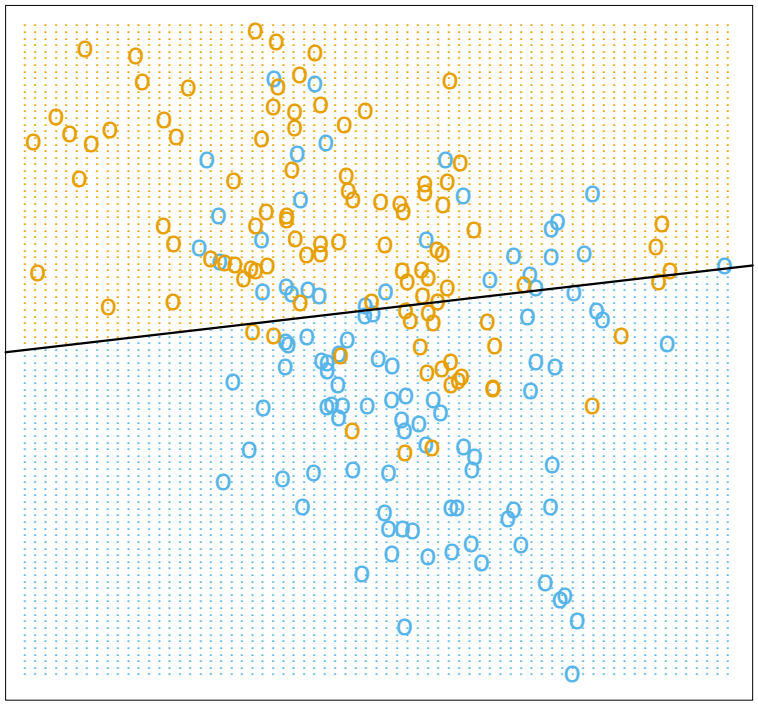


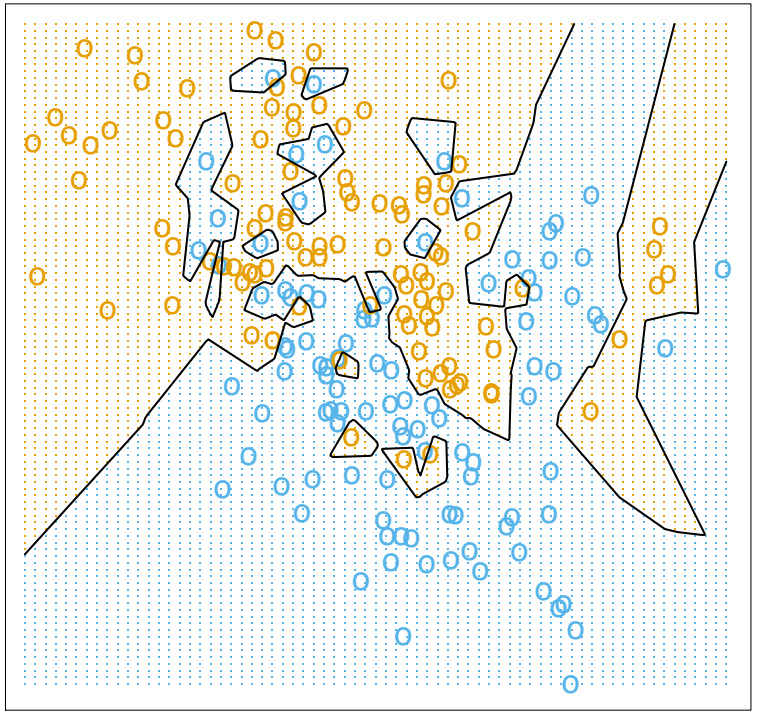
(Hastie et al. The elements of statistical learning. Springer, 2001.)

Simple example 1 : 1-nearest neighbor (1-NN)



What's wrong?





- OLS: the linear separation is not appropriate = "large bias"
- 1-NN: the classifier seems too unstable = "large variance"

The fundamental "bias-variance" trade-off

- Assume $Y = f(X) + \epsilon$, where ϵ is some noise
- From the training set $\mathcal{S}$ we estimate the predictor $\hat{f}$
- On a new point x_0, we predict $\hat{f}(x_0)$ but the "true" observation will be $Y_0 = f(x_0) + \epsilon$
- On average, we make an error of:

$$
\begin{aligned}
& E_{\epsilon,\mathcal{S}} \left(Y_0 - \hat{f}(x_0) \right)^2 \\
&= E_{\epsilon,\mathcal{S}} \left(f(x_0) + \epsilon - \hat{f}(x_0) \right)^2 \\
&= E\epsilon^2 + E_{\mathcal{S}} \left(f(x_0) - \hat{f}(x_0) \right)^2 \\
&= E\epsilon^2 + \left(f(x_0) - E_{\mathcal{S}}\hat{f}(x_0) \right)^2 + E_{\mathcal{S}} \left(\hat{f}(x_0) - E_{\mathcal{S}}\hat{f}(x_0) \right)^2 \\
&= \textit{noise} + \textit{bias}^2 + \textit{variance}
\end{aligned}
$$

Back to OLS

- Parametric model for $\beta \in \mathbb{R}^{p+1}$:

$$f_\beta(X) = \beta_0 + \sum_{i=1}^{p} \beta_i X_i = X^\top \beta$$

- Estimate $\hat{\beta}$ from training data to minimize

$$RSS(\beta) = \sum_{i=1}^{n} (y_i - f_\beta(x_i))^2$$

- Solution if $\mathbf{X}^\top \mathbf{X}$ is non-singular:

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

Optimality of OLS

Gauss-Markov theorem

- Assume $\mathbf{Y} = \mathbf{X}\beta + \epsilon$, where $E\epsilon = 0$ and $E\epsilon\epsilon^\top = \sigma^2 I$.
- Then the least squares estimator $\hat{\beta}$ is BLUE (best linear unbiased estimator), i.e., for any other estimator $\tilde{\beta} = C\mathbf{Y}$ with $E\tilde{\beta} = \beta$,

$$Var(\hat{\beta}) \leq Var(\tilde{\beta})$$

Nevertheless, if variance may be very large, we may have smaller total risk by increasing bias to decrease variance

Optimality of OLS

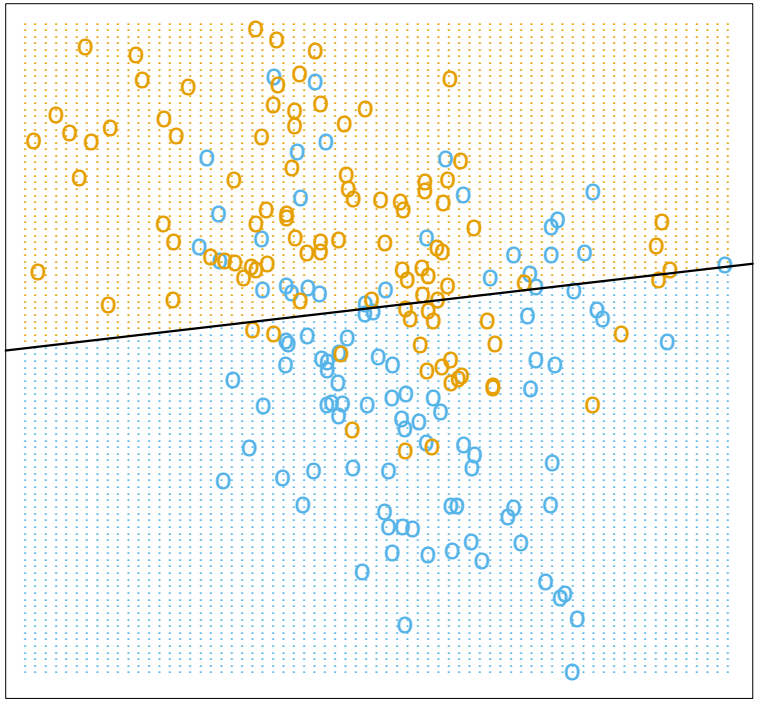
Gauss-Markov theorem

- Assume $\mathbf{Y} = \mathbf{X}\beta + \epsilon$, where $E\epsilon = 0$ and $E\epsilon\epsilon^\top = \sigma^2 I$.
- Then the least squares estimator $\hat{\beta}$ is BLUE (best linear unbiased estimator), i.e., for any other estimator $\tilde{\beta} = C\mathbf{Y}$ with $E\tilde{\beta} = \beta$,

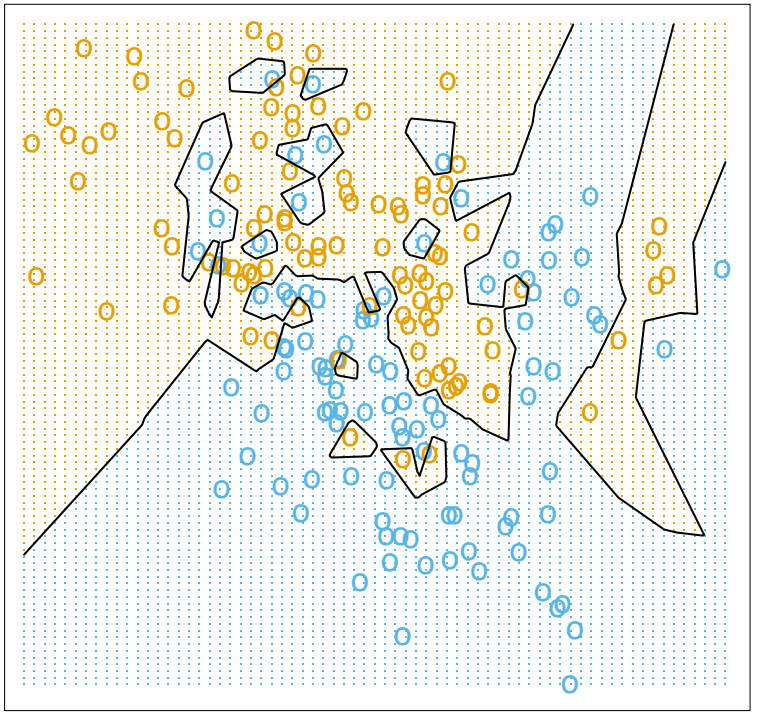
$$Var(\hat{\beta}) \leq Var(\tilde{\beta})$$

Nevertheless, if variance may be very large, we may have smaller total risk by increasing bias to decrease variance

The curse of dimensionality



Small dimension



Large dimension

In high dimensions, variance dominates. BLUE estimators are useless.

A solution: shrinkage estimators

1. Define a large family of "candidate classifiers", e.g., linear predictors:

$$f_\beta(x) = \beta^\top x \quad \text{for } x \in \mathbb{R}^p$$

2. For any candidate classifier f_β, quantify how "good" it is on the training set with some empirical risk, e.g.:

$$R(\beta) = \frac{1}{n}\sum_{i=1}^{n}(f_\beta(x_i) - y_i)^2 \,.$$

3. Choose β that achieves the minimium empirical risk, subject to some constraint:

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

A solution: shrinkage estimators

1. Define a large family of "candidate classifiers", e.g., linear predictors:
$$f_\beta(x) = \beta^\top x \quad \text{for } x \in \mathbb{R}^p$$

2. For any candidate classifier f_β, quantify how "good" it is on the training set with some empirical risk, e.g.:
$$R(\beta) = \frac{1}{n}\sum_{i=1}^{n}(f_\beta(x_i) - y_i)^2 \,.$$

3. Choose β that achieves the minimium empirical risk, subject to some constraint:
$$\min_\beta R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C \,.$$

A solution: shrinkage estimators

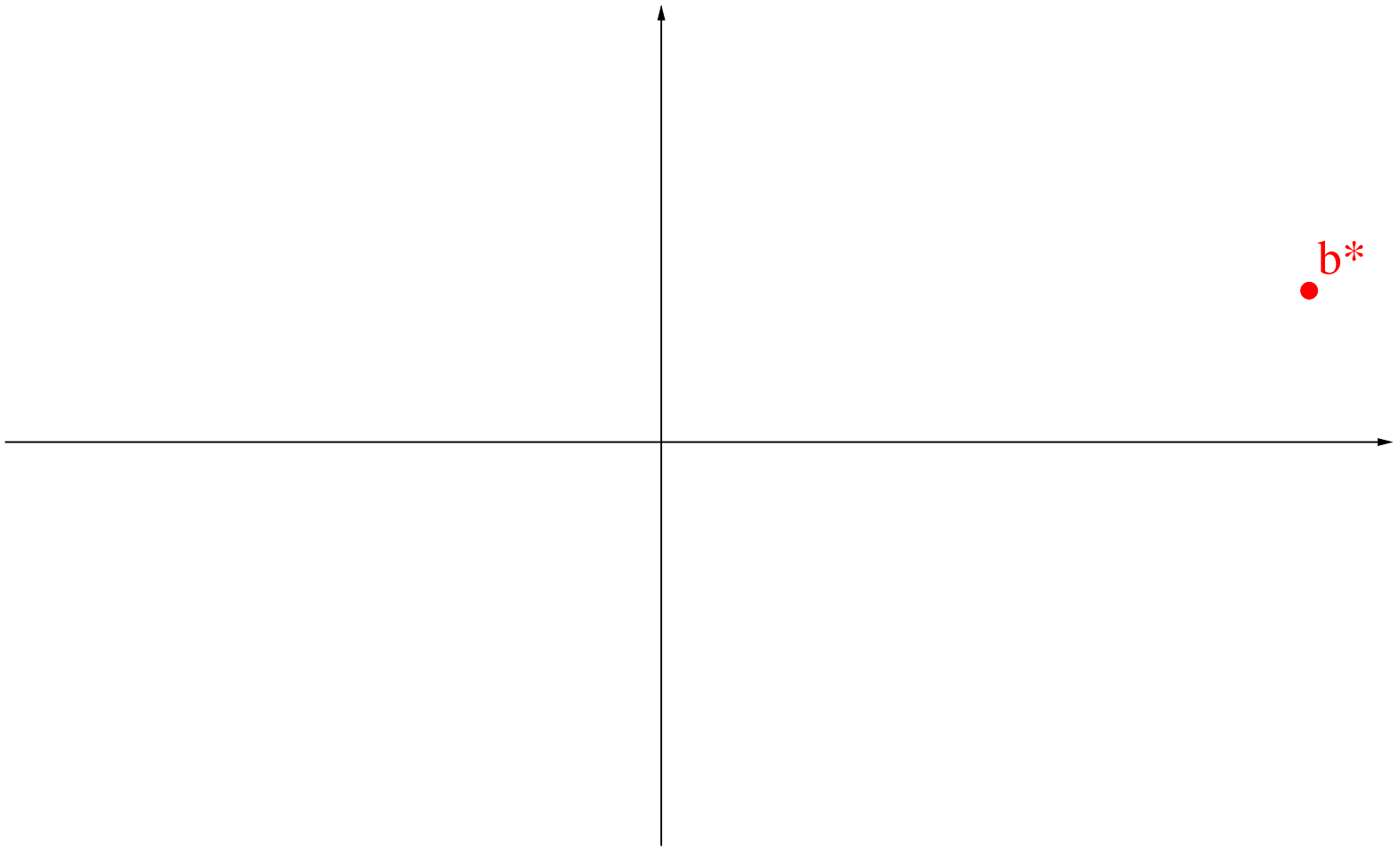
1. Define a large family of "candidate classifiers", e.g., linear predictors:
$$f_\beta(x) = \beta^\top x \quad \text{for } x \in \mathbb{R}^p$$

2. For any candidate classifier f_β, quantify how "good" it is on the training set with some empirical risk, e.g.:
$$R(\beta) = \frac{1}{n}\sum_{i=1}^{n}(f_\beta(x_i) - y_i)^2 \,.$$

3. Choose β that achieves the minimium empirical risk, subject to some constraint:
$$\min_\beta R(\beta) \quad \text{subject to} \quad \Omega(\beta) \leq C \,.$$

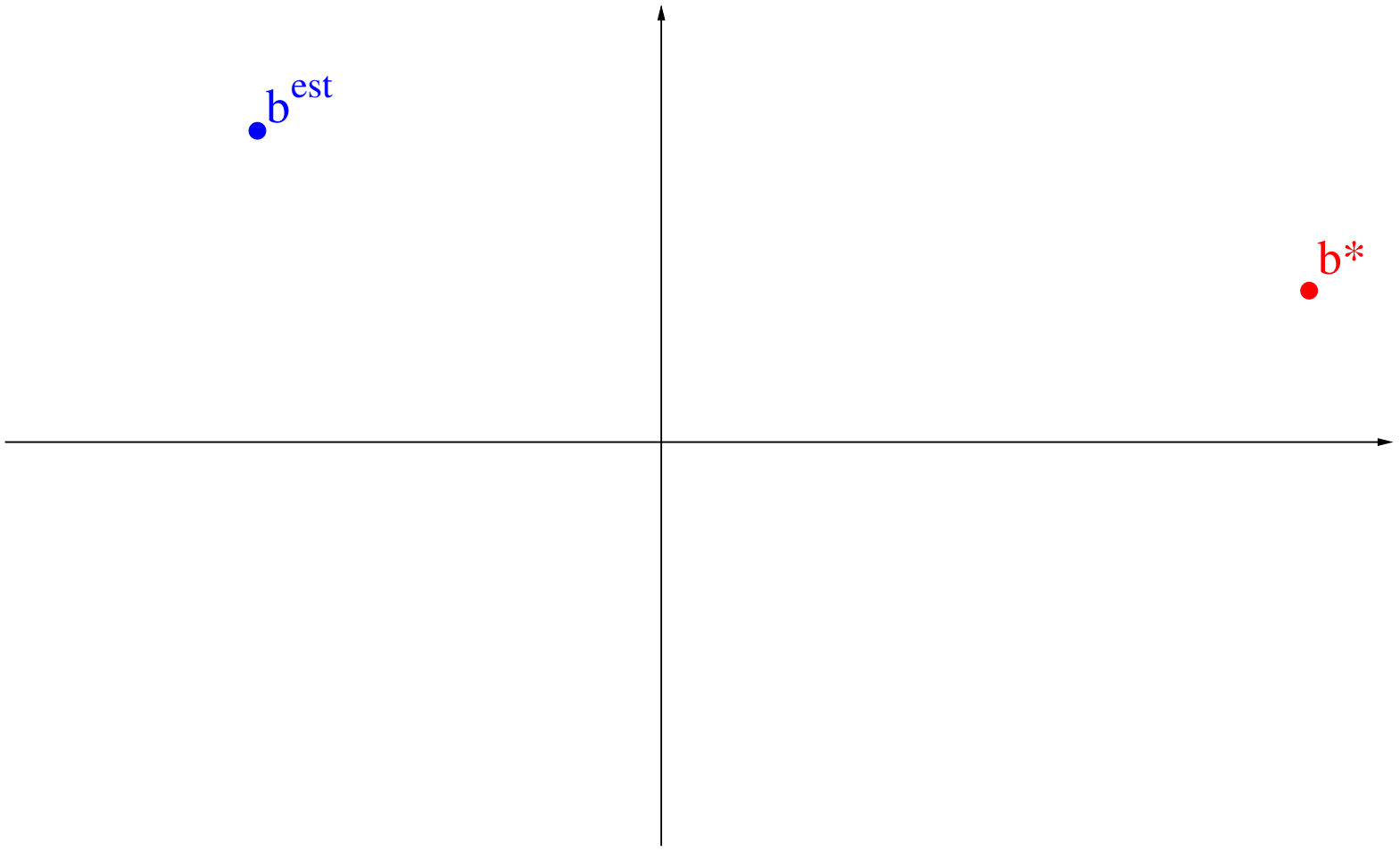
Why skrinkage classifiers?

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



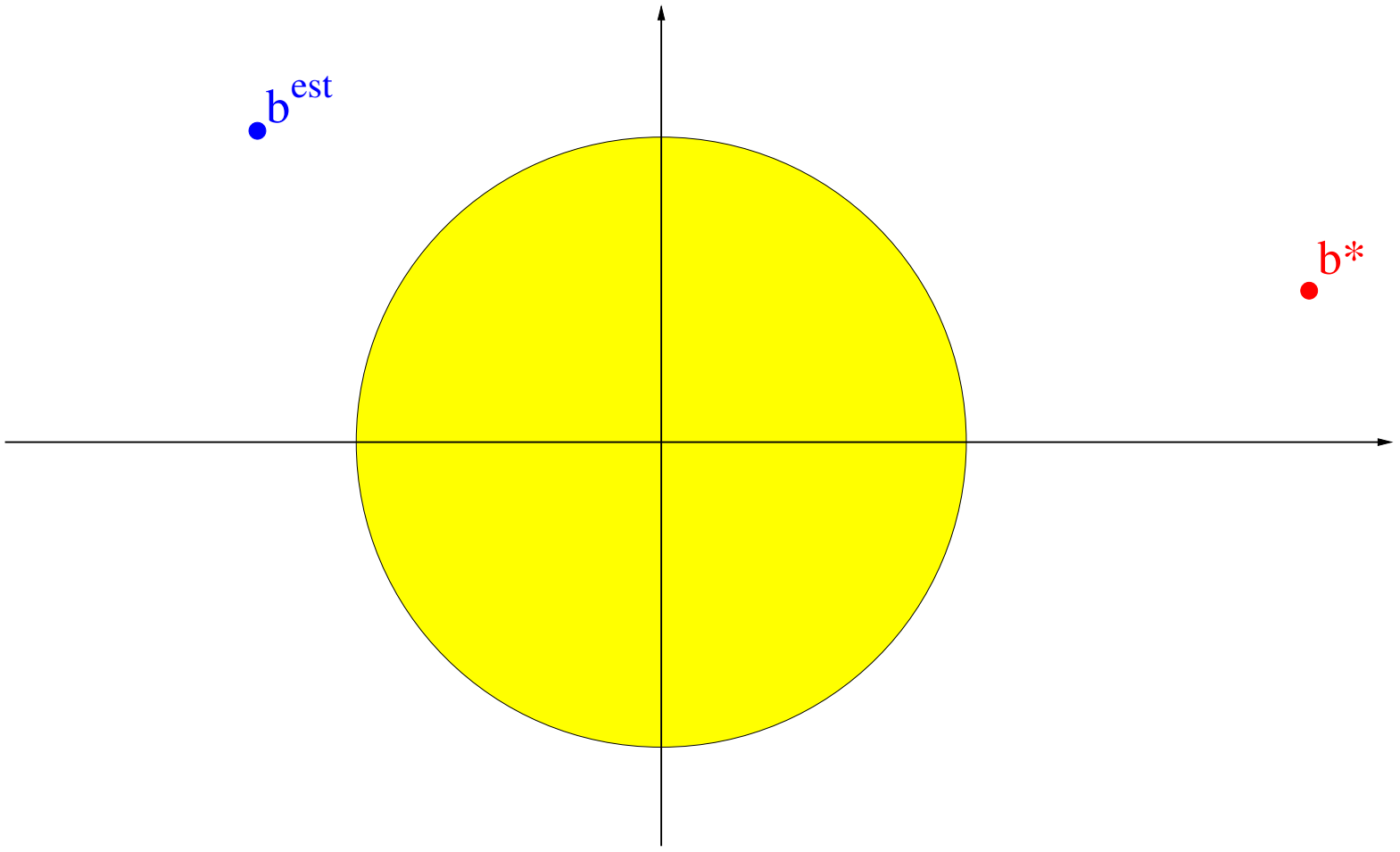
Why skrinkage classifiers?

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



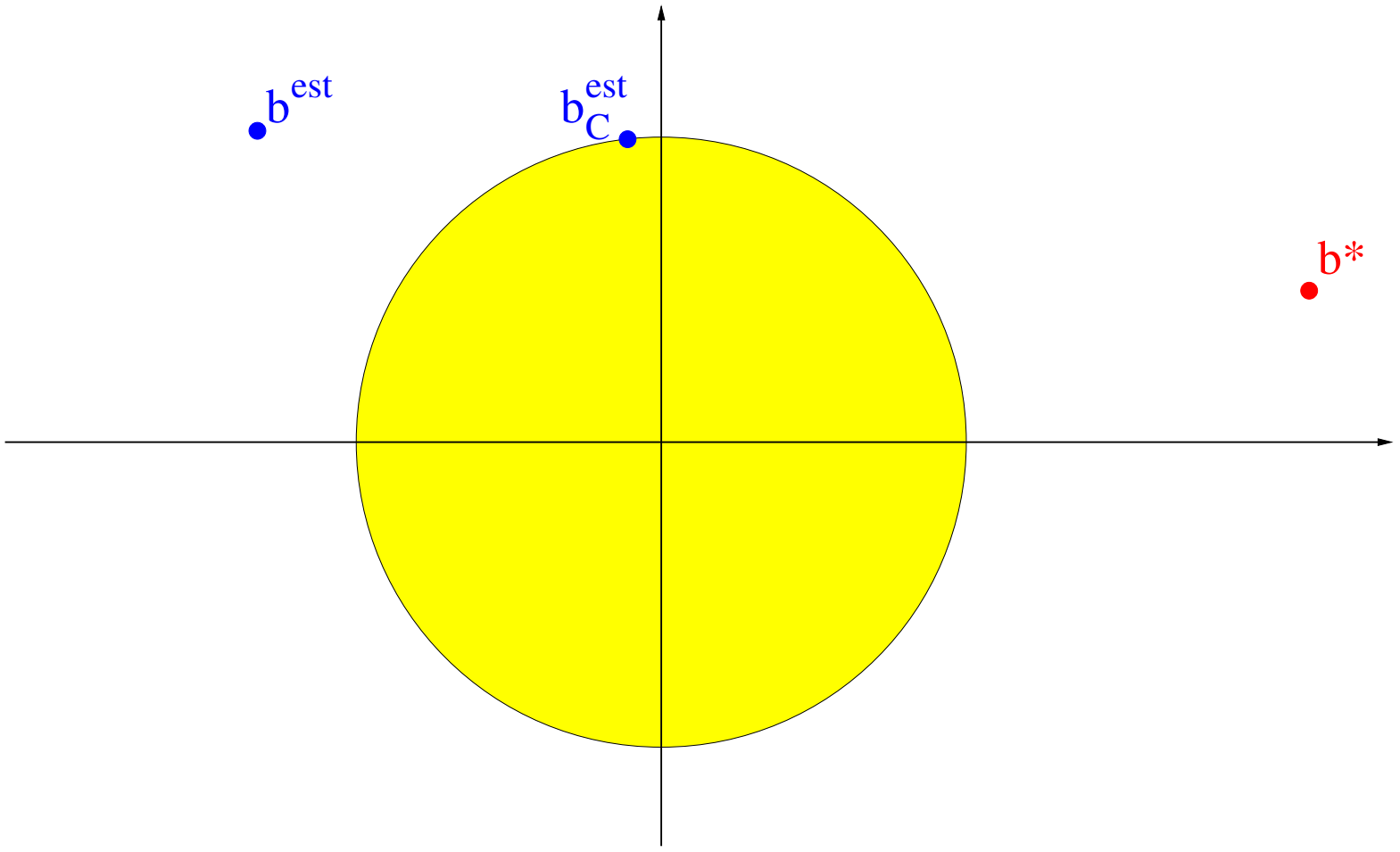
Why skrinkage classifiers?

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



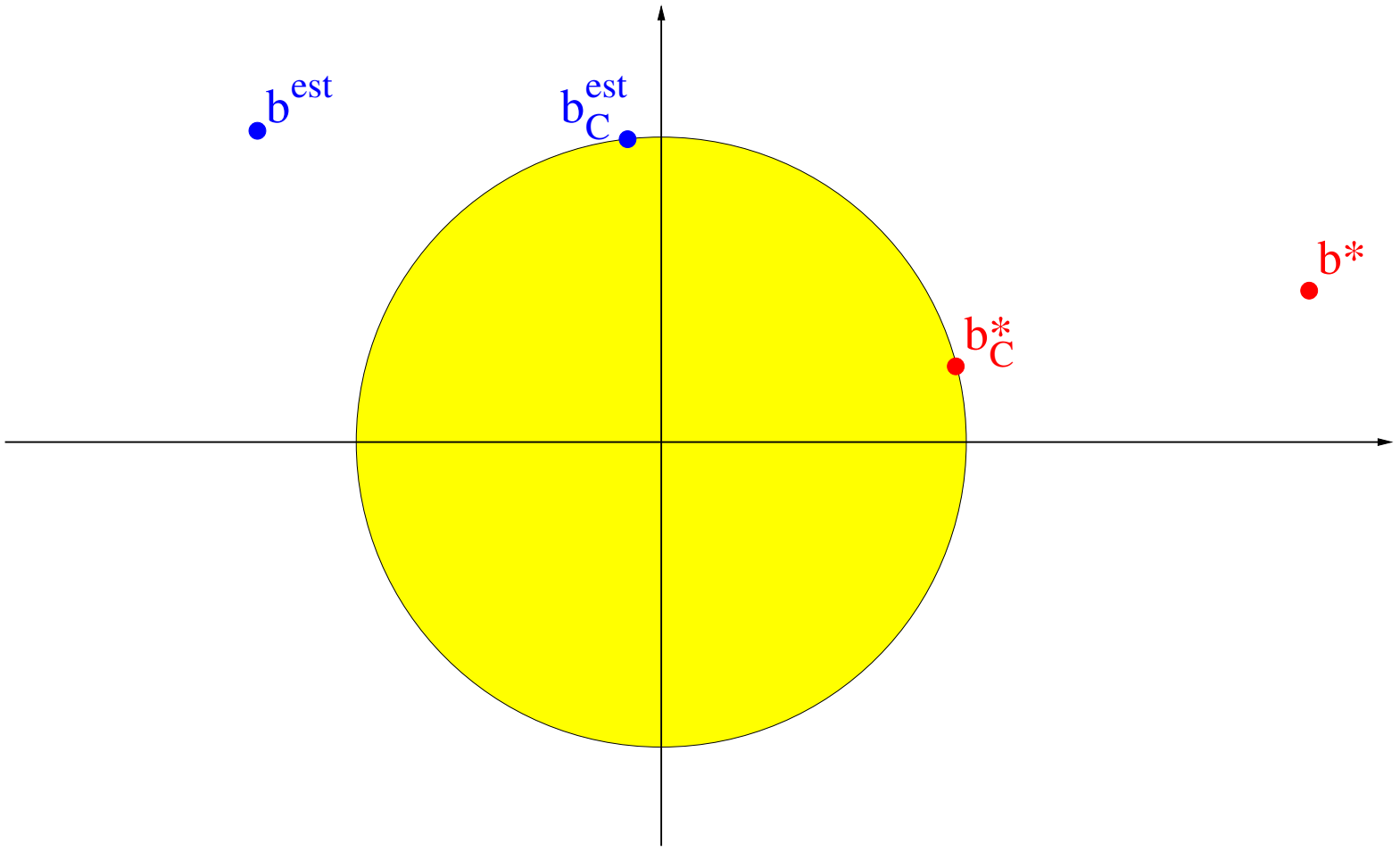
Why skrinkage classifiers?

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



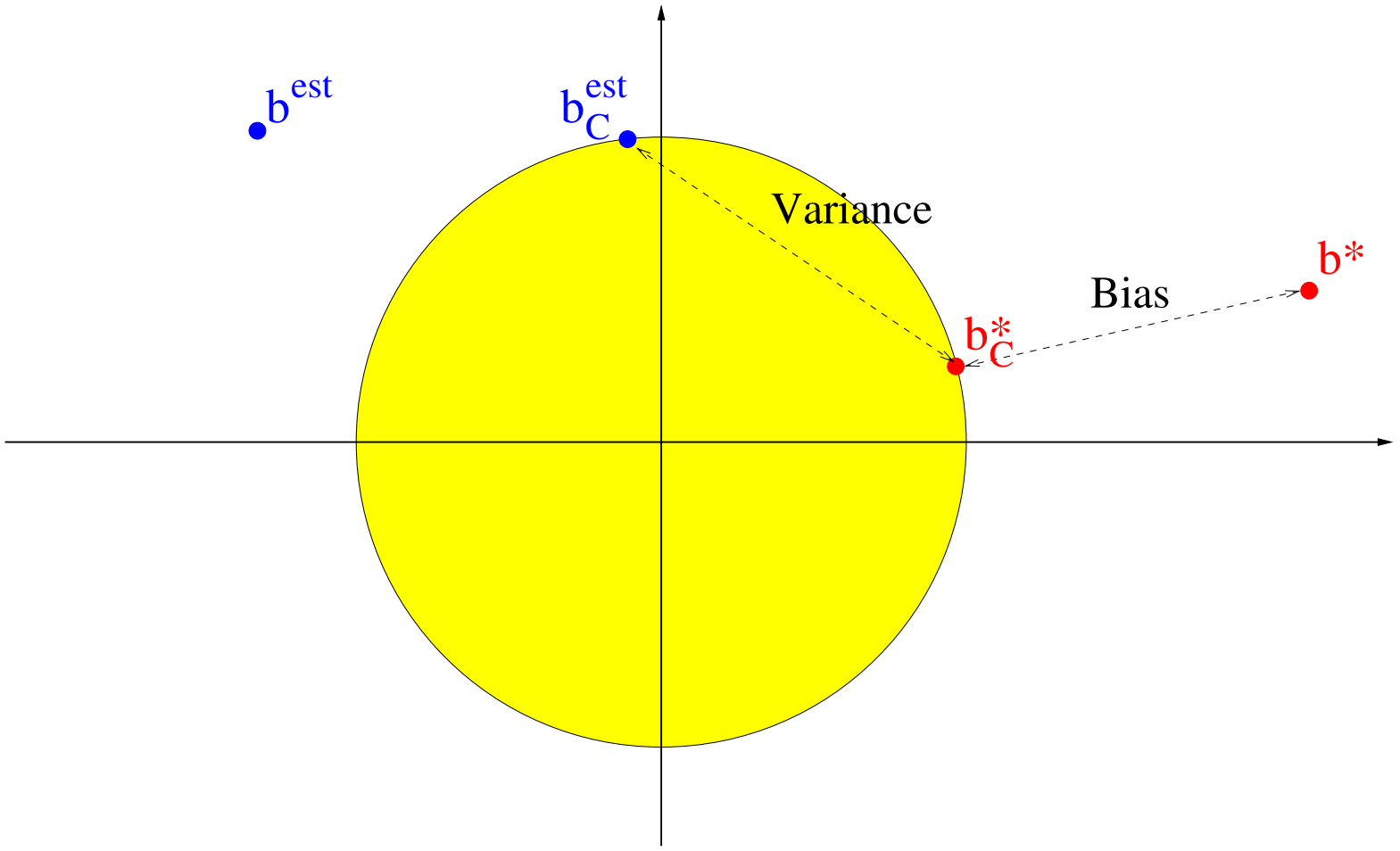
Why skrinkage classifiers?

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



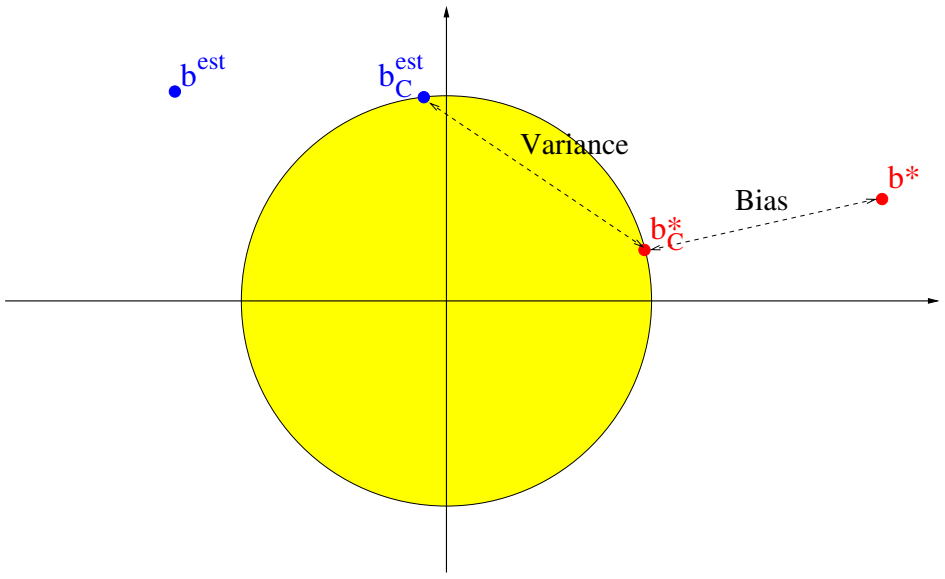
Why skrinkage classifiers?

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



Why skrinkage classifiers?

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

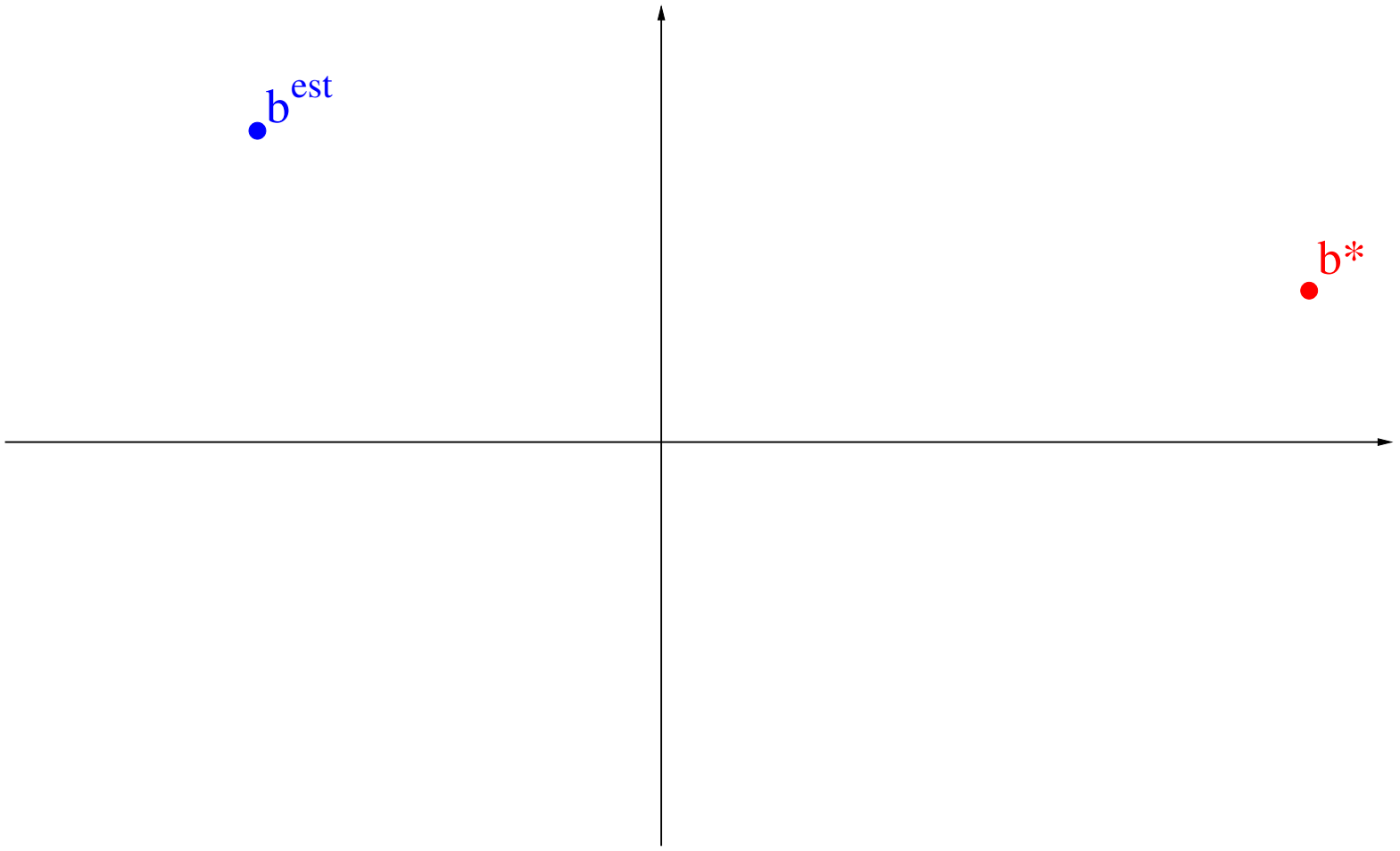


- "Increases bias and decreases variance"
- Equivalent formulation:

$$\min_{\beta} R(\beta) + \lambda\Omega(\beta)\,.$$

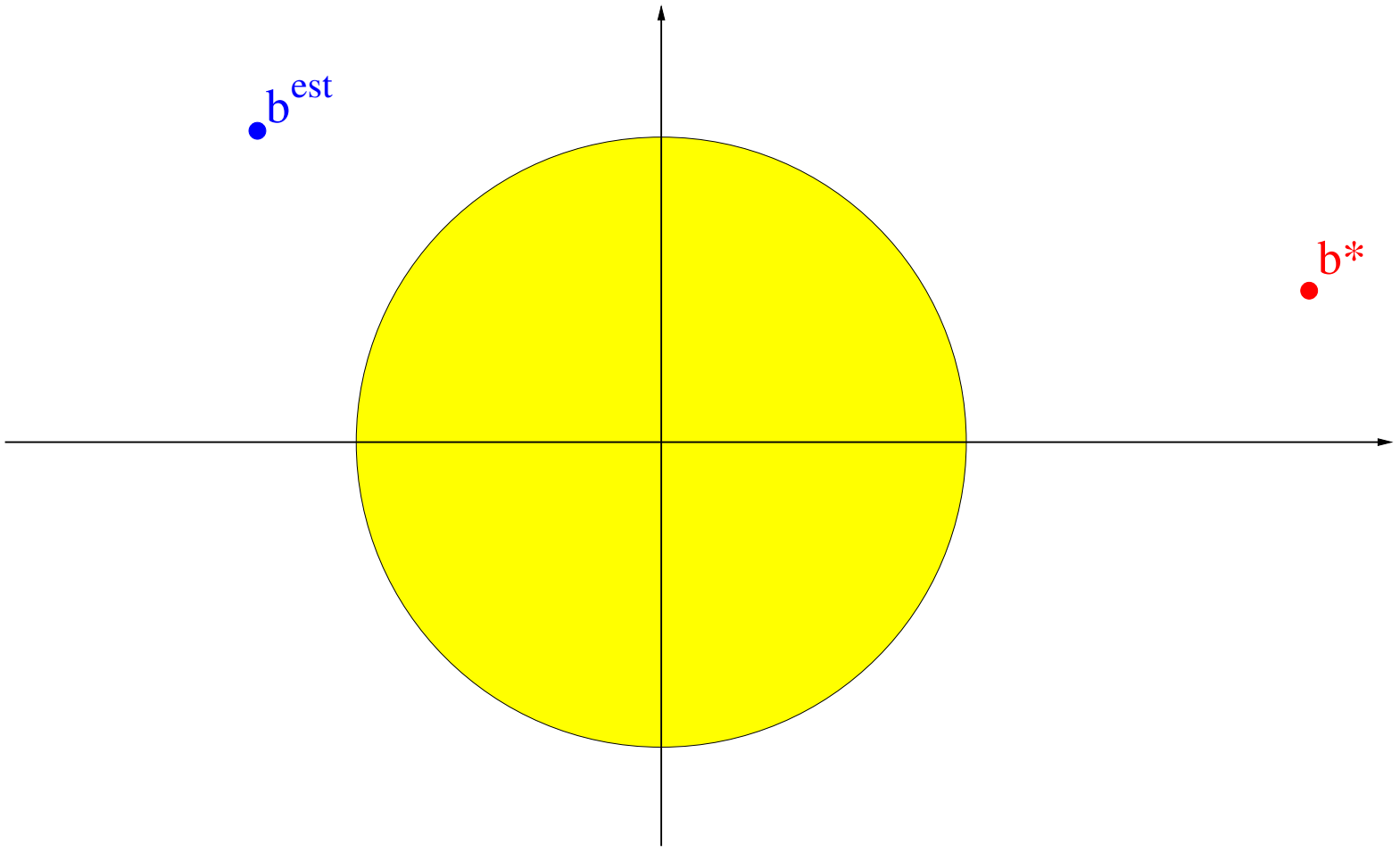
Choice of Ω can decrease the bias

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



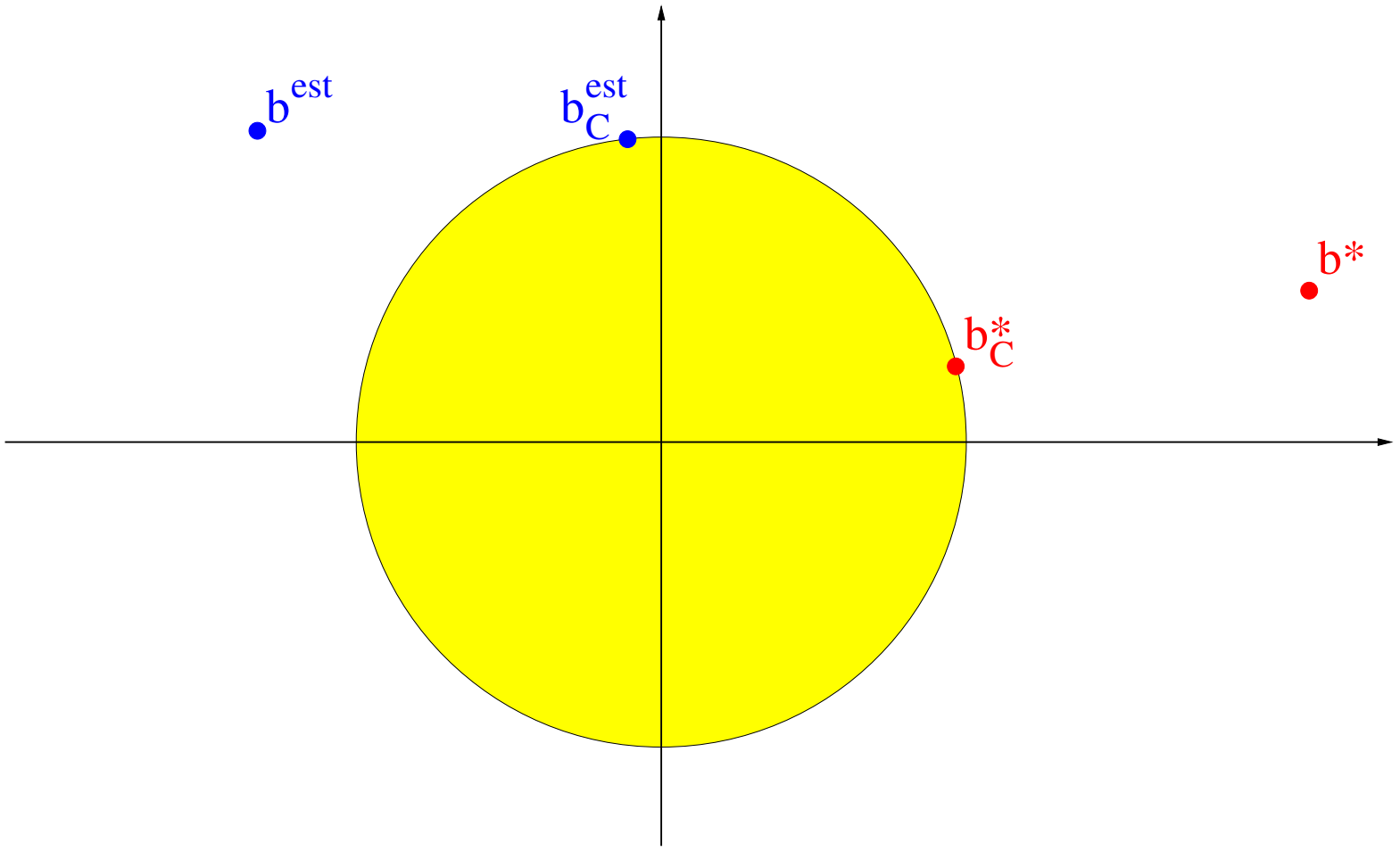
Choice of Ω can decrease the bias

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



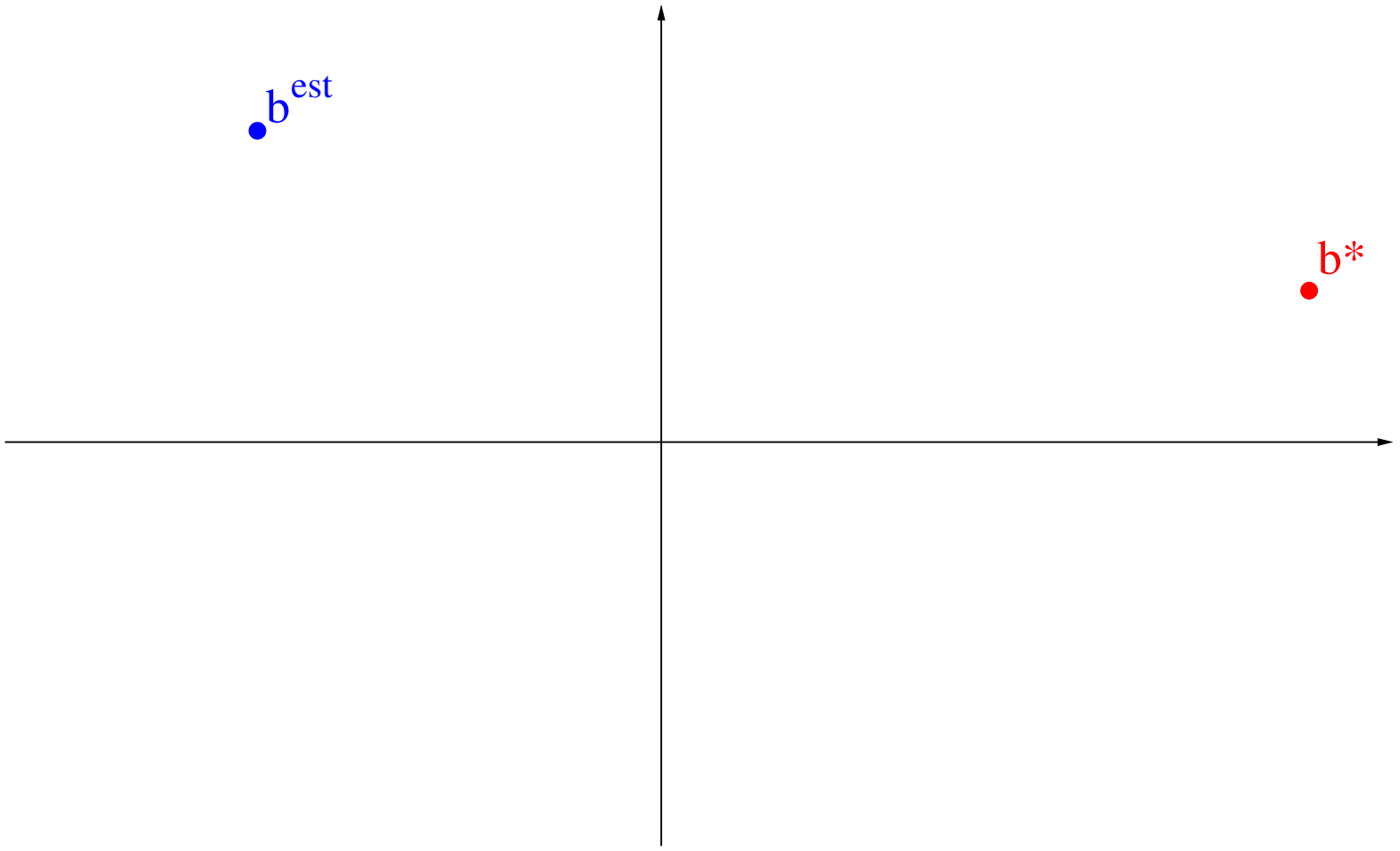
Choice of Ω can decrease the bias

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



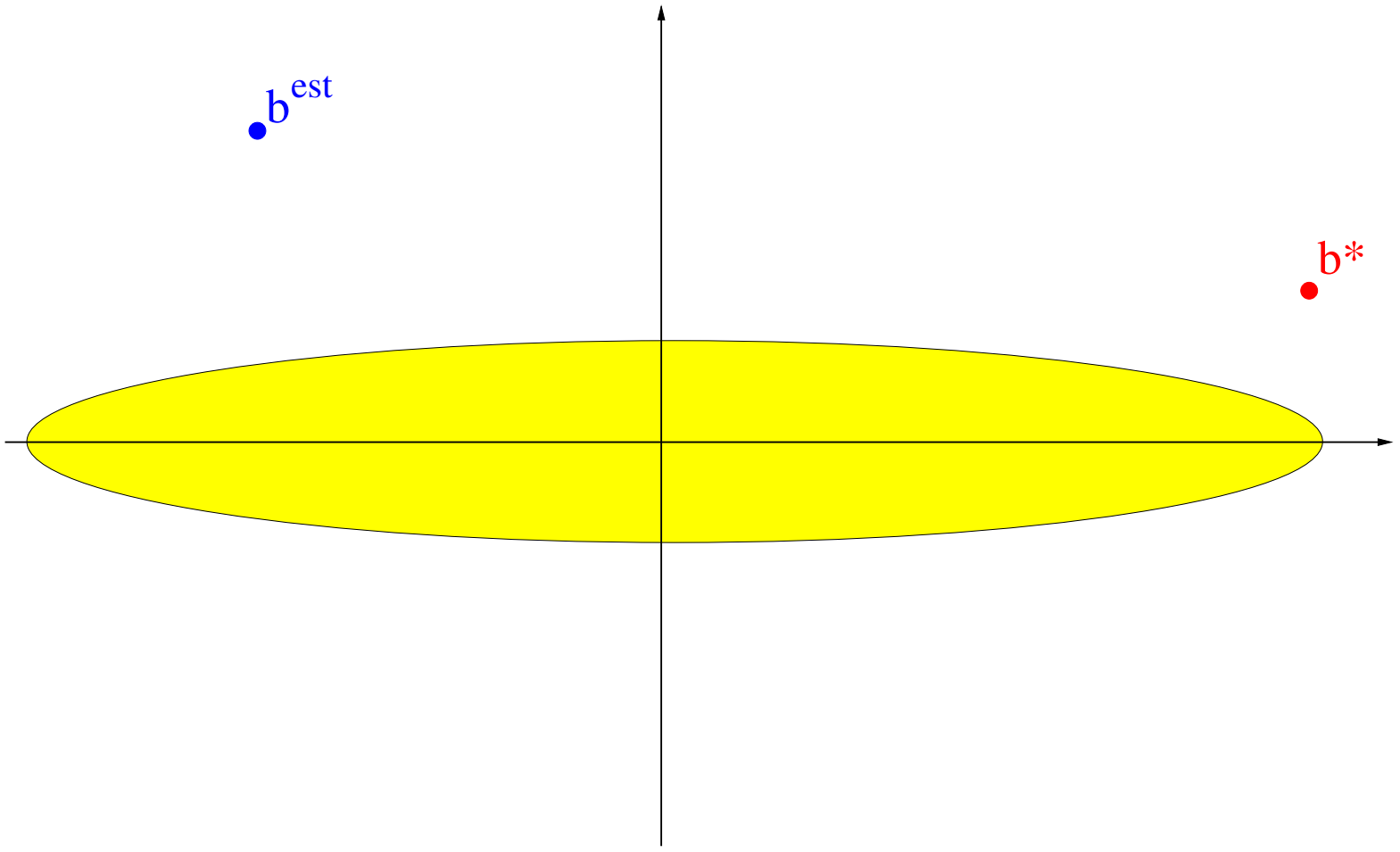
Choice of Ω can decrease the bias

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



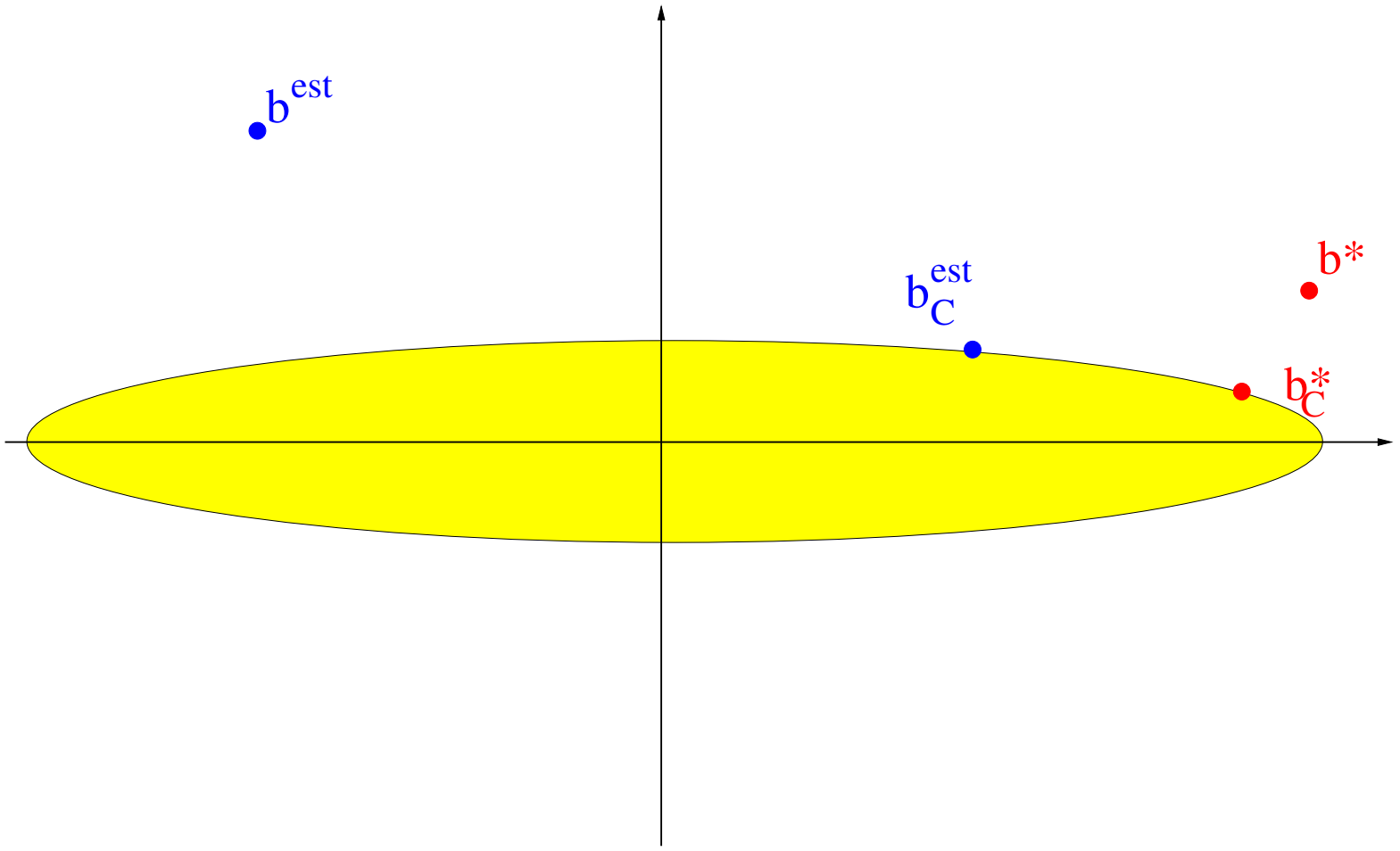
Choice of Ω can decrease the bias

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



Choice of Ω can decrease the bias

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

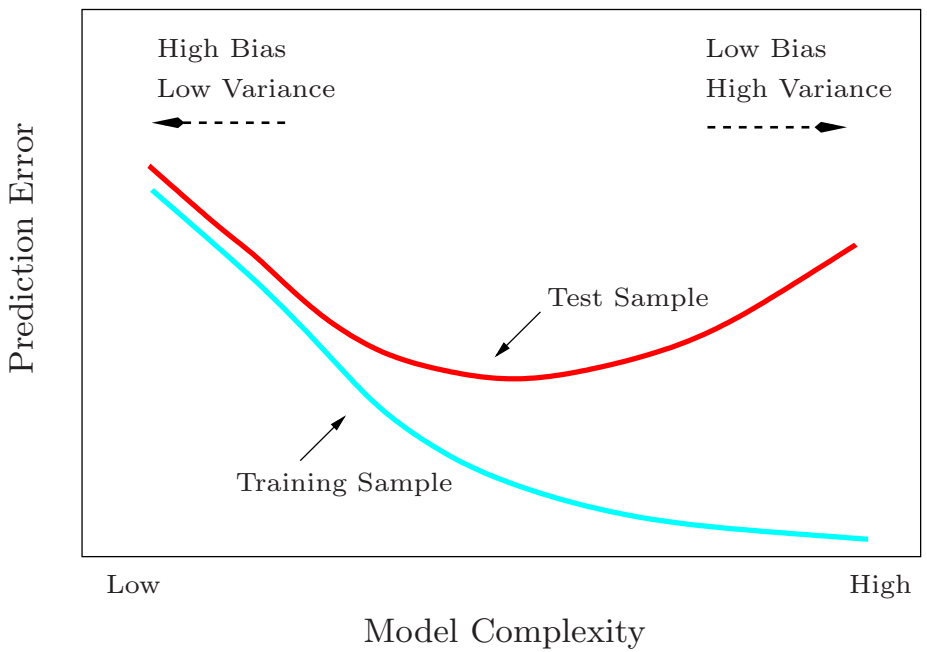


Choice of C or λ: structured regression and model selection

- Define a family of function classes $\mathcal{F}_\lambda$, where λ controls the "complexity"
- For each λ, define

$$\hat{f}_\lambda = \underset{\mathcal{F}_\lambda}{\operatorname{argmin}}\ EPE(f)$$

- Select $\hat{f} = \hat{f}_{\hat{\lambda}}$ to minimize the bias-variance tradeoff.



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.

Summary

1. Many problems in computational biology and medicine can be formulated as high-dimensional classification or regression tasks
2. The total error of a learning system is the sum of a bias and a variance error
3. In high dimension, the variance term often dominates
4. Shrinkage methods allow to control the bias/variance trade-off
5. The choice of the penalty is where we can put prior knowledge to decrease bias

Choosing or designing a penalty...

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

We will only focus on convex penalties, which lead to efficient algorithms. We will touch upon two important families of penalties:

1. Smooth convex penalty: ridge regression, SVM, kernels...
2. Nonsmooth convex penalty: lasso, group lasso, fused lasso,...

Outline

Outline

Ridge regression (RR)

1. Consider linear predictors:

$$f_\beta(x) = \beta^\top x \quad \text{for } x \in \mathbb{R}^p$$

2. Consider the RSS empirical risk:

$$R(\beta) = \frac{1}{n}\sum_{i=1}^{n}(f_\beta(x_i) - y_i)^2 .$$

3. Consider the Euclidean norm as a penalty:

$$\Omega(\beta) = \|\beta\|^2 = \sum_{i=1}^{p}\beta_i^2$$

Ridge regression (RR)

1. Consider linear predictors:

$$f_\beta(x) = \beta^\top x \quad \text{for } x \in \mathbb{R}^p$$

2. Consider the RSS empirical risk:

$$R(\beta) = \frac{1}{n}\sum_{i=1}^{n}(f_\beta(x_i) - y_i)^2 \,.$$

3. Consider the Euclidean norm as a penalty:

$$\Omega(\beta) = \|\beta\|^2 = \sum_{i=1}^{p}\beta_i^2$$

Ridge regression (RR)

1. Consider linear predictors:

$$f_\beta(x) = \beta^\top x \quad \text{for } x \in \mathbb{R}^p$$

2. Consider the RSS empirical risk:

$$R(\beta) = \frac{1}{n} \sum_{i=1}^{n} (f_\beta(x_i) - y_i)^2 \,.$$

3. Consider the Euclidean norm as a penalty:

$$\Omega(\beta) = \|\beta\|^2 = \sum_{i=1}^{p} \beta_i^2$$

Ridge regression (RR)

$$\min_{\beta \in \mathbb{R}^{p+1}} R(\beta) + \lambda \Omega(\beta)$$

$$= \sum_{i=1}^{n} \left(f_\beta \left(\mathbf{x}_i \right) - \mathbf{y}_i \right)^2 + \lambda \sum_{i=1}^{p} \beta_i^2 \tag{1}$$

$$= \left(y - X\beta \right)^\top \left(y - X\beta \right) + \lambda \beta^\top \beta \, .$$

Explicit solution:

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

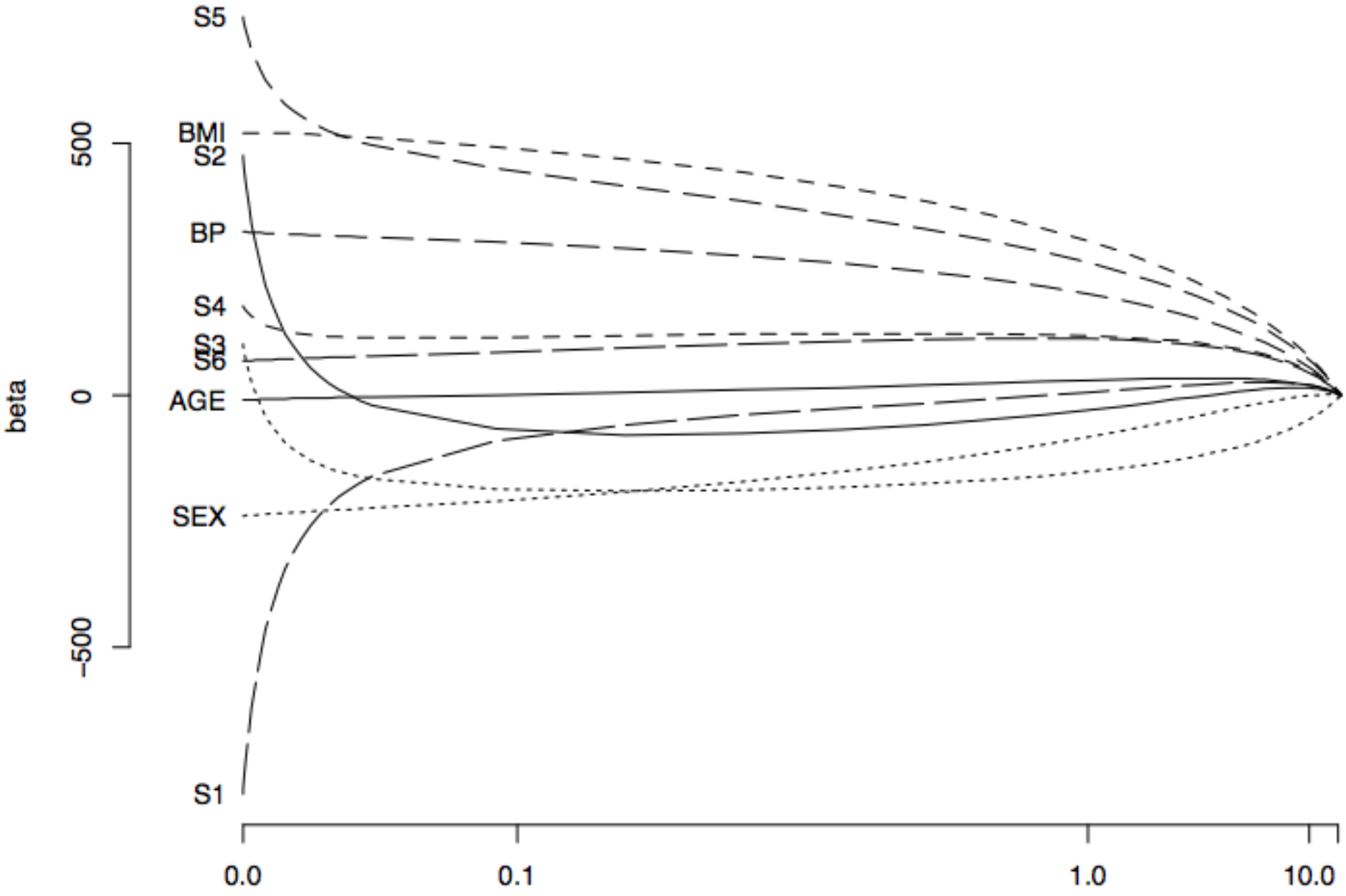
Ridge regression (RR)

$$\begin{aligned}
&\min_{\beta \in \mathbb{R}^{p+1}} R(\beta) + \lambda \Omega(\beta) \\
&\qquad = \sum_{i=1}^{n} \left(f_\beta \left(\mathbf{x}_i \right) - \mathbf{y}_i \right)^2 + \lambda \sum_{i=1}^{p} \beta_i^2 \qquad (1) \\
&\qquad = \left(y - X\beta \right)^\top \left(y - X\beta \right) + \lambda \beta^\top \beta \, .
\end{aligned}$$

Explicit solution:

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

Ridge regression example



Generalizations

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

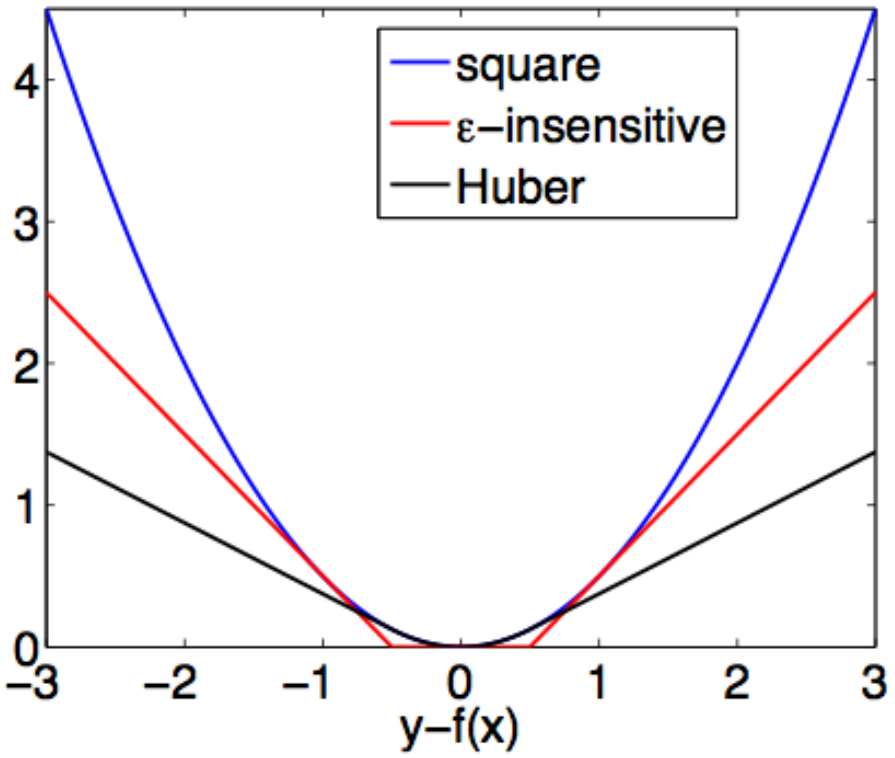
where

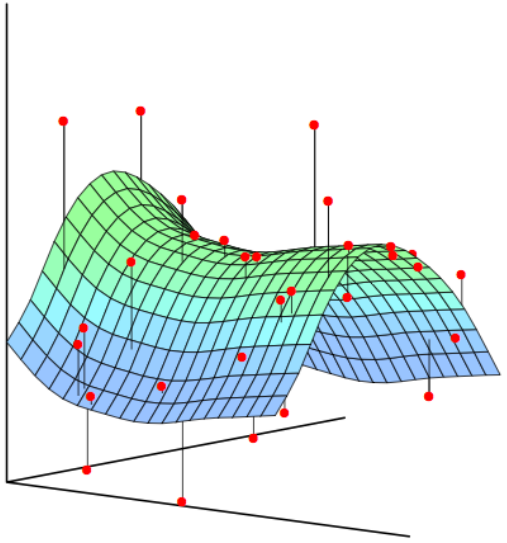
$$R(\beta) = \frac{1}{n} \sum_{i=1}^{n} \ell(f_\beta(x_i), y_i) \,.$$

for more general loss functions ℓ

Loss for regression

- Square loss : $\ell\left(f\left(\mathbf{x}\right),\mathbf{y}\right)=\left(f\left(\mathbf{x}\right)-\mathbf{y}\right)^2$
- ϵ-insensitive loss : $\ell\left(f\left(\mathbf{x}\right),\mathbf{y}\right)=\left(\left|f\left(\mathbf{x}\right)-\mathbf{y}\right|-\epsilon\right)_+$
- Huber loss : mixed quadratic/linear

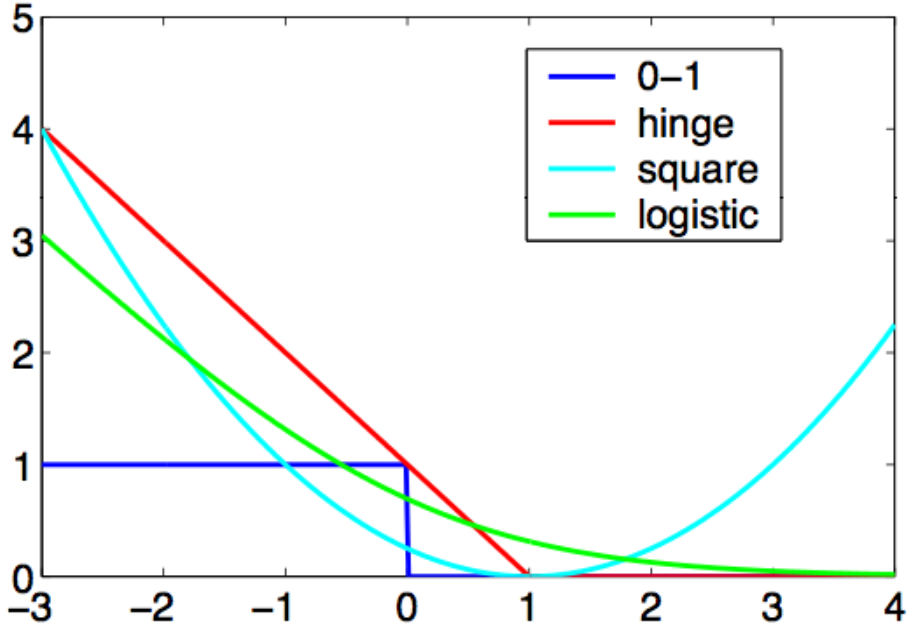


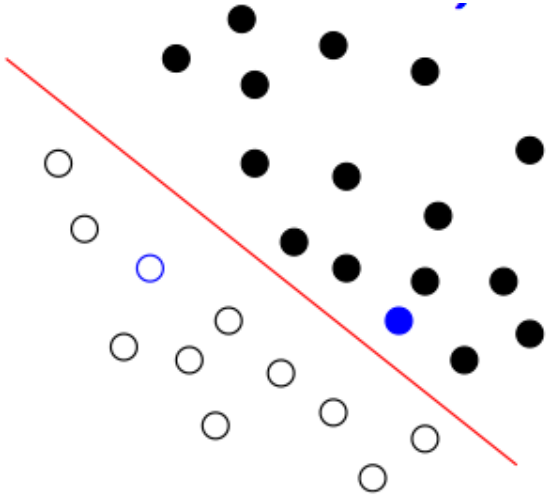


Loss for pattern recognition

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 $(\mathbf{x}, \mathbf{y})$ is: $\mathbf{y}f(\mathbf{x})$.
- The loss function is usually a decreasing function of the margin : $\ell(f(\mathbf{x}), \mathbf{y}) = \phi(\mathbf{y}f(\mathbf{x}))$,





Example: logistic regression

$$\ell(f(x), y) = \ln\left(1 + e^{-yf(x)}\right)$$

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

No explicit solution, optimization by Newton-Raphson (called iteratively reweighted least squares, IRLS)

$$\frac{\partial J}{\partial \beta}(\beta) = -\frac{1}{n}\sum_{i=1}^{n} \frac{y_i x_i}{1 + e^{y_i \beta^\top x_i}} + 2\lambda\beta = -\frac{1}{n}\sum_{i=1}^{n} y_i p(-y_i \mid x_i) x_i + 2\lambda\beta$$

$$\frac{\partial^2 J}{\partial \beta \partial \beta^\top}(\beta) = \frac{1}{n}\sum_{i=1}^{n} \frac{x_i x_i^\top e^{\beta^\top x_i}}{\left(1 + e^{\beta^\top x_i}\right)^2} + 2\lambda I$$

$$= \frac{1}{n}\sum_{i=1}^{n} p(1 \mid x_i)\left(1 - p(1 \mid x_i)\right) x_i x_i^\top + 2\lambda I$$

Probabilistic interpretation of logistic regression

Exercice

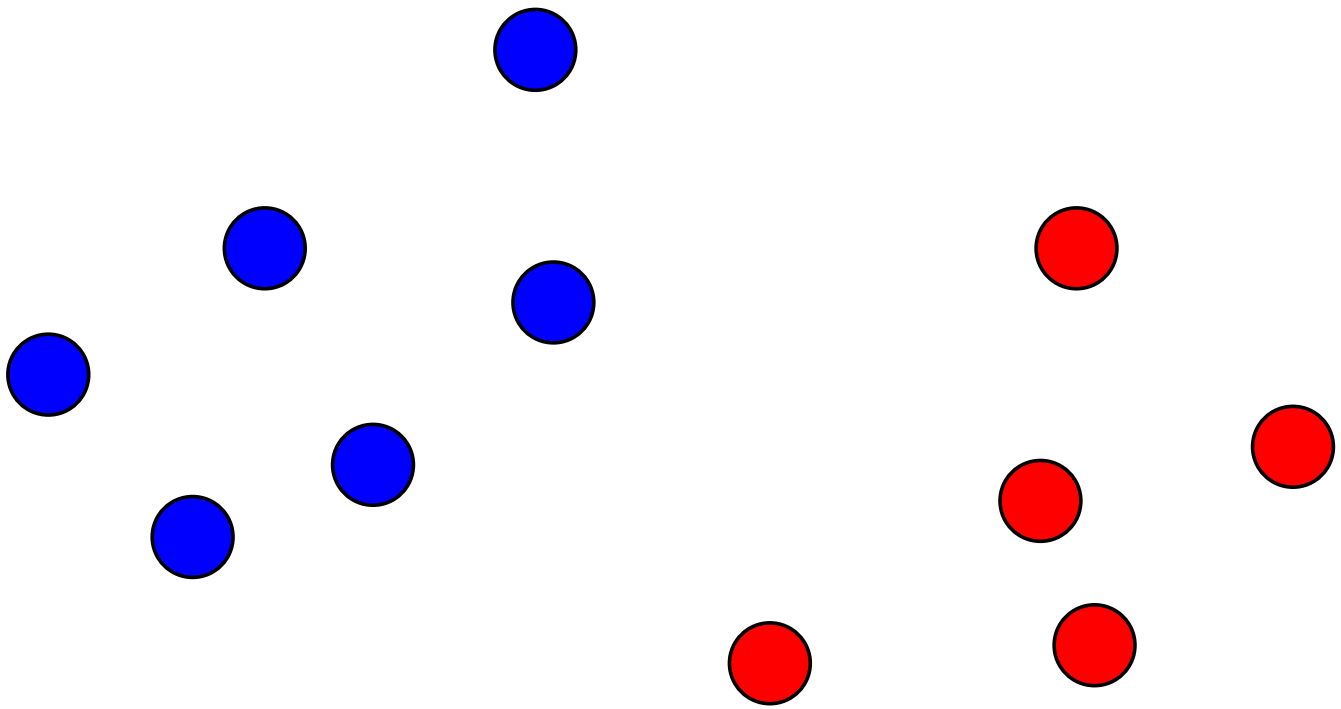
Show that logistic regression finds the penalized maximum likelihood estimator:

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

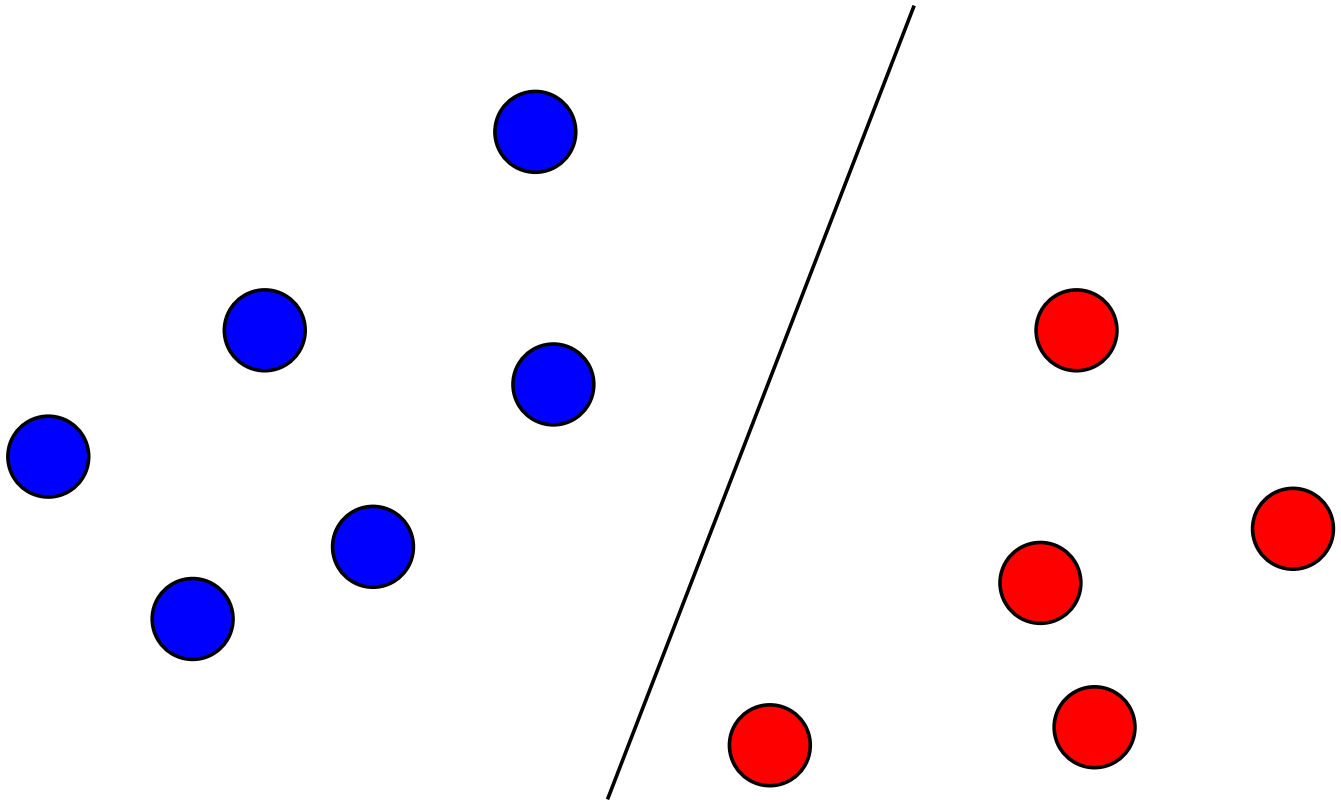
for the following model:

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

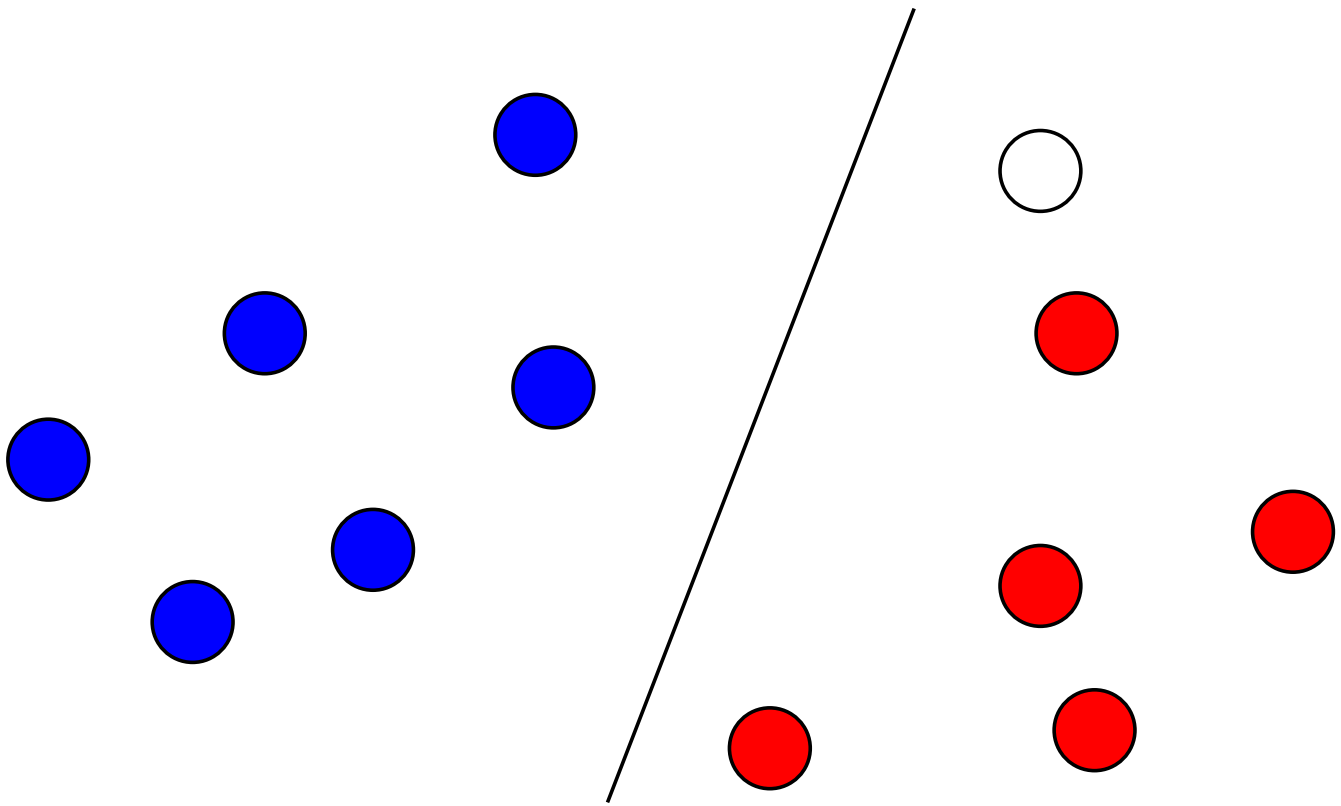
Example: hard-margin SVM



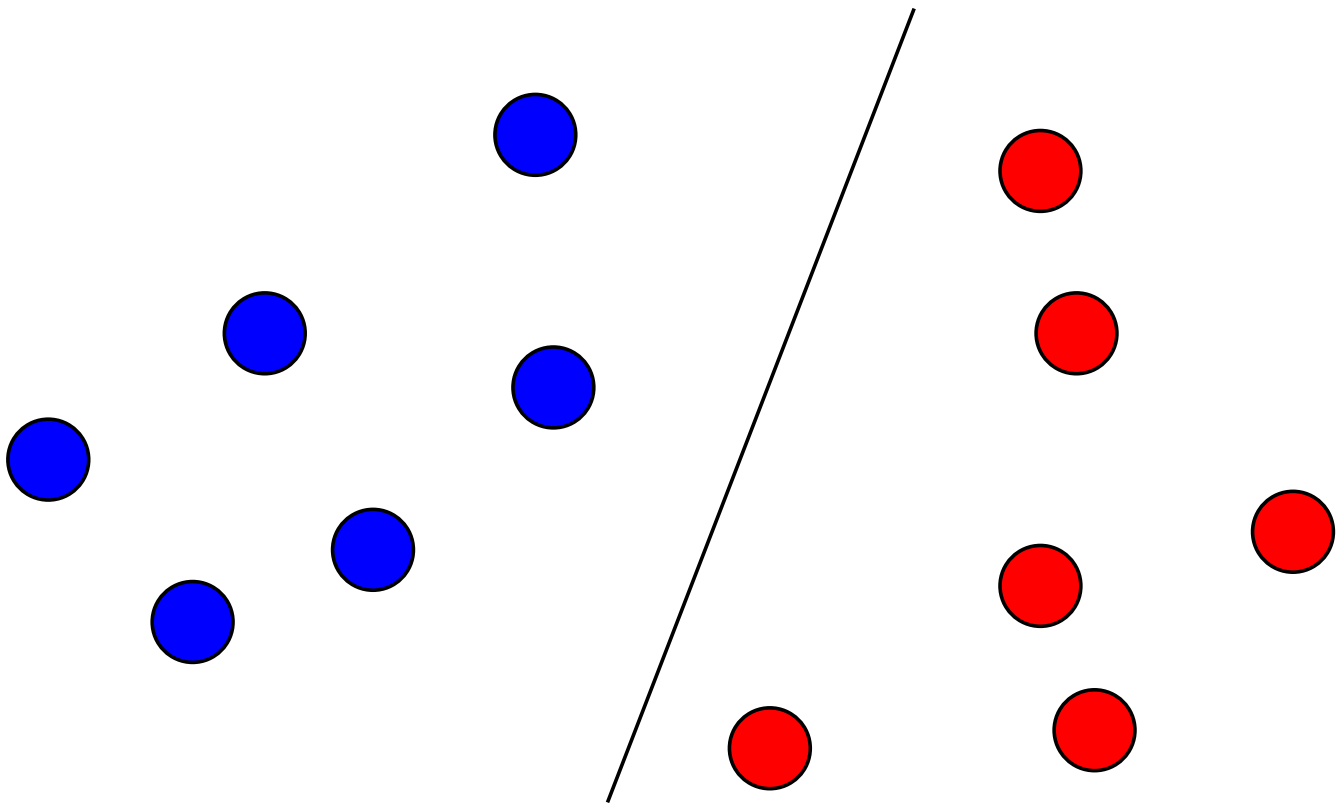
Example: hard-margin SVM



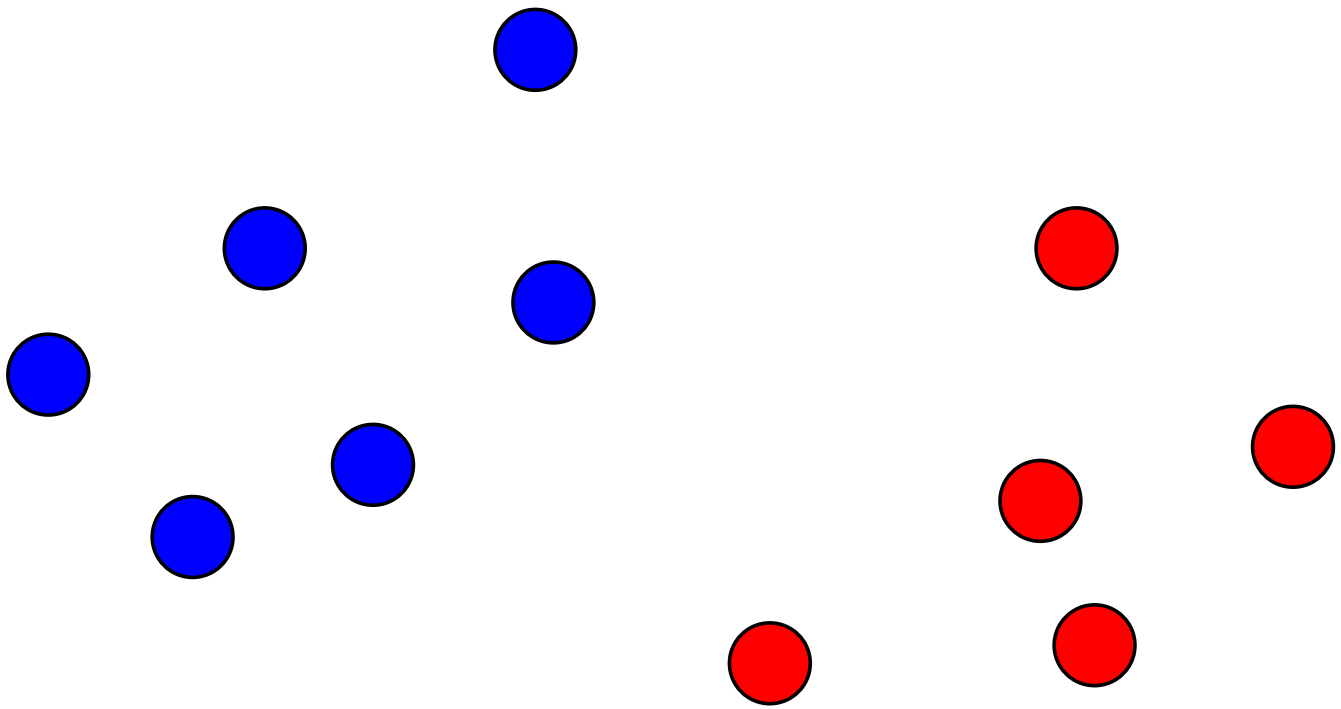
Example: hard-margin SVM



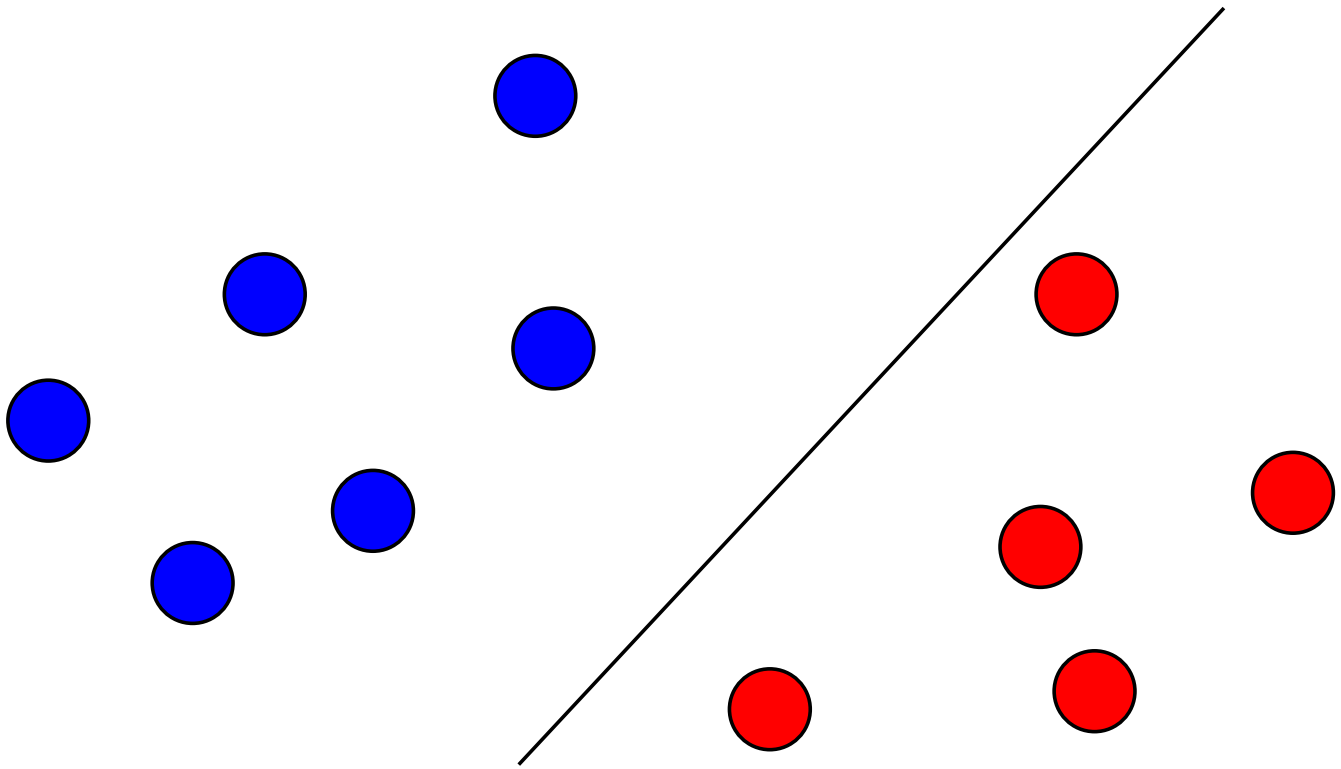
Example: hard-margin SVM



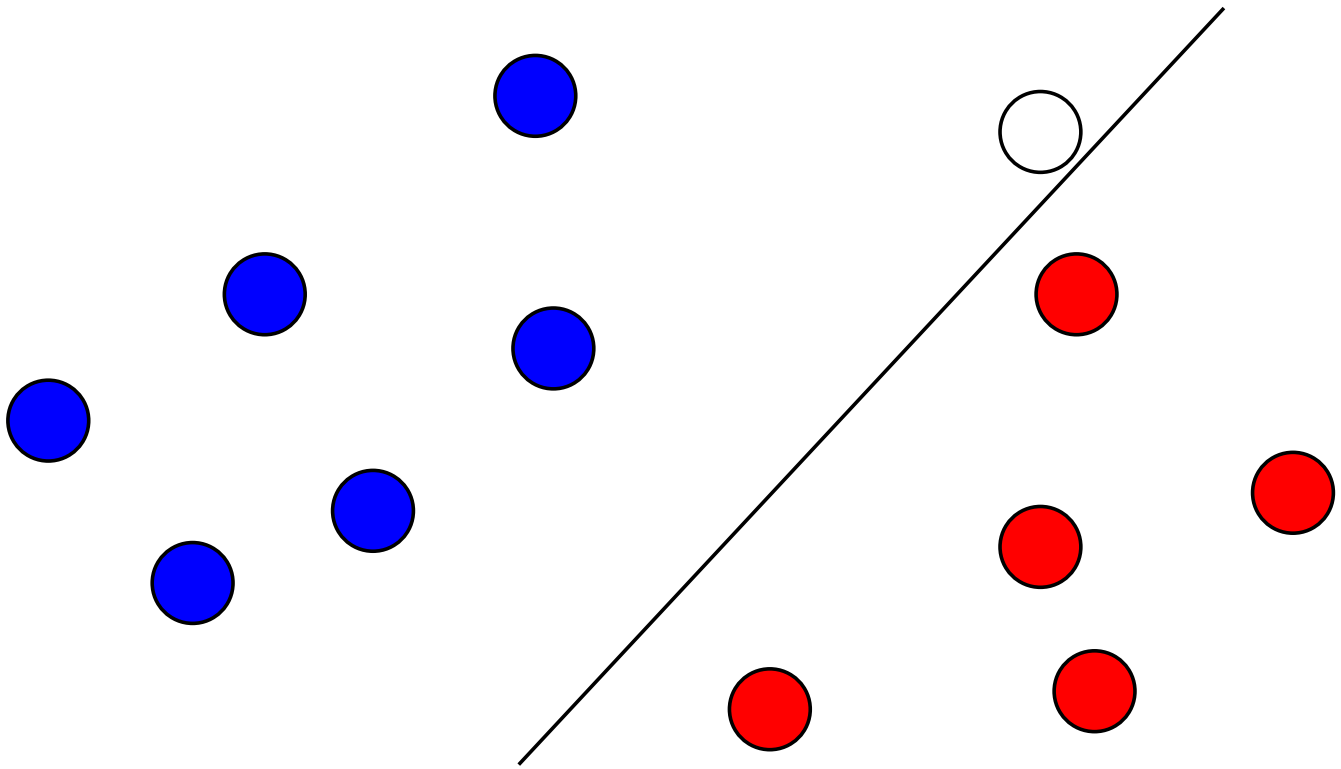
Example: hard-margin SVM



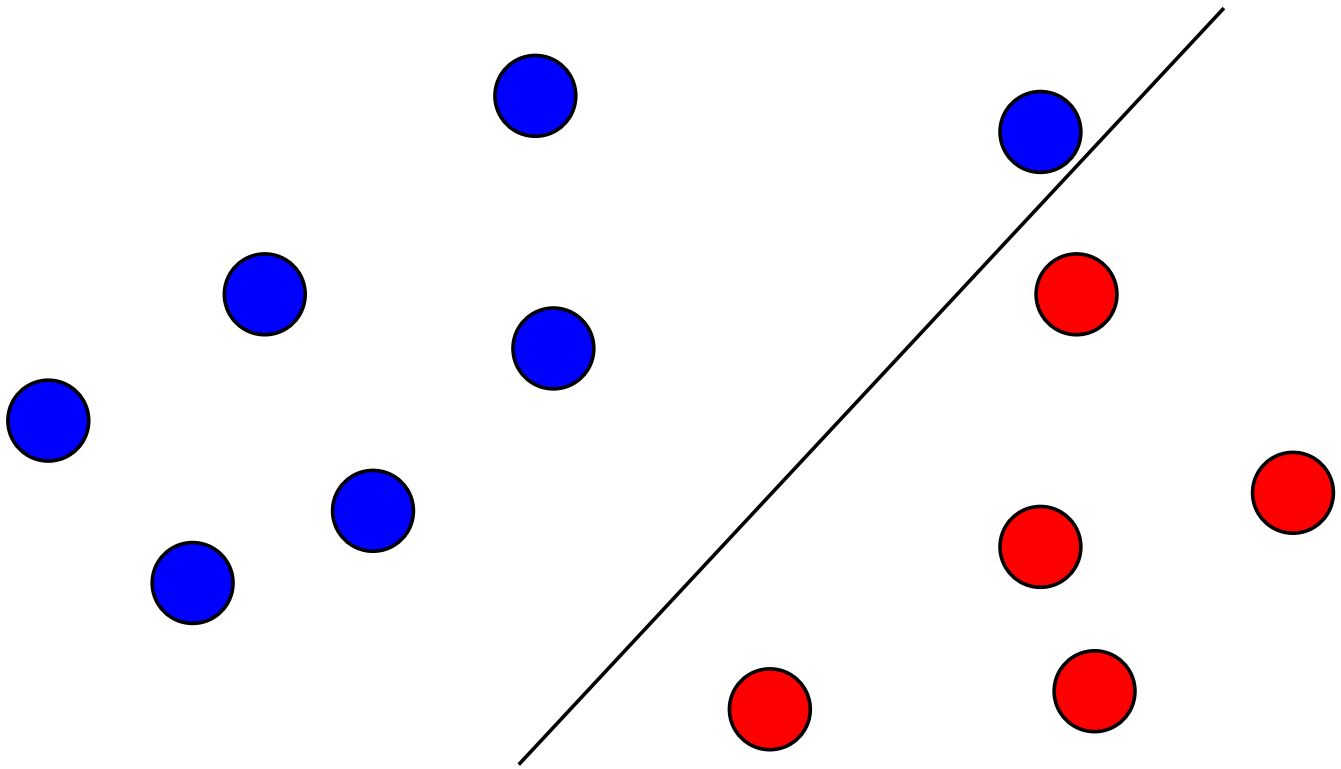
Example: hard-margin SVM



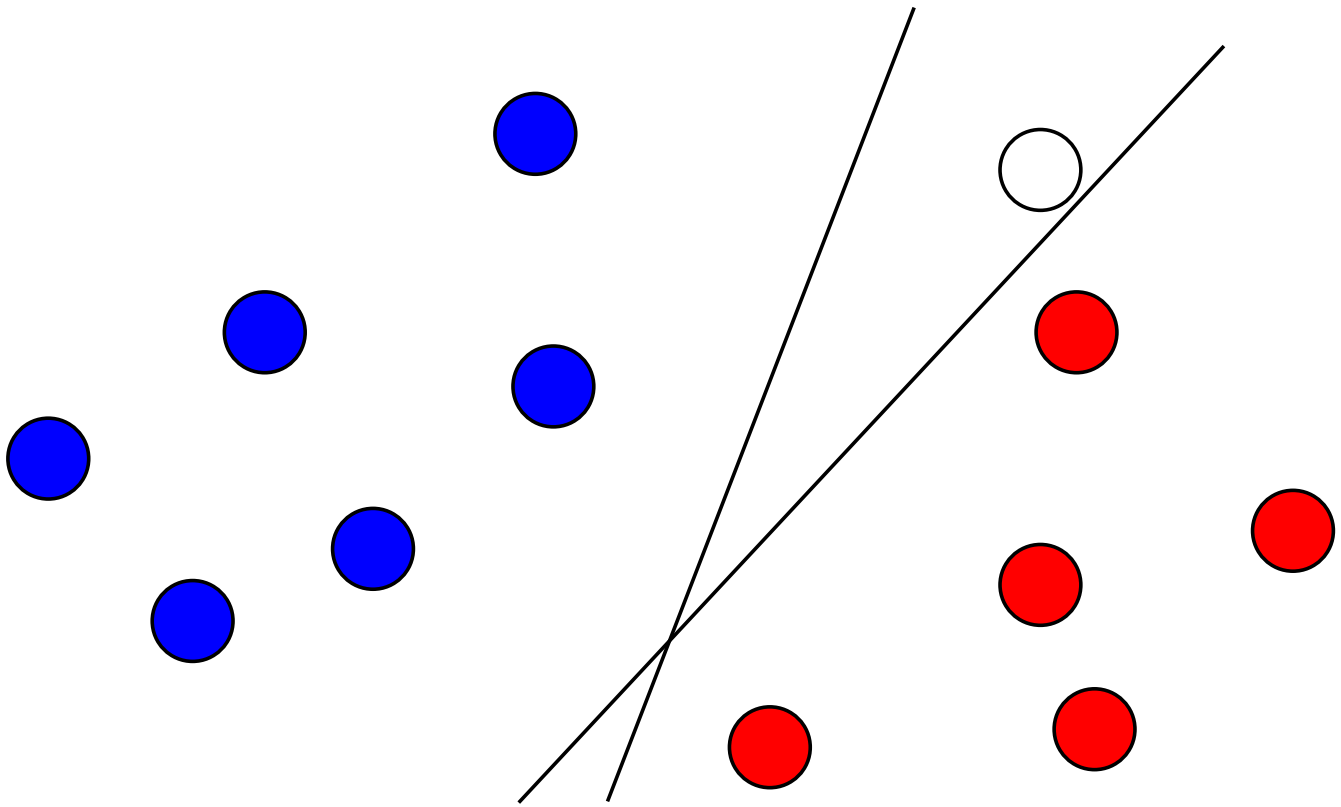
Example: hard-margin SVM



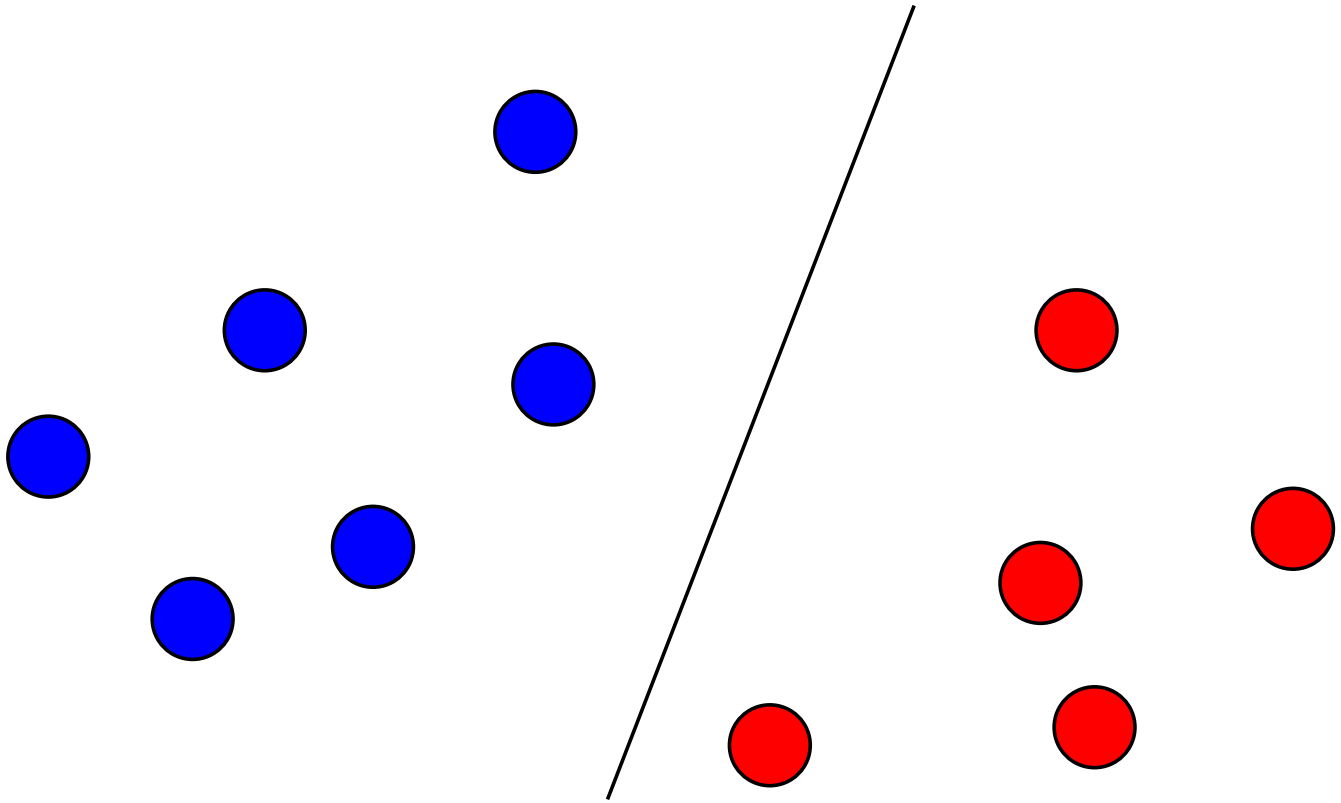
Example: hard-margin SVM



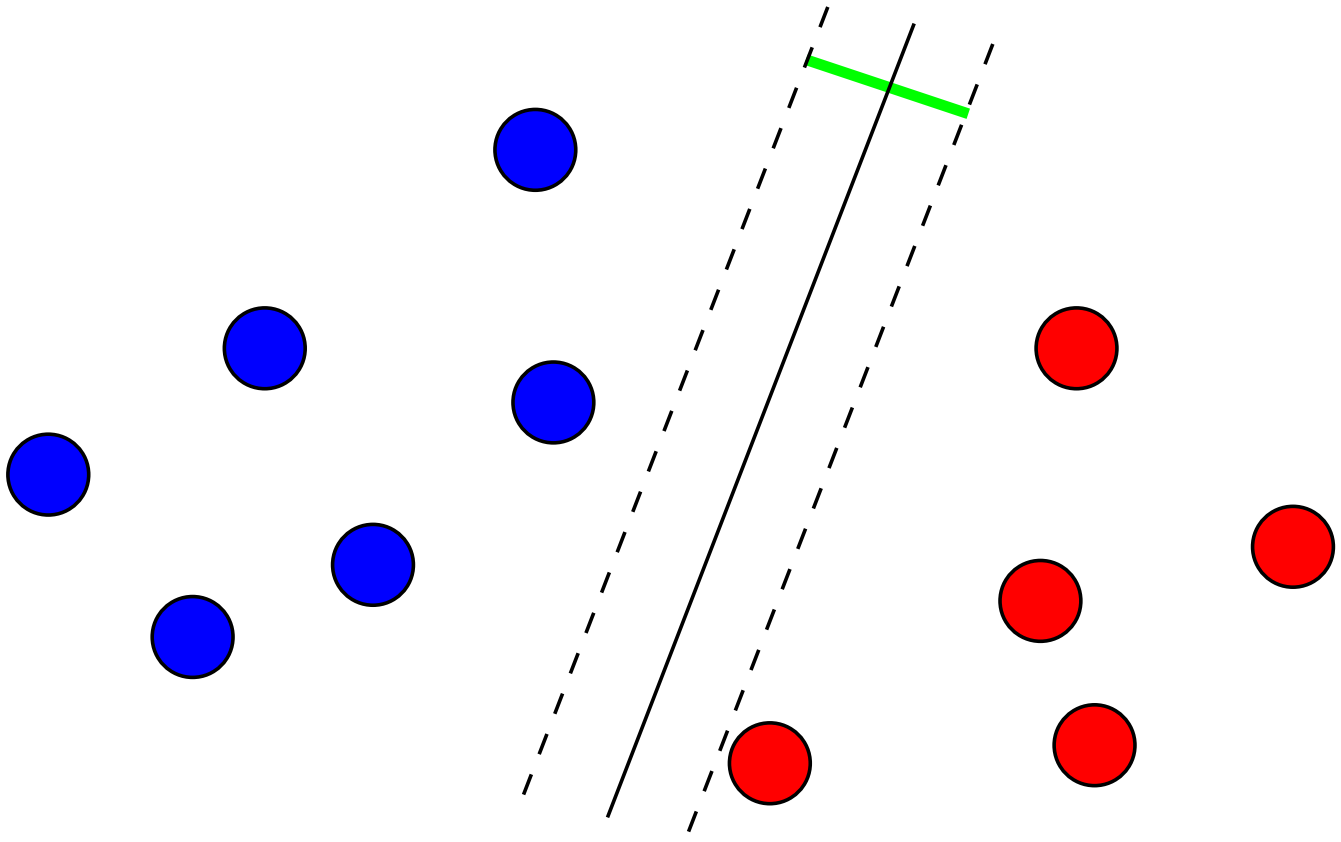
Which one is better?



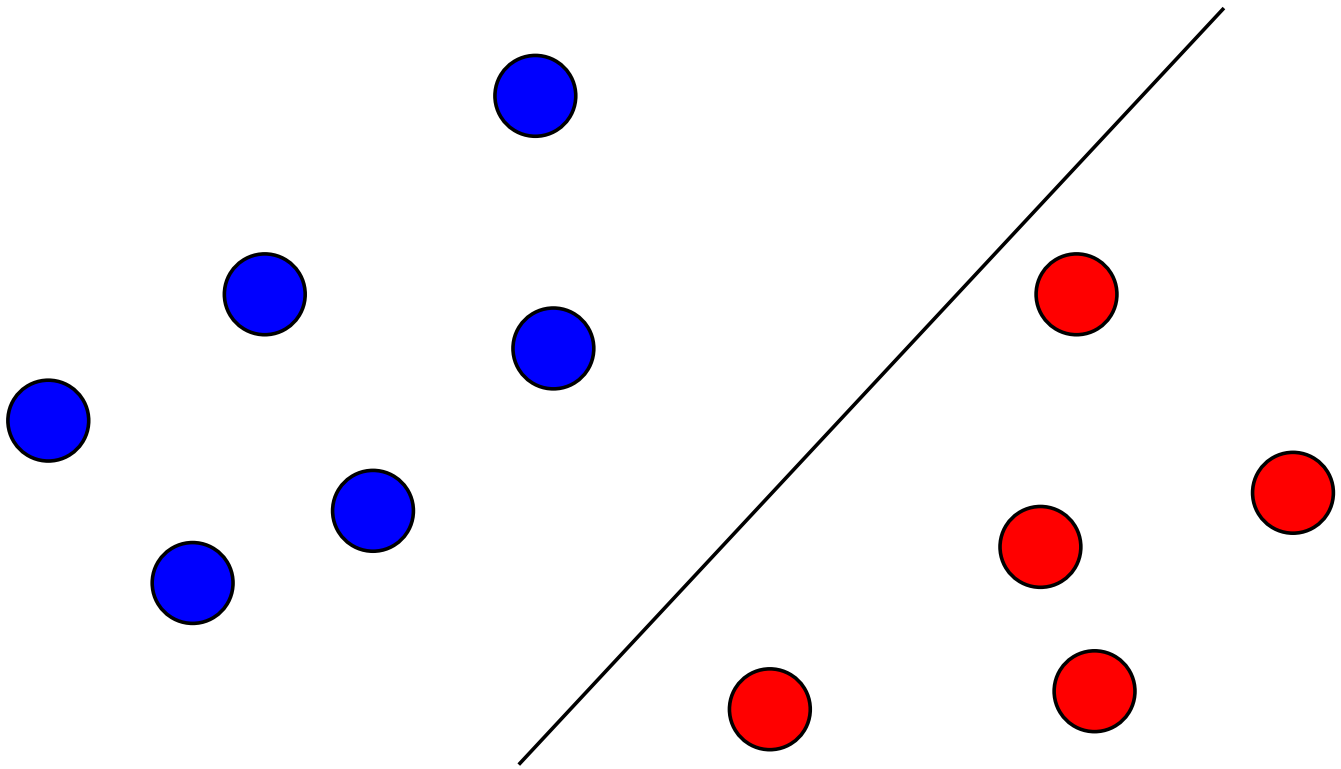
The margin of a linear classifier



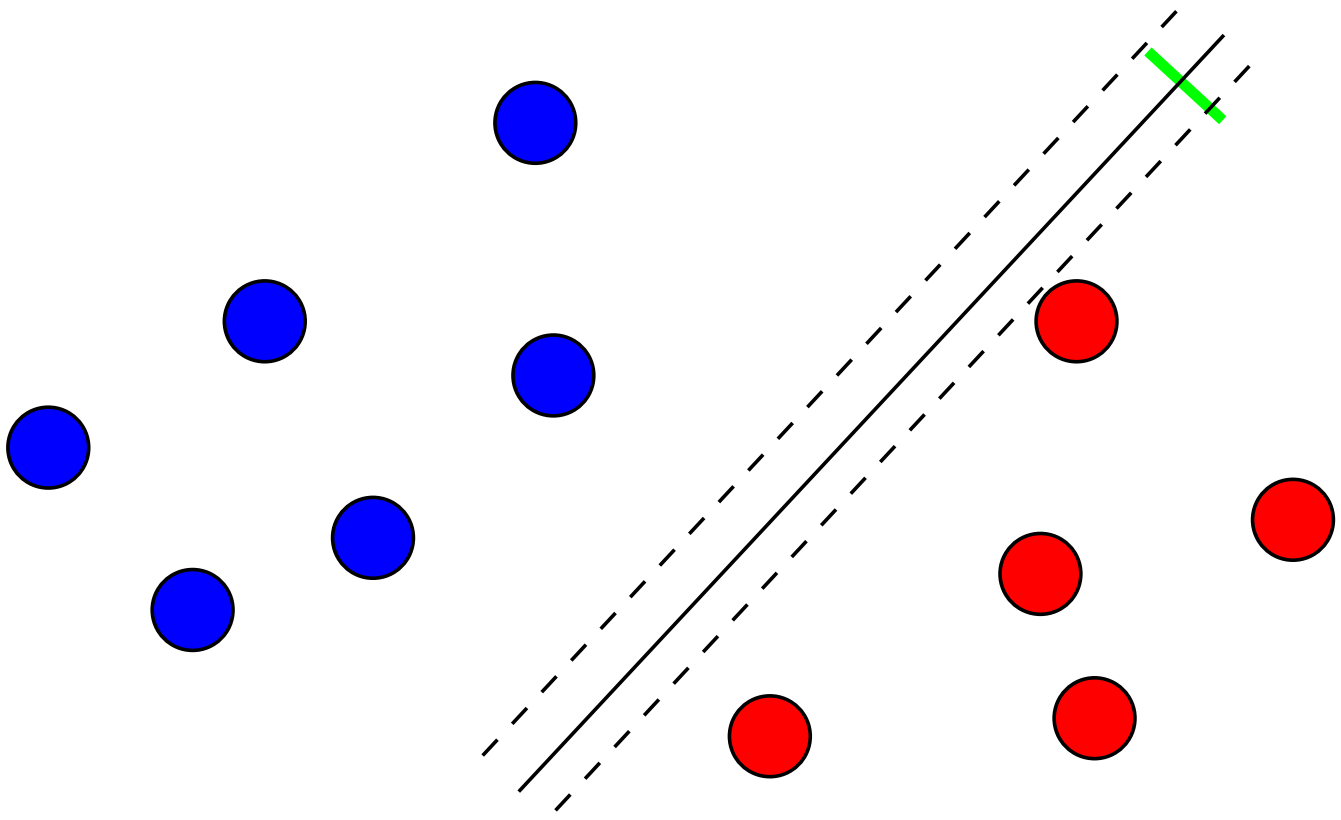
The margin of a linear classifier



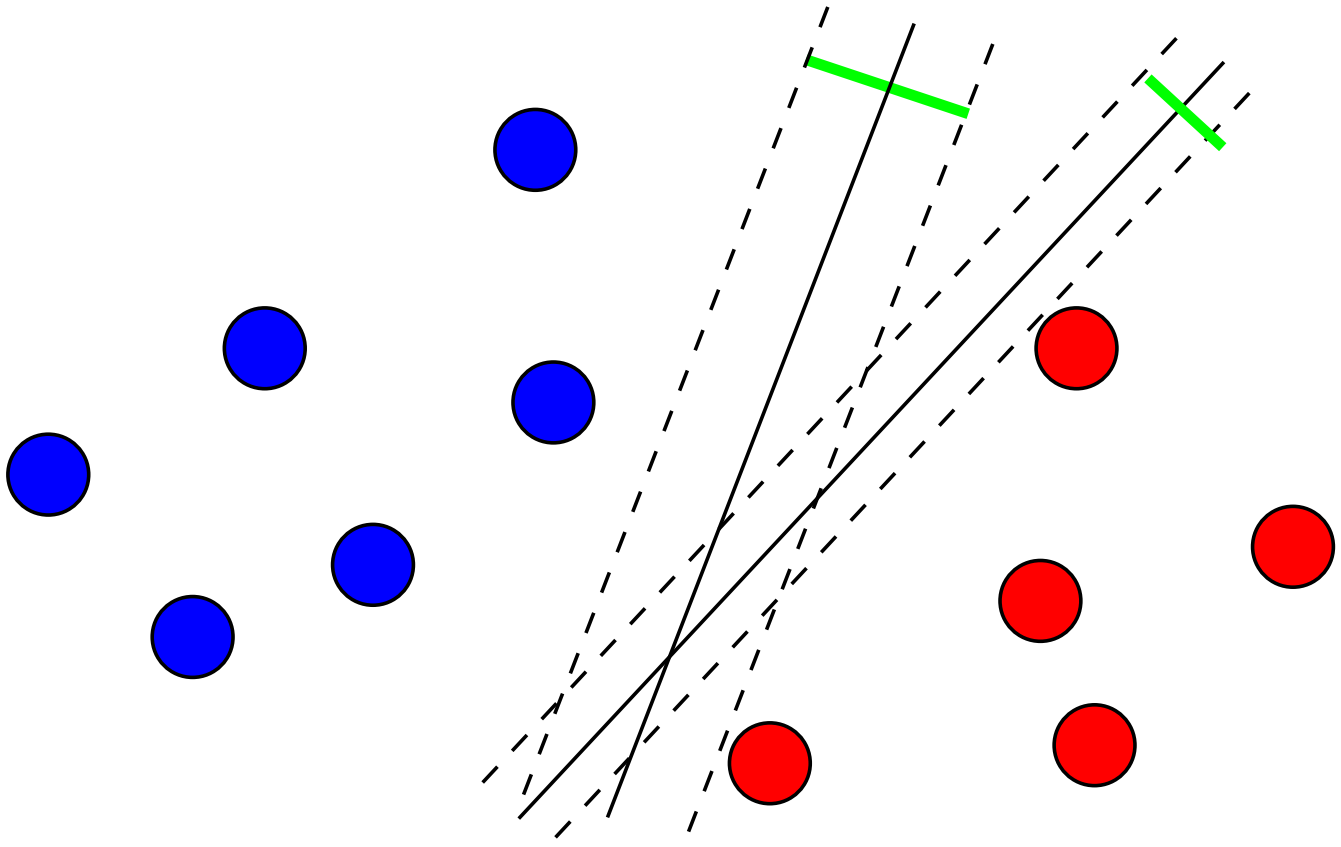
The margin of a linear classifier



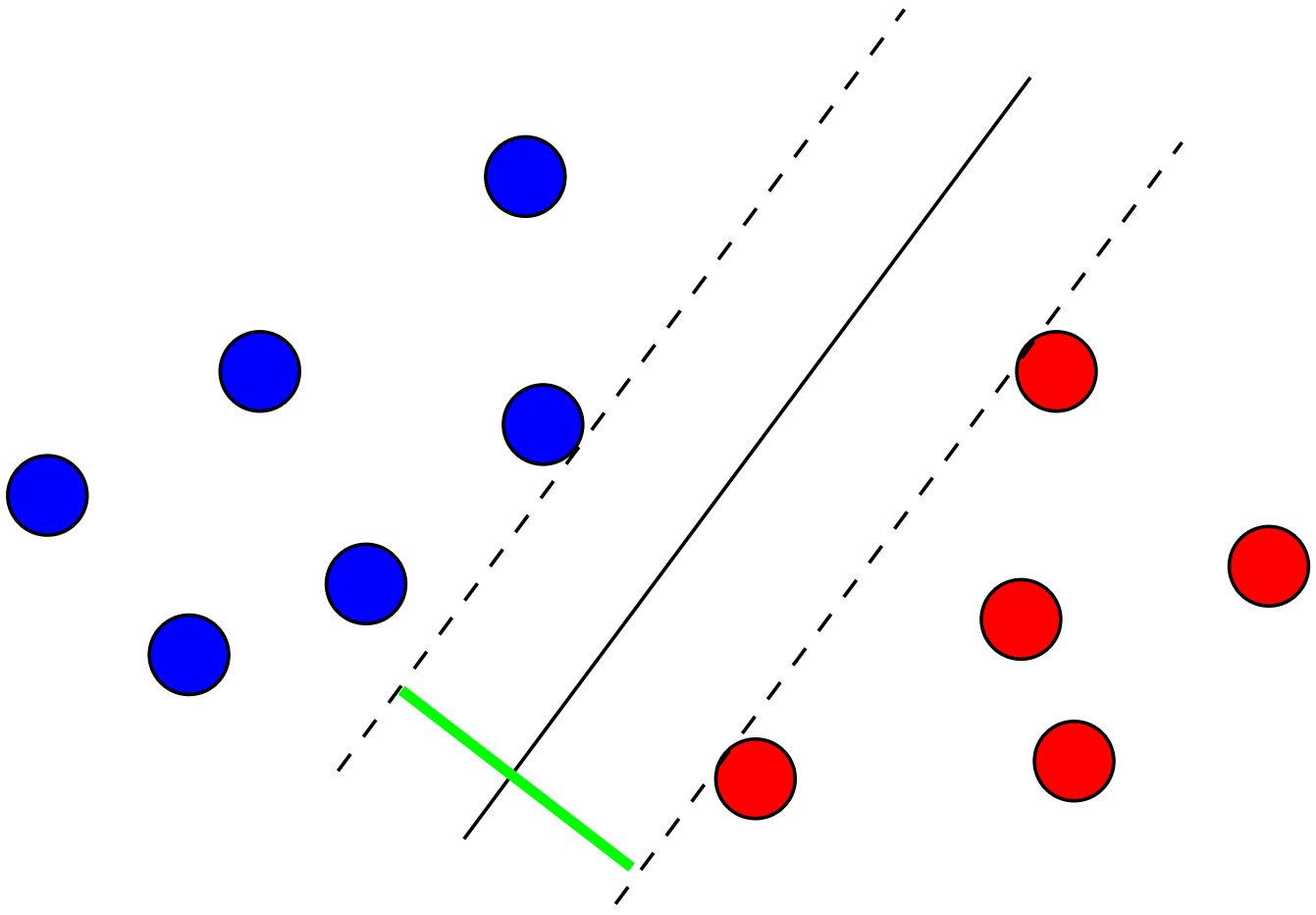
The margin of a linear classifier



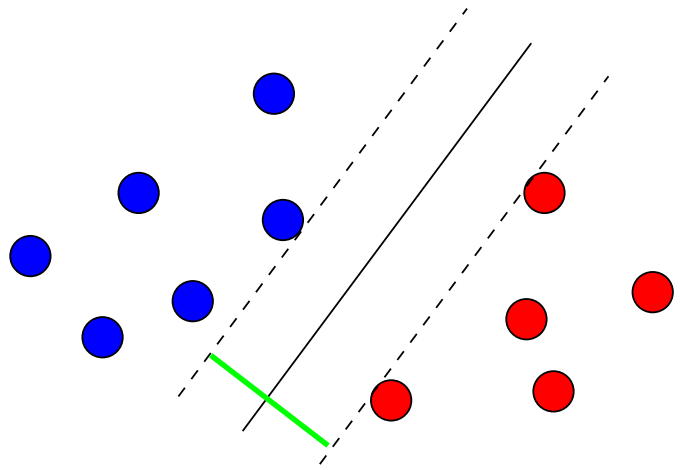
The margin of a linear classifier



Hard-margin SVM



Hard-margin SVM



Exercice

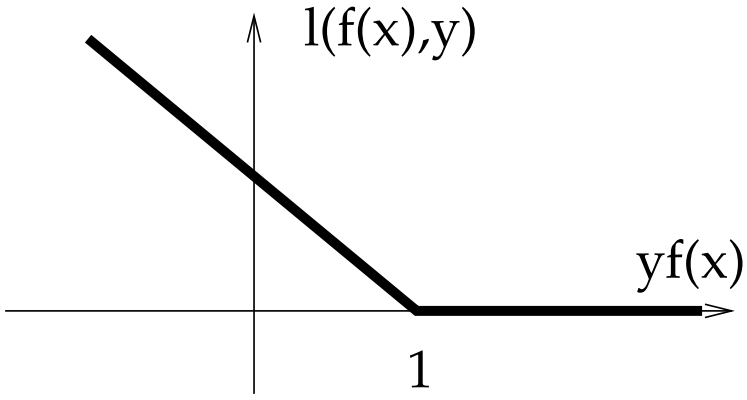
Show that hard-margin SVM solves a problem of the form:

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \ell_{HM-SVM}(f_\beta(x_i), y_i) \; + \lambda \|\beta\|_2^2 \, .$$

What is ℓ_{HM-SVM}?

Example: (soft-margin) SVM

- The hinge loss



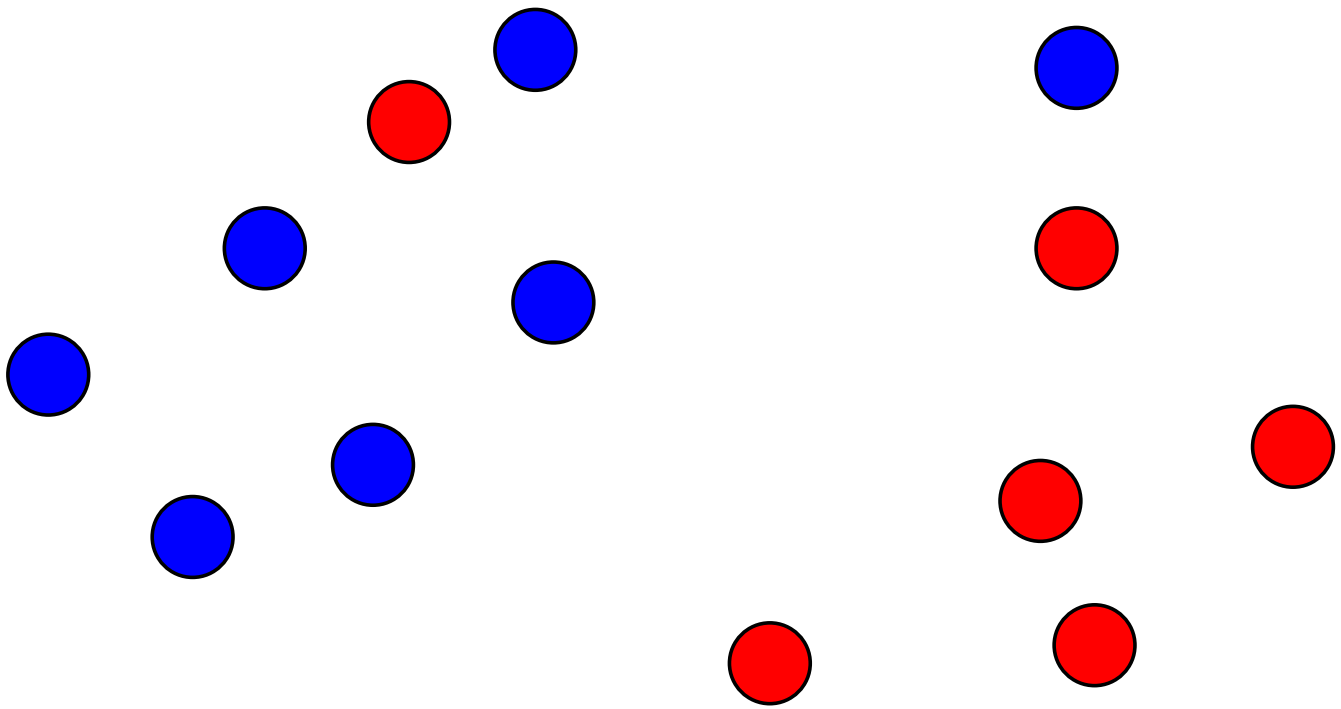
$$\phi_{\text{hinge}}(u) = \max(1-u, 0) = \begin{cases} 0 & \text{if } u \geq 1, \\ 1-u & \text{otherwise.} \end{cases}$$

- SVM solves:

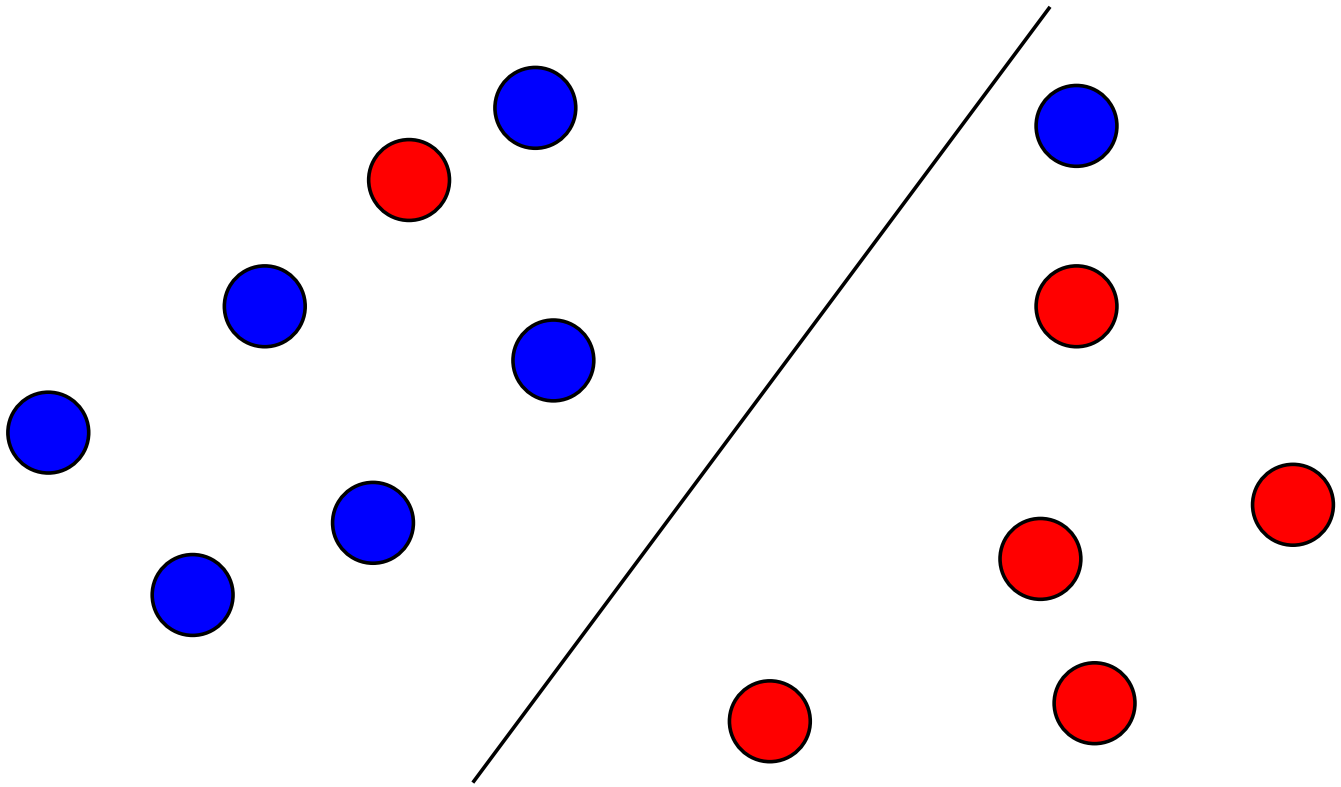
$$\min_{\beta} \left\{ \frac{1}{n} \sum_{i=1}^{n} \phi_{\text{hinge}}\left(\mathbf{y}_i f_\beta\left(\mathbf{x}_i\right)\right) + \lambda \| \beta \|_2^2 \right\} .$$

- No explicit solution. This is a convex but non-smooth optimization problem, equivalent to a quadratic program (QP) which can be solved efficiently.

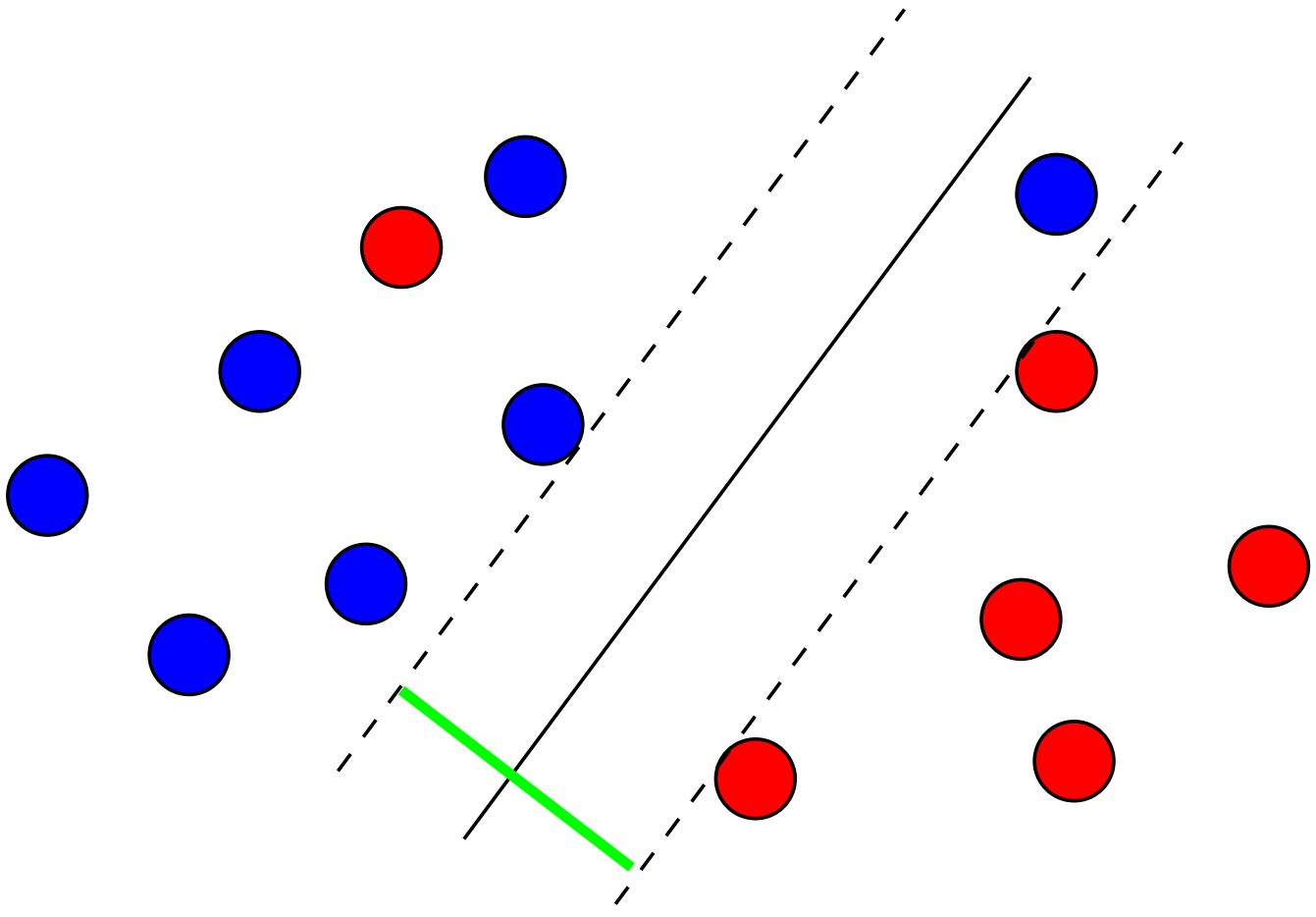
SVM: graphical interpretation



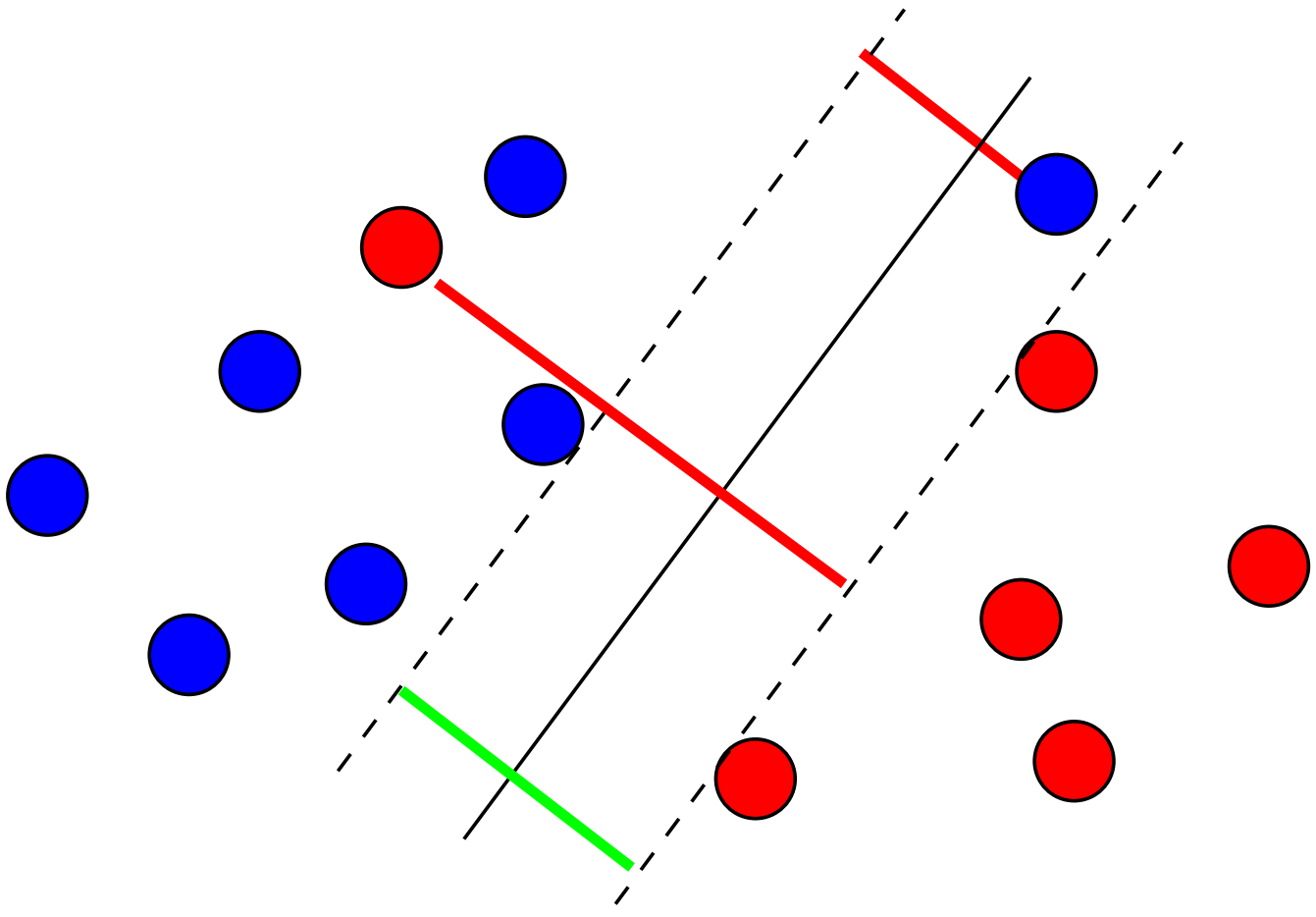
SVM: graphical interpretation



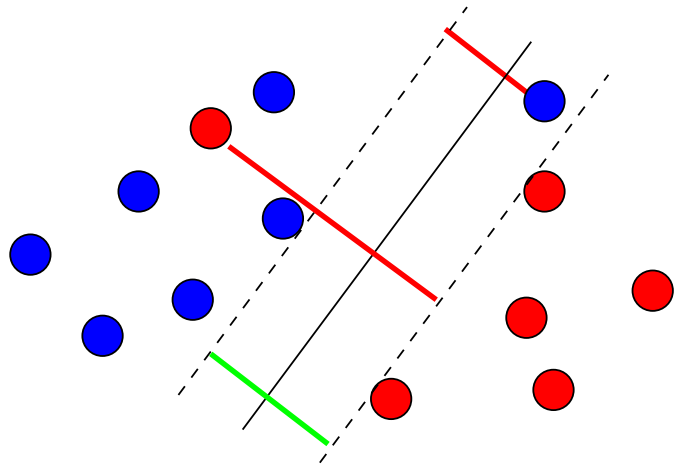
SVM: graphical interpretation



SVM: graphical interpretation



SVM: graphical interpretation



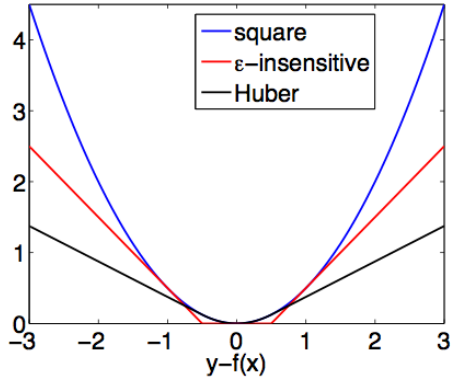
Exercice

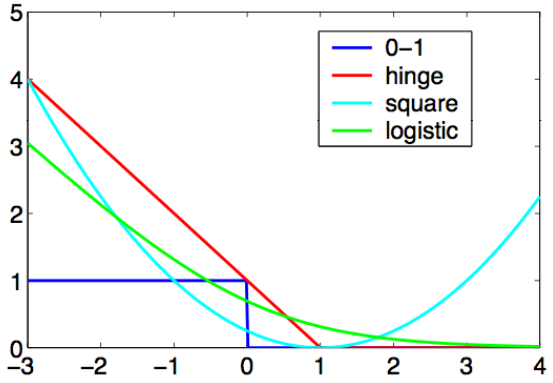
Show that SVM finds a trade-off between large margin and few errors, by minimizing a function of the form:

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

Explicit C and $error(f)$.

Summary: ℓ_2-regularize linear methods



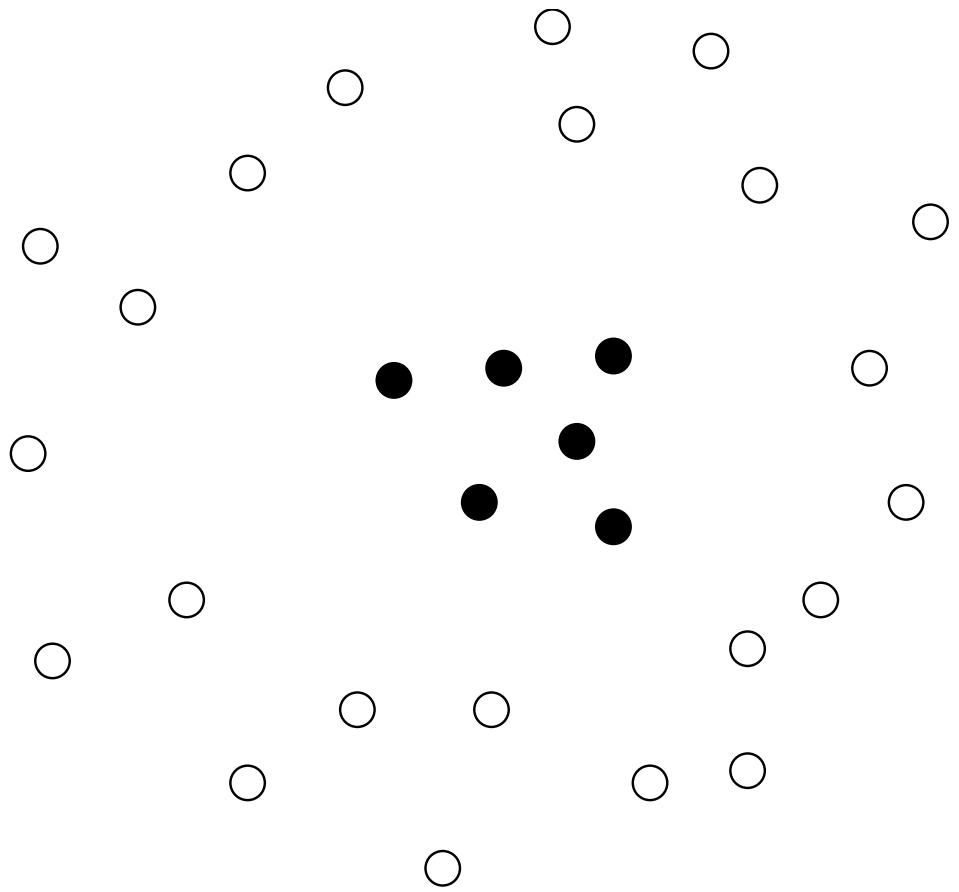


$$f_\beta(x) = \beta^\top x\,, \quad \min_\beta \frac{1}{n}\sum_{i=1}^{n} \ell(f_\beta(x_i), y_i) + \lambda\|\beta\|_2^2$$

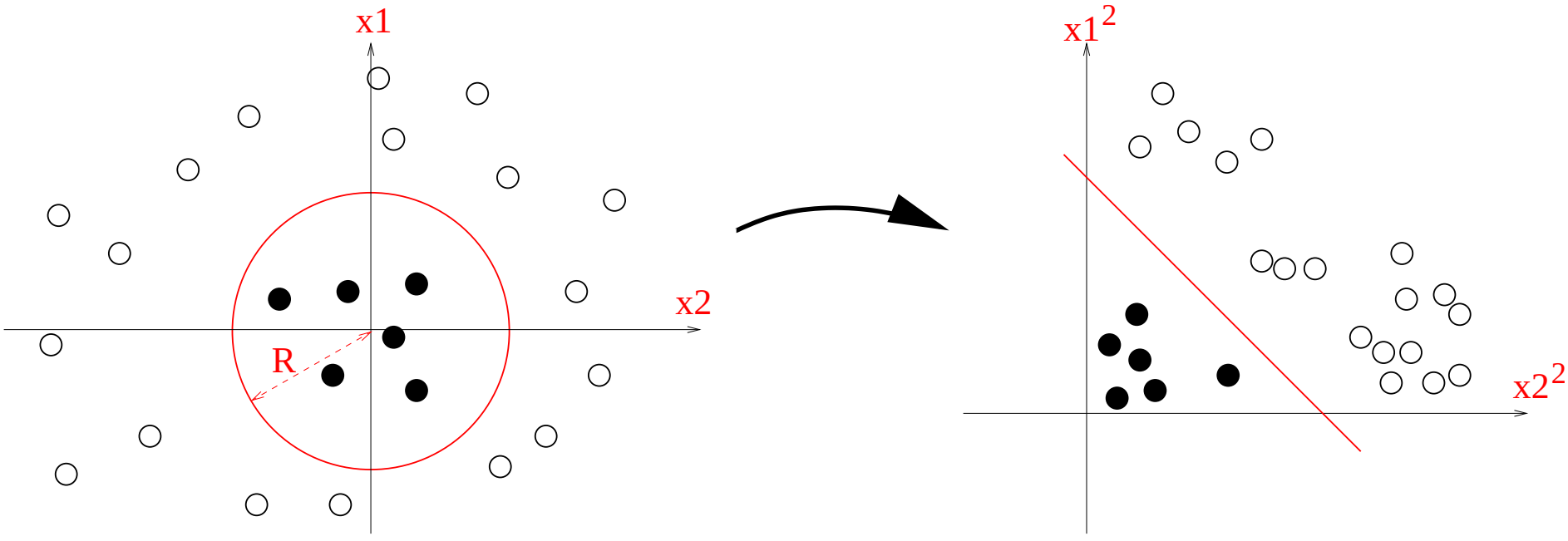
- Many popular methods for regression and classification are obtained by changing the loss function: ridge regression, logistic regression, SVM...
- Needs to solve numerically a convex optimization problem, well adapted to large datasets (stochastic gradient...)
- In practice, very similar performance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

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

Example: if $\vec{\Phi}(\vec{x}) = (x_1^2, x_2^2)'$, *then*

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

Representer theorem

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 $\|\beta\|^2 = \|\beta_{\mathcal{S}}\|^2 + \|\beta_{\perp}\|^2 \geq \|\beta_{\mathcal{S}}\|^2$, 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 $f_\beta(x) = \beta^\top \Phi(x)$ and K the corresponding kernel.
- By the representer theorem, any solution of:

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

can be expanded as:

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

Example: kernel ridge regression

- Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{R}^n$,
- Let K be the $n \times n$ Gram matrix: $K_{i,j} = K\left(\mathbf{x}_i, \mathbf{x}_j\right)$.
- We can then write in matrix form:

$$\left(\hat{f}\left(\mathbf{x}_1\right), \ldots, \hat{f}\left(\mathbf{x}_n\right)\right)^\top = K\boldsymbol{\alpha},$$

- The following holds as usual:

$$\| \beta \|_2^2 = \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}.$$

Example: kernel ridge regression

- The problem is therefore equivalent to:

$$\arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \frac{1}{N}(K\boldsymbol{\alpha}-y)^\top(K\boldsymbol{\alpha}-y)+\lambda\boldsymbol{\alpha}^\top K\boldsymbol{\alpha}.$$

- 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:

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

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 $(K + \lambda N I)^{-1}$ leaves $Im(K)$ and $Ker(K)$ invariant.
- The problem is therefore equivalent to:

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

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:

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

  with

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

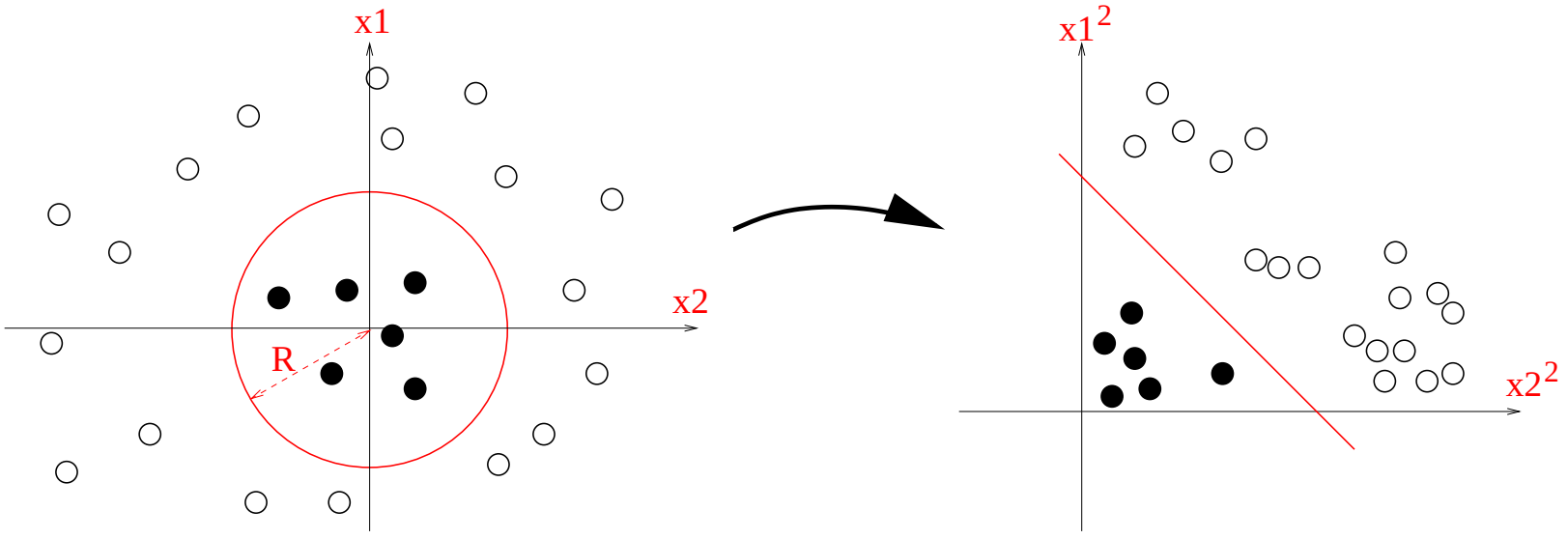
Example: kernel logistic regression of kernel SVM

- 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 $n < p$)

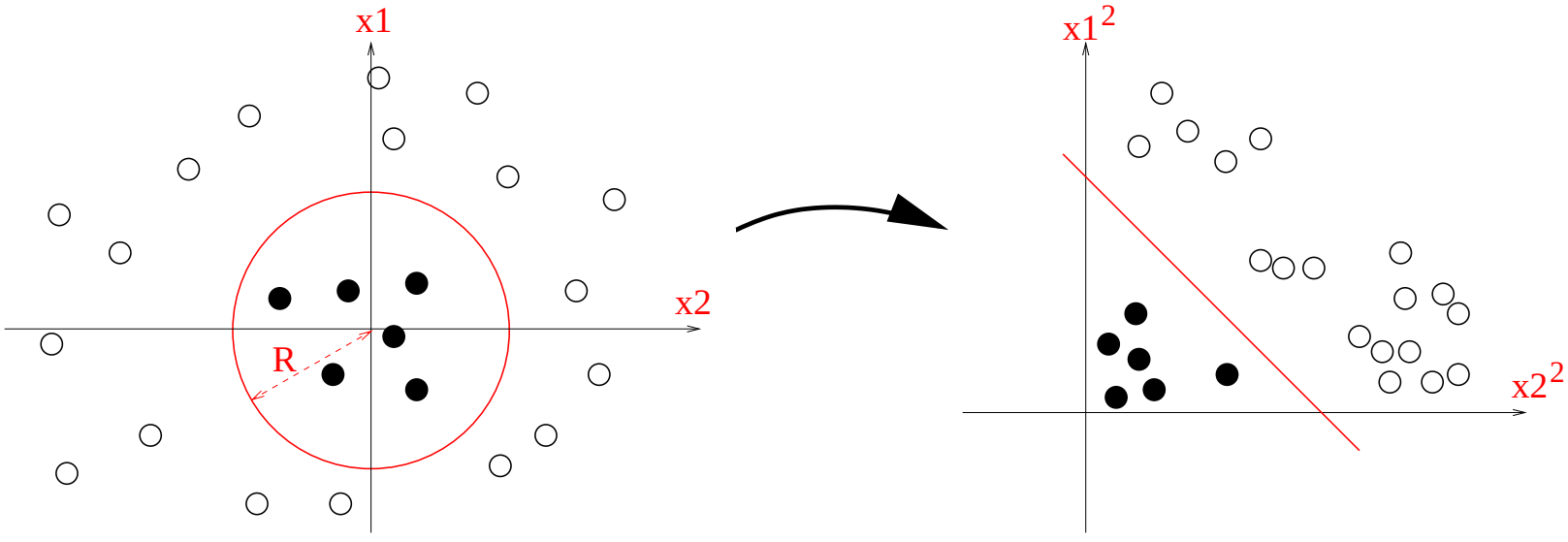
Kernel example: polynomial kernel



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

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

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?

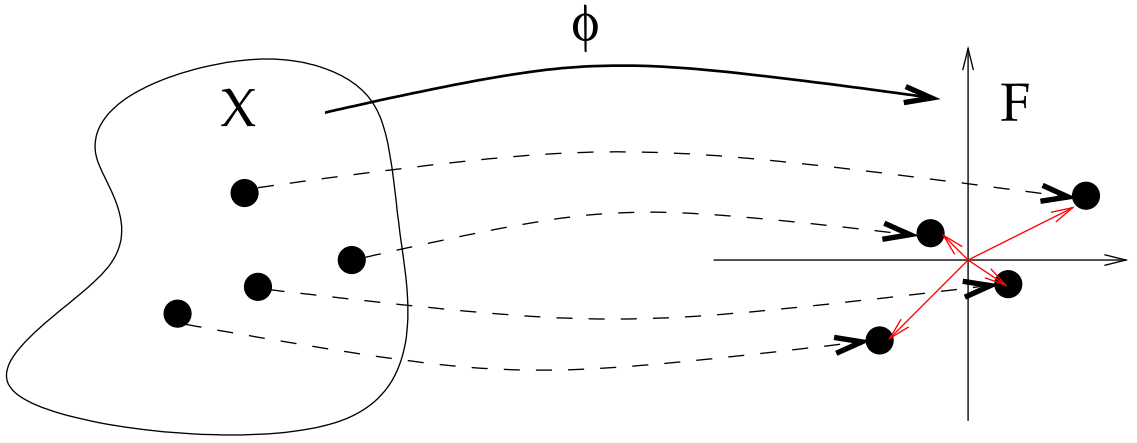
Definition

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

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



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:

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

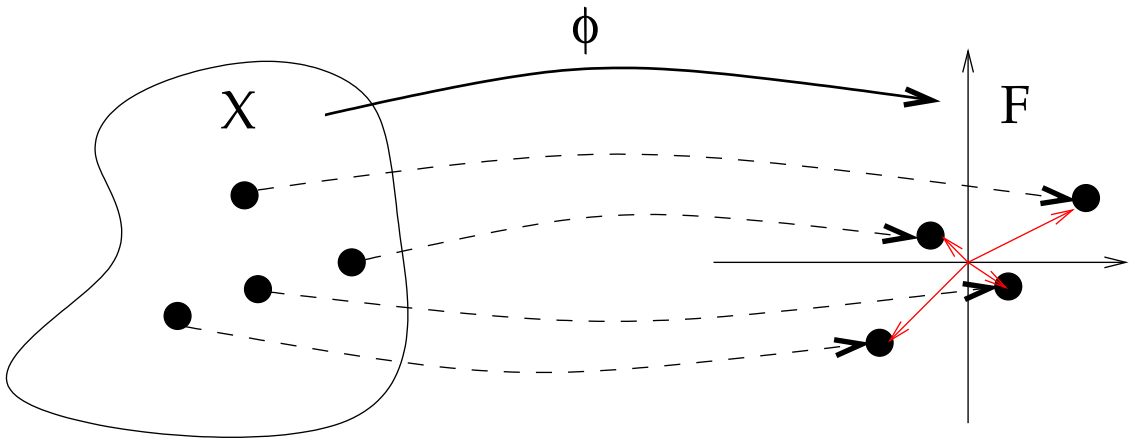
and which satisfies, for all $N \in \mathbb{N}$, $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \in \mathcal{X}^N$ et $(a_1, a_2, \ldots, a_N) \in \mathbb{R}^N$:

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

Kernels are p.d. functions

Theorem (Aronszajn, 1950)

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



Proof: kernel $\Longrightarrow$ p.d.

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

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

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

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

  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. $\Longrightarrow$ 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). $\square$

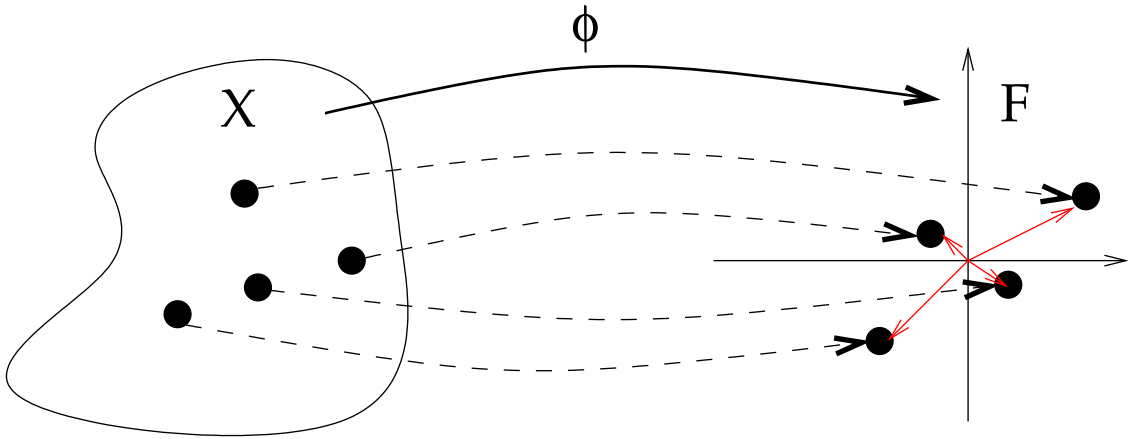
Proof: p.d. $\Longrightarrow$ kernel (5/5)

- 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 $\mathbb{R}_+$)

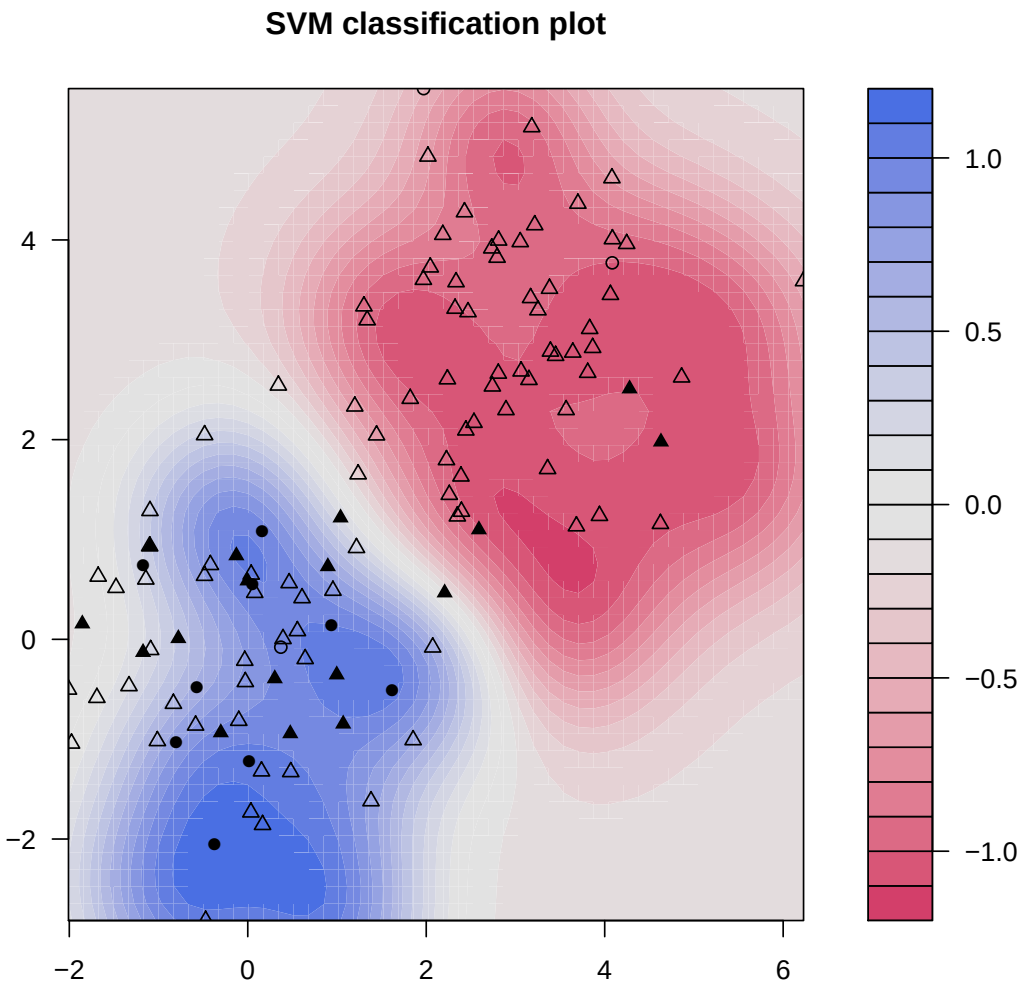
$$K(x, x') = \min(x, x')$$

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

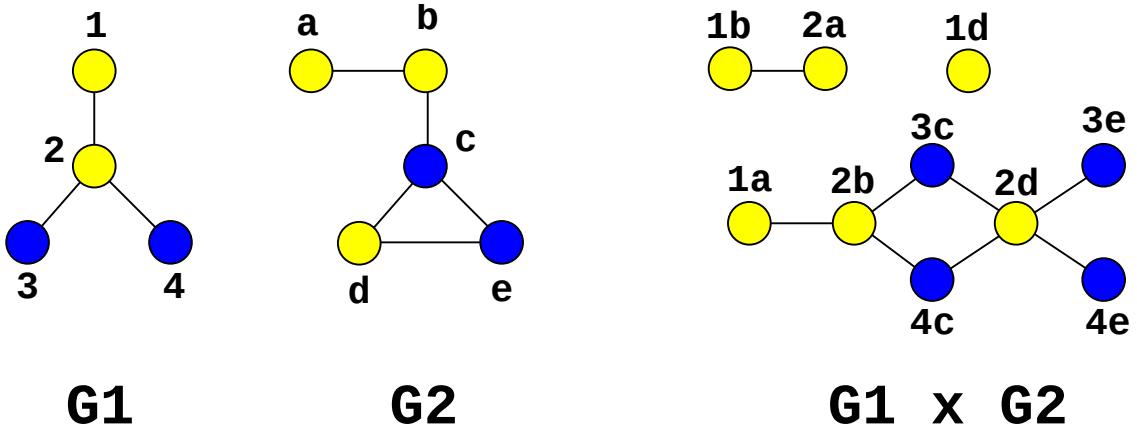
$$f(\vec{x}) = \sum_{i=1}^{N} \alpha_i \exp\left(-\frac{||\vec{x} - \vec{x}_i||^2}{2\sigma^2}\right)$$



How to choose or make a kernel?

- I don't really know...
- Design features?
- Adapt a distance or similarity measure?
- Design a regularizer on f?

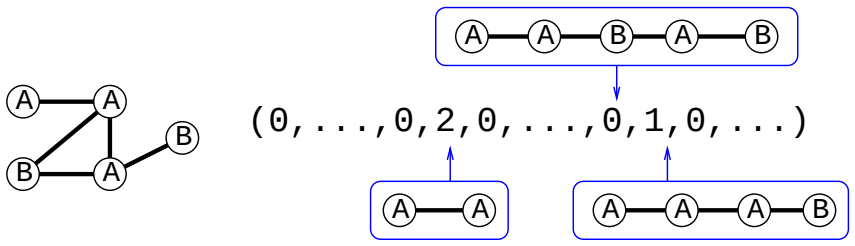
Example: design features (Gärtner et al., 2003)



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

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
```

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

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

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

Example: design a regularizer

- Remember $f_\beta(x) = x^\top \Phi(x)$, the regularizer is $\Omega(f_\beta) = \|\beta\|^2$
- Regularize in the Fourier domain:

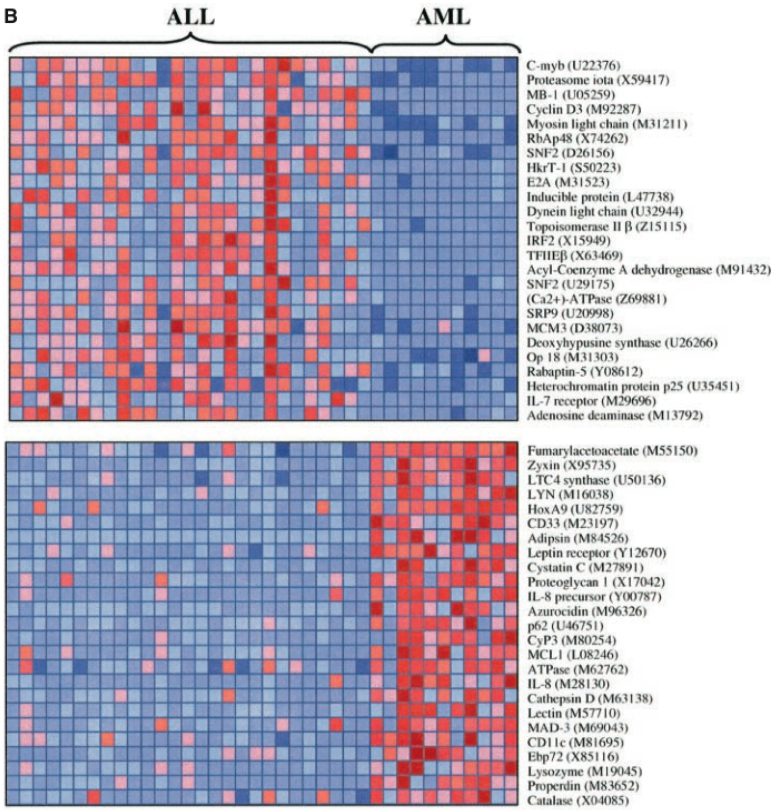
$$\Omega(f) = \int \|\hat{f}(\omega)\|^2 \exp \frac{\sigma^2\omega^2}{2} d\omega \qquad K(x,y) = \exp\left(-\frac{(x-y)^2}{2\sigma^2}\right)$$

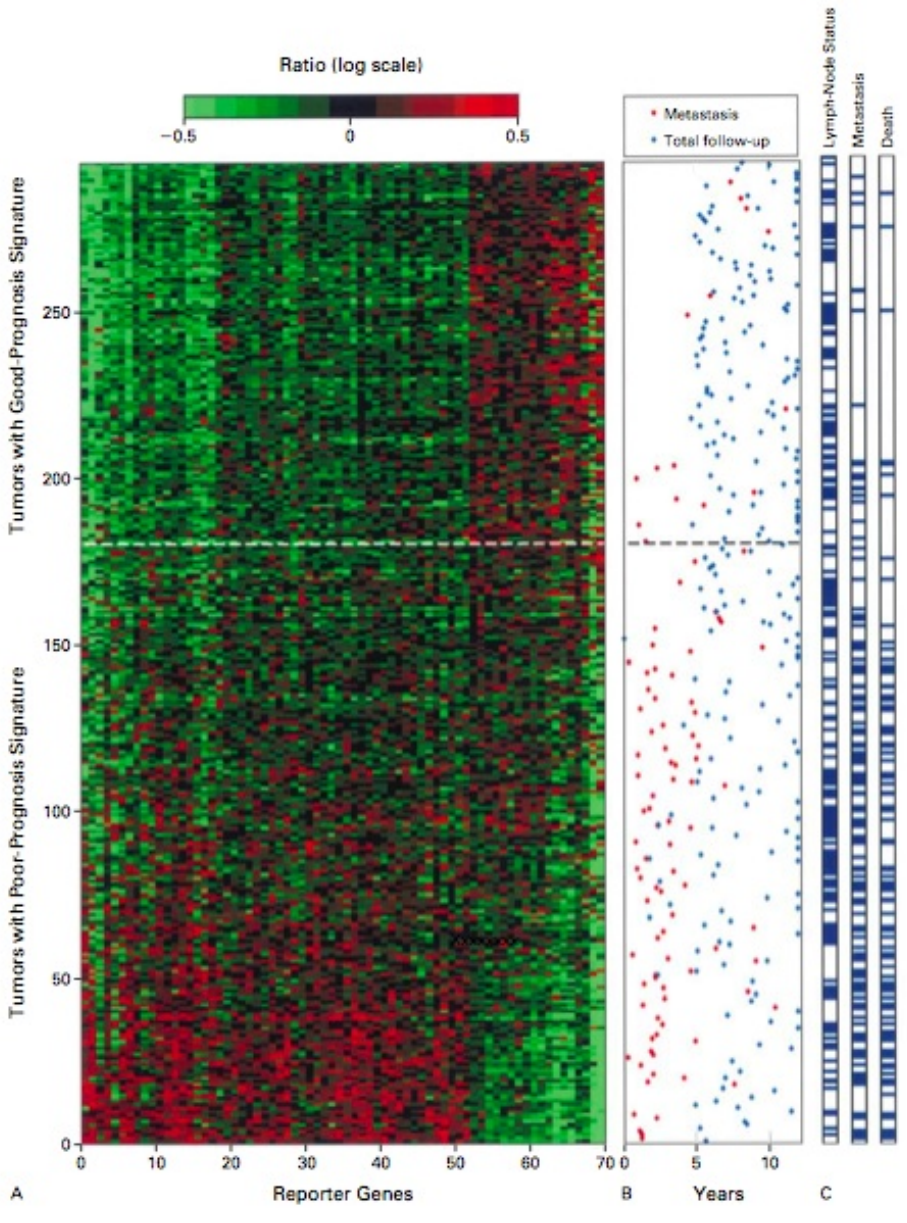
- Sobolev norms

$$\Omega(f) = \int_0^1 f'(u)^2 du \qquad K(x,y) = \min(x,y)$$

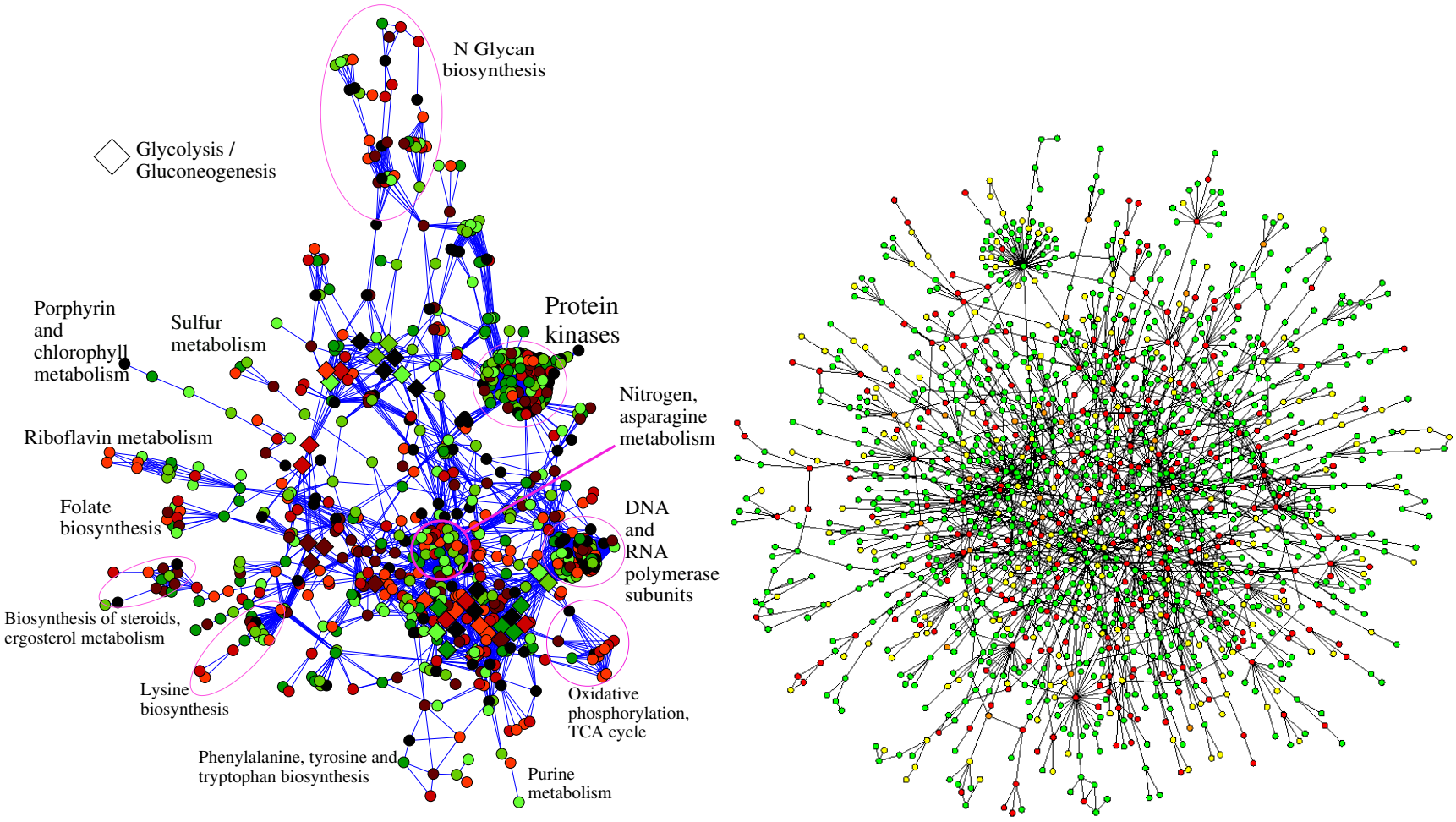
Outline

Molecular diagnosis / prognosis / theragnosis





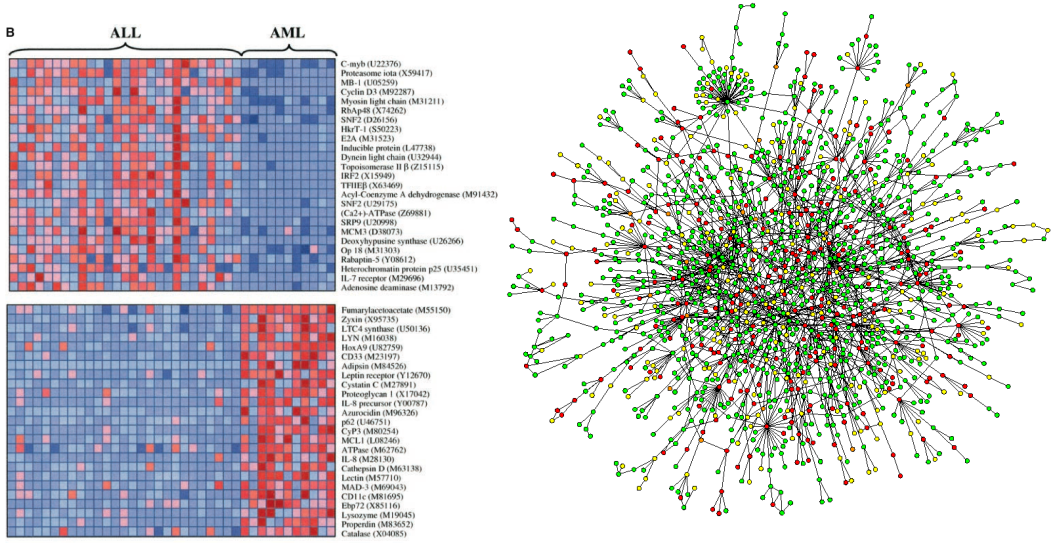
Gene networks



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)

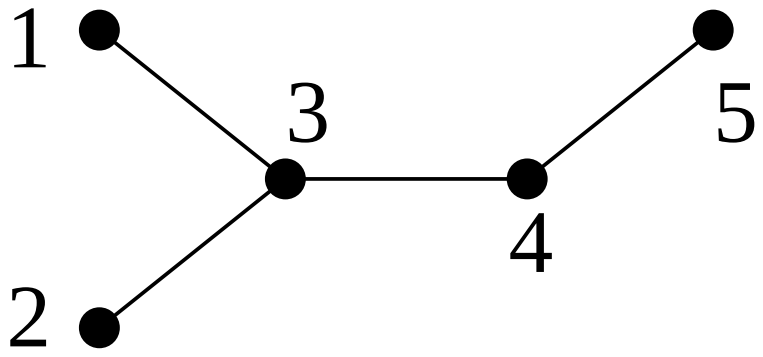
$$\Omega(\beta) = \sum_{i\sim j}(\beta_i - \beta_j)^2 \,,$$

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

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

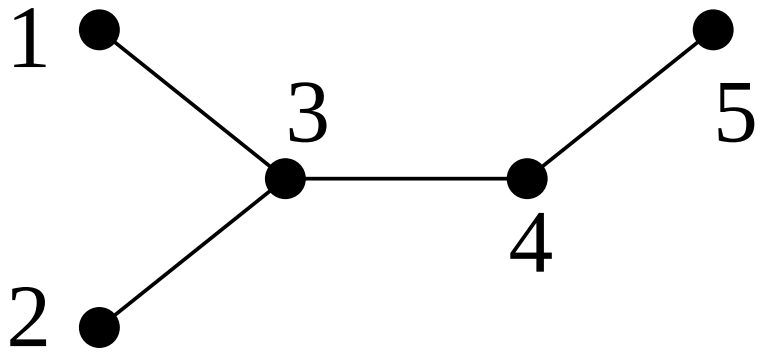
$$\min_{\gamma \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \ell\left(\gamma^\top \Phi(x_i), y_i\right) + \lambda \gamma^\top \gamma \,,$$

and where

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

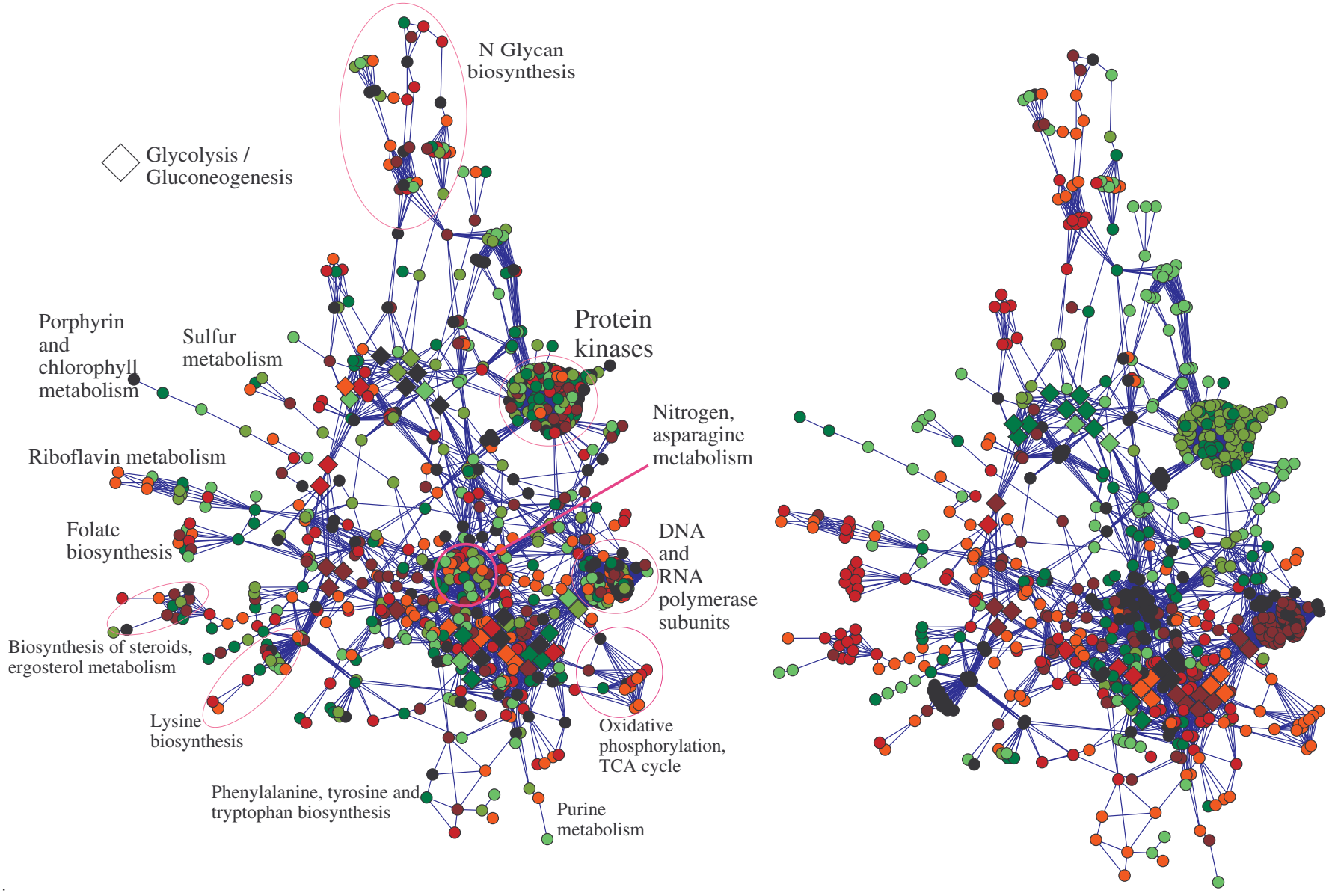
for $K_G = L^*$, the pseudo-inverse of the graph Laplacian.

Example

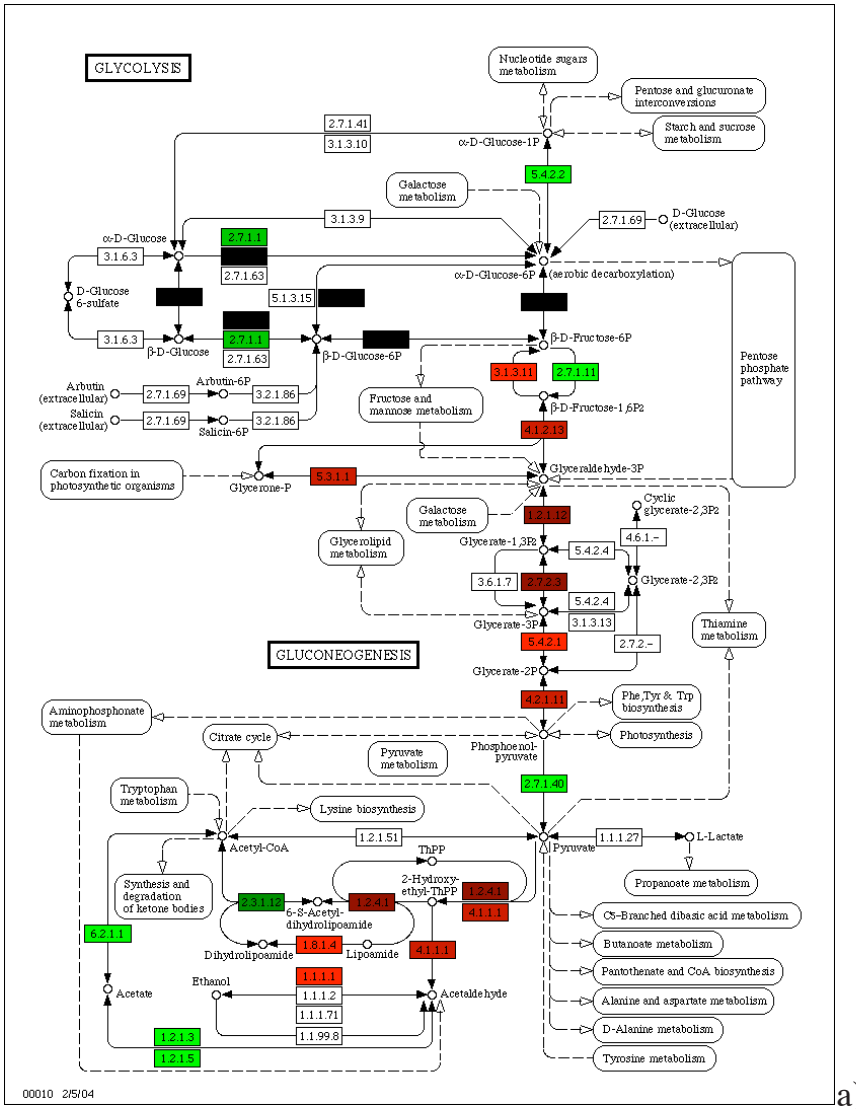


$$L^* = \begin{pmatrix} 0.88 & -0.12 & 0.08 & -0.32 & -0.52 \\ -0.12 & 0.88 & 0.08 & -0.32 & -0.52 \\ 0.08 & 0.08 & 0.28 & -0.12 & -0.32 \\ -0.32 & -0.32 & -0.12 & 0.48 & 0.28 \\ -0.52 & -0.52 & -0.32 & 0.28 & 1.08 \end{pmatrix}$$

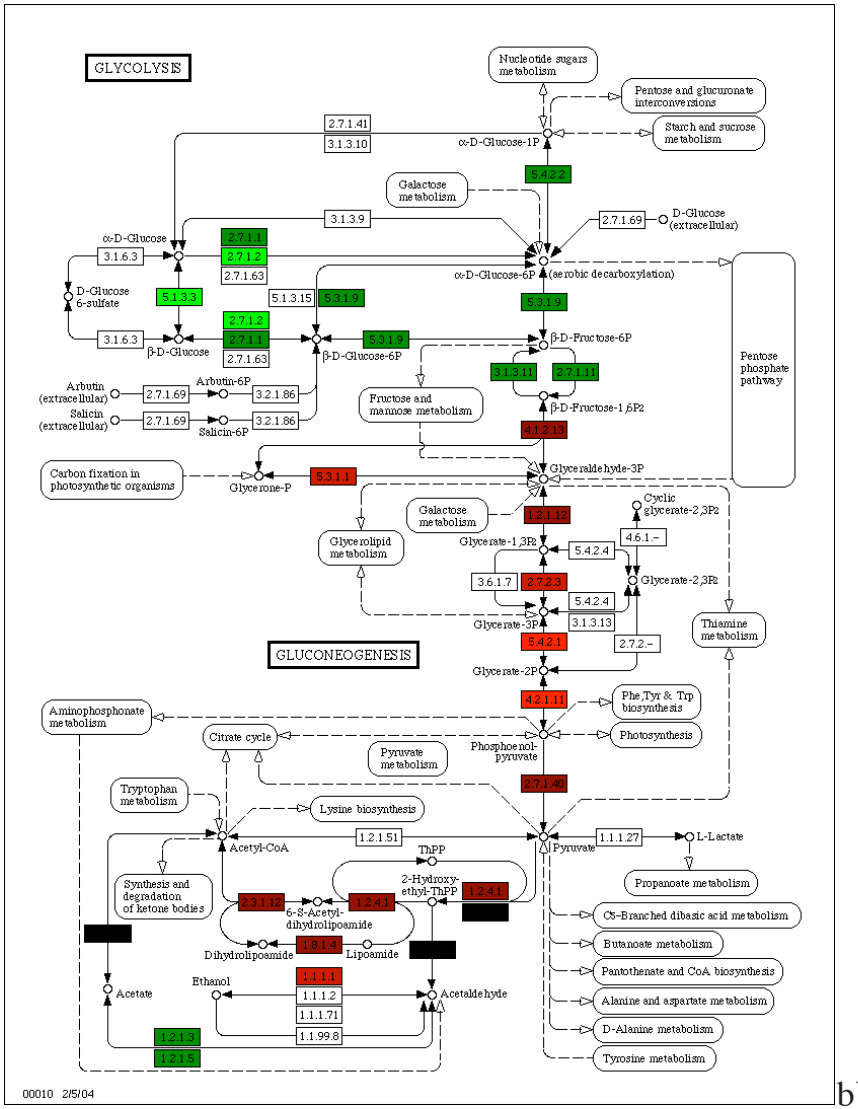
Classifiers



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

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

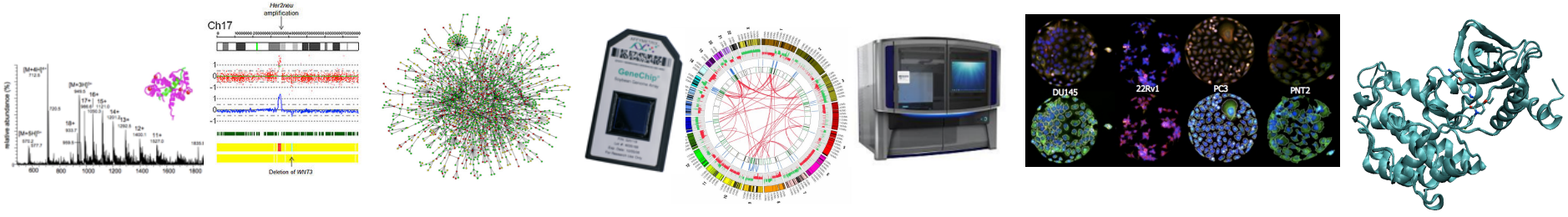
- The diffusion kernel:

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

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

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

- 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:

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

The sum kernel

- Let $K_1, \ldots, K_M$ be M kernels corresponding to M sources of data
- Summing the kernel together defines a new "integrated" kernel

Theorem

Learning with $K = \sum_{i=1}^{M} K_i$ is equivalent to work with a feature vector $\Phi(x)$ obtained by concatenation of $\Phi_1(x), \ldots, \Phi_M(x)$. It solves the following problem:

$$\min_{f_{\beta_1}, \ldots, f_{\beta_M}} R\left(\sum_{i=1}^{M} f_{\beta_i}\right) + \lambda \sum_{i=1}^{M} \|\beta_i\|^2$$

Proof left as exercise.

Example: protein network inference

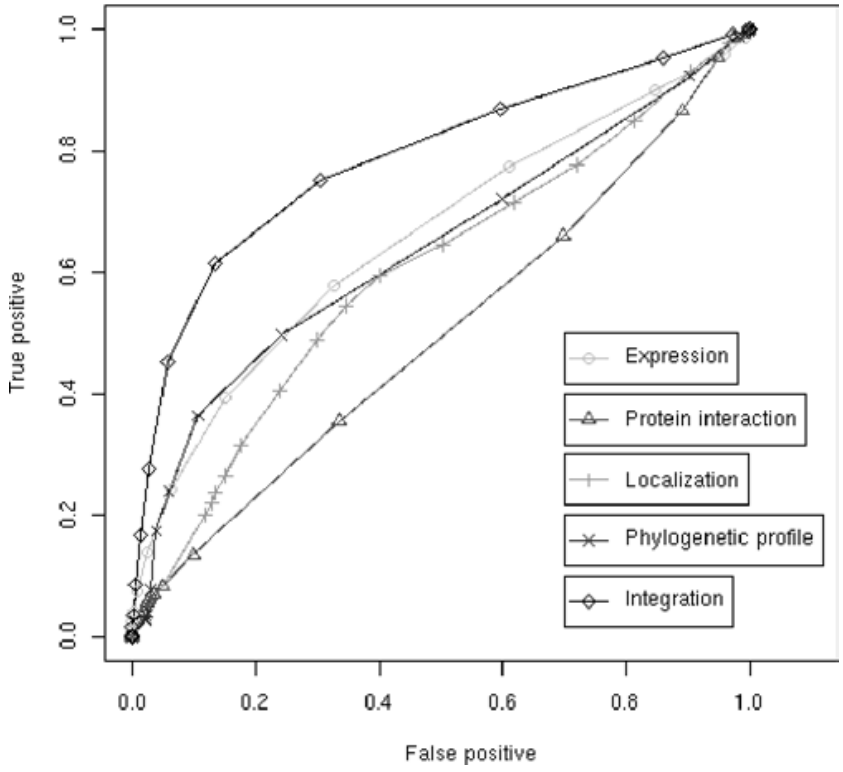
BIOINFORMATICS

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)
$K_{\mathrm{exp}} + K_{\mathrm{ppi}} + K_{\mathrm{loc}} + K_{\mathrm{phy}}$ (Integration)

Multiple kernel learning (Lanckriet et al., 2004)

- Perhaps a more clever approach is to learn a weighted linear combination of kernels:

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

- MKL learns the weights with the predictor by solving:

$$\min_{\eta, \boldsymbol{\alpha}} J_{K_\eta}(\boldsymbol{\alpha}) \quad \text{such that} \quad \mathit{Trace}(K_\eta) = 1 \,.$$

- 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.

Example: protein annotation

BIOINFORMATICS

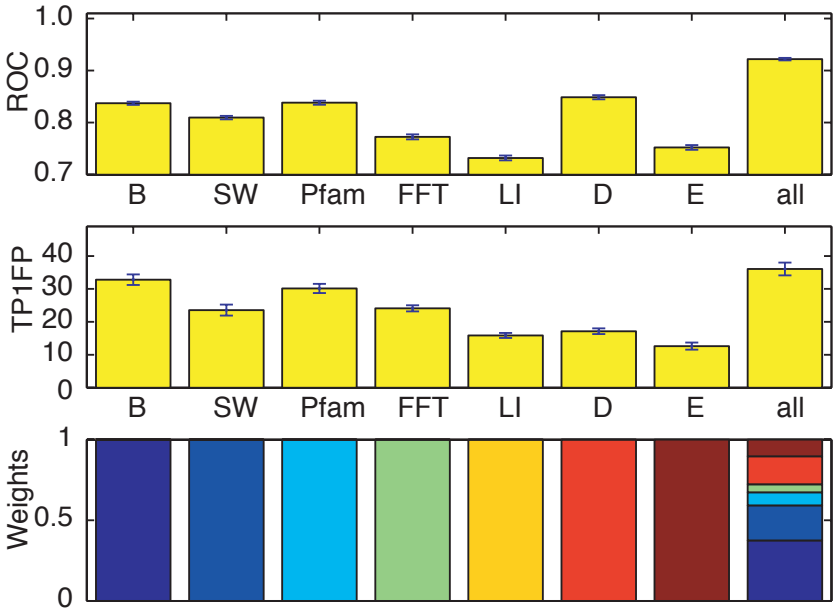
Vol. 20 no. 16 2004, pages 2626–2635
doi:10.1093/bioinformatics/bth294

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

MKL revisited

Theorem (Bach et al., 2004)

MKL solves the following problem:

$$\min_{f_{\beta_1},\ldots,f_{\beta_M}} R\left(\sum_{i=1}^{M} f_{\beta_i}\right) + \lambda \sum_{i=1}^{M} \|\beta_i\|$$

- This is an instance of (kernelized) group lasso (more later...)
- This promotes sparsity at the kernel level
- MKL is mostly useful if only a few kernels are relevant; otherwise the sum kernel may be a better option.

Outline

Outline

Motivation

- In feature selection, we look for a linear function $f(\mathbf{x}) = \mathbf{x}^\top \beta$, where only a limited number of coefficients in β are non-zero.
- Motivations
  - Accuracy: by imposing a constraint on β, we increase the bias but decrease the variance. This should be helpful in particular in high dimension.
  - Interpretation: simpler to understand and communicate a sparse model.
  - Implementation: a device based on a few markers can be cheaper and faster.

Of course, this is particularly relevant if we believe that there exist good predictors which are sparse (prior knowledge).

Best subset selection

$$\Omega(\beta) = \|\beta\|_0 = \text{number of non-zero coefficients}$$

- In best subset selection, we must solve the problem:

$$\min R(f_\beta) \quad \text{s.t.} \quad \| \beta \|_0 \leq k$$

  for $k = 1, \ldots, p$.
- The state-of-the-art is branch-and-bound optimization, known as *leaps and bound* for least squares (Furnival and Wilson, 1974).
- This is usually a NP-hard problem, feasible for p as large as 30 or 40

Efficient feature selection

To work with more variables, we must use different methods. The state-of-the-art is split among

- Filter methods : the predictors are preprocessed and ranked from the most relevant to the less relevant. The subsets are then obtained from this list, starting from the top.
- Wrapper method: here the feature selection is iterative, and uses the ERM algorithm in the inner loop
- Embedded methods : here the feature selection is part of the ERM algorithm itself (see later the shrinkage estimators).

Filter methods

- Associate a score $S(i)$ to each feature i, then rank the features by decreasing score.
- Many scores / criteria can be used
  - Loss of the ERM trained on a single feature
  - Statistical tests (Fisher, T-test)
  - Other performance criteria of the ERM restricted to a single feature (AUC, ...)
  - Information theoretical criteria (mutual information...)

Pros

Simple, scalable, good empirical success

Cons

- Selection of redundant features
- Some variables useless alone can become useful together

Filter methods

- Associate a score $S(i)$ to each feature i, then rank the features by decreasing score.
- Many scores / criteria can be used
  - Loss of the ERM trained on a single feature
  - Statistical tests (Fisher, T-test)
  - Other performance criteria of the ERM restricted to a single feature (AUC, ...)
  - Information theoretical criteria (mutual information...)

Pros

Simple, scalable, good empirical success

Cons

- Selection of redundant features
- Some variables useless alone can become useful together

Filter methods

- Associate a score $S(i)$ to each feature i, then rank the features by decreasing score.
- Many scores / criteria can be used
  - Loss of the ERM trained on a single feature
  - Statistical tests (Fisher, T-test)
  - Other performance criteria of the ERM restricted to a single feature (AUC, ...)
  - Information theoretical criteria (mutual information...)

Pros

Simple, scalable, good empirical success

Cons

- Selection of redundant features
- Some variables useless alone can become useful together

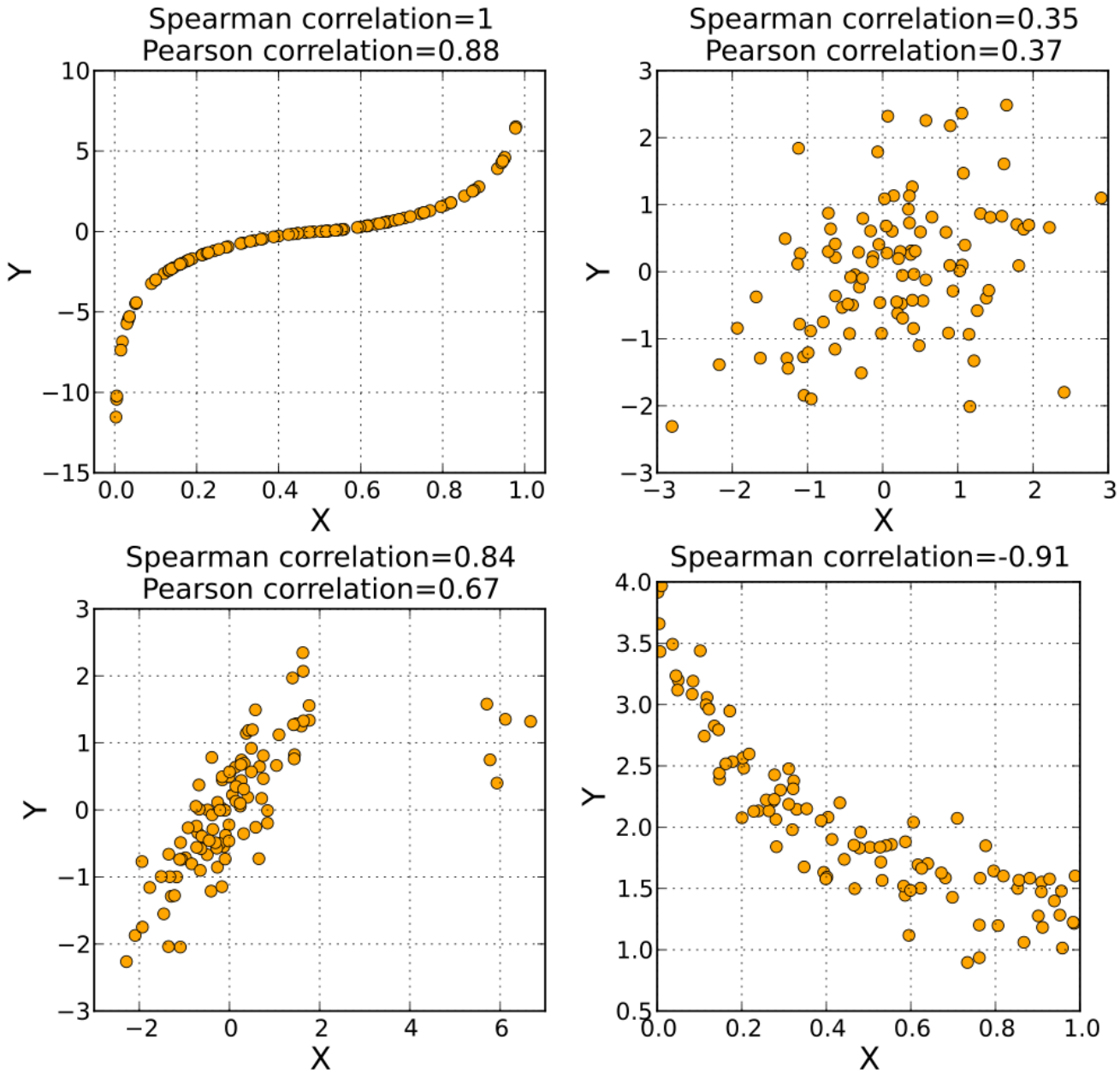
Measuring dependency: correlation coefficients

- Assume X and Y take continuous values
- $(X_1, Y_1), \ldots, (X_n, Y_n)$ the n expression values of both genes
- Pearson correlation:

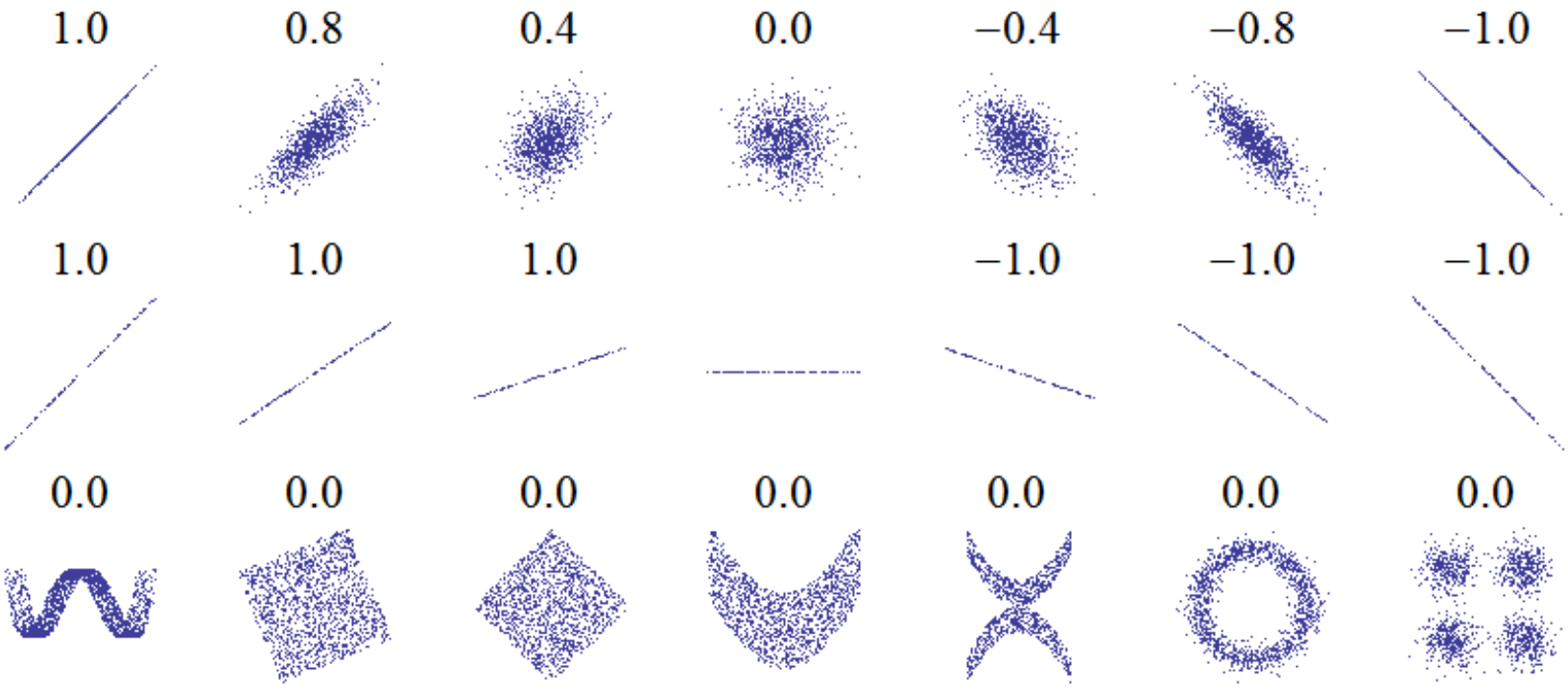
$$\rho = \frac{cov(X, Y)}{\sigma_X \sigma_Y} = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2} \sqrt{\sum_i (Y_i - \bar{Y})^2}}$$

- Spearman correlation: similar but replace X_i by its rank.

Illustration



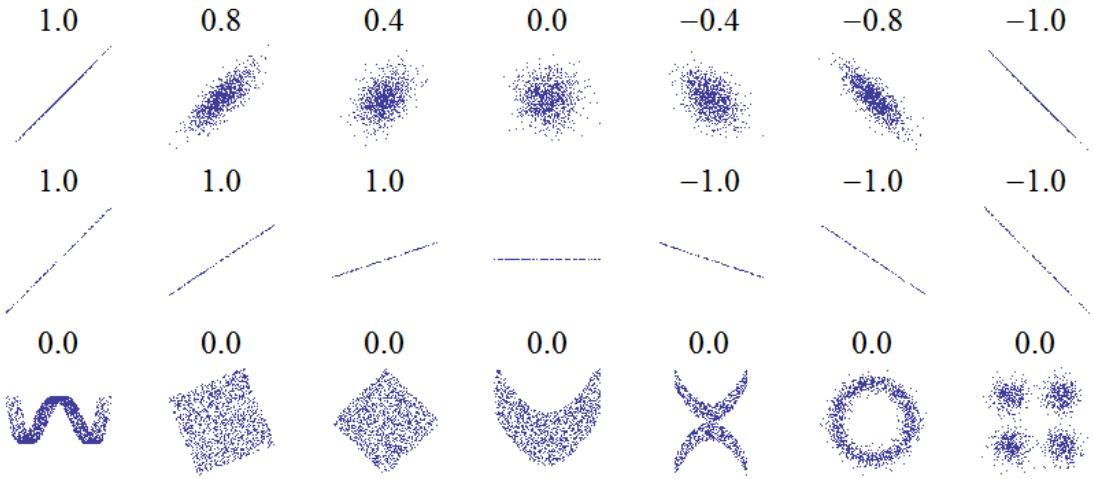
Limit of correlations



Mutual information

$$I(X;Y) = \int_Y \int_X p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) dxdy$$

- $I(X;Y) \geq 0$
- $I(X;Y) = 0$ if and only if X and Y are independent



Wrapper methods

The idea

- A greedy approach to

$$\min R(f_\beta) \quad \text{s.t.} \quad \| \beta \|_0 \leq k$$

- For a given set of seleted features, we know how to minimize $R(f)$
- We iteratively try to find a good set of features, by adding/removing features which contribute most to decrease the risk (using ERM as an internal loop)

Two flavors of wrapper methods

Forward stepwise selection

- Start from no features
- Sequentially add into the model the feature that most improves the fit

Backward stepwise selection (if n>p)

- Start from all features
- Sequentially removes from the model the feature that least degrades the fit

Other variants

Hybrid stepwise selection strategies that consider both forward and backward moves at each stage, and make the "best" move

Two flavors of wrapper methods

Forward stepwise selection

- Start from no features
- Sequentially add into the model the feature that most improves the fit

Backward stepwise selection (if n>p)

- Start from all features
- Sequentially removes from the model the feature that least degrades the fit

Other variants

Hybrid stepwise selection strategies that consider both forward and backward moves at each stage, and make the "best" move

Two flavors of wrapper methods

Forward stepwise selection

- Start from no features
- Sequentially add into the model the feature that most improves the fit

Backward stepwise selection (if n>p)

- Start from all features
- Sequentially removes from the model the feature that least degrades the fit

Other variants

Hybrid stepwise selection strategies that consider both forward and backward moves at each stage, and make the "best" move

Outline

The idea

- The following problem is NP-hard:

$$\min R(f_\beta) \quad \text{s.t.} \quad \| \beta \|_0 \leq k$$

- As a proxy we can consider the more general problem:

$$\min R(f_\beta) \quad \text{s.t.} \quad \Omega(\beta) \leq \gamma$$

where $\Omega(\beta)$ is a penalty function that leads to sparse solutions and to computationally efficient algorithms.

LASSO regression (Tibshirani, 1996)
Basis Pursuit (Chen et al., 1998)

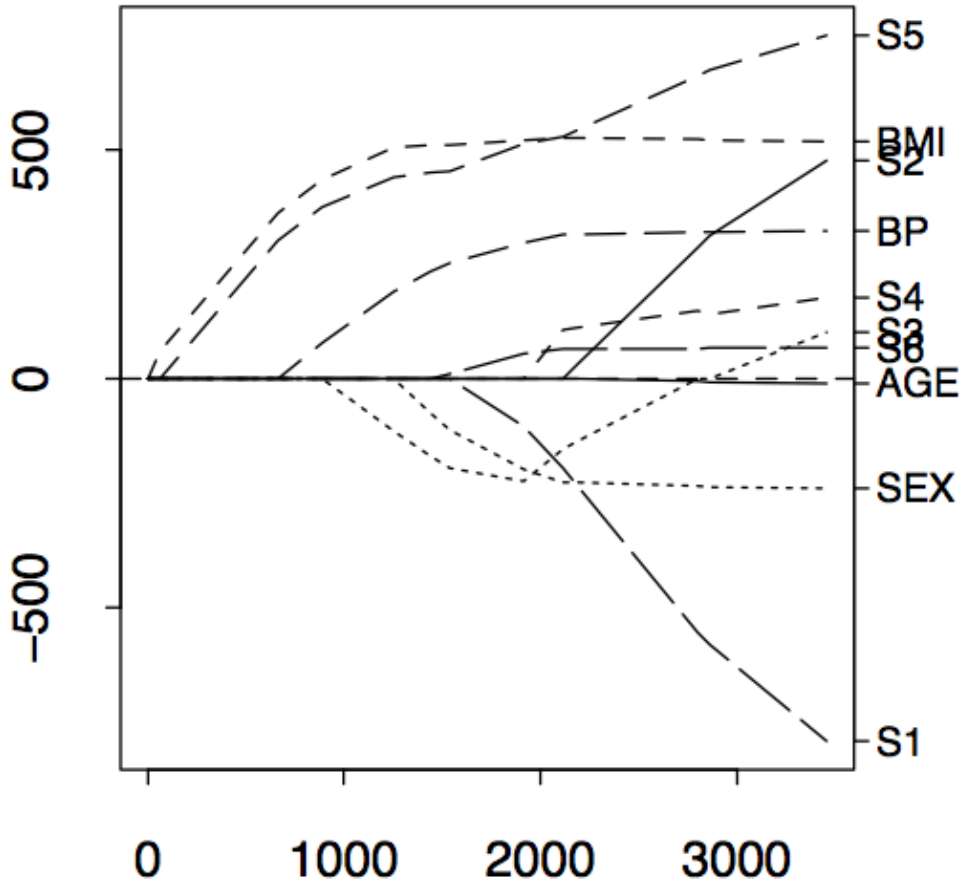
$$\Omega(\beta) = \| \beta \|_1 = \sum_{i=1}^{p} | \beta_i |$$

- LASSO or BP:

$$\min_{\beta} R(f_\beta) = \sum_{i=1}^{n} (f_\beta (\mathbf{x}_i) - \mathbf{y}_i)^2 + \lambda \sum_{i=1}^{p} | \beta_i | \qquad (2)$$

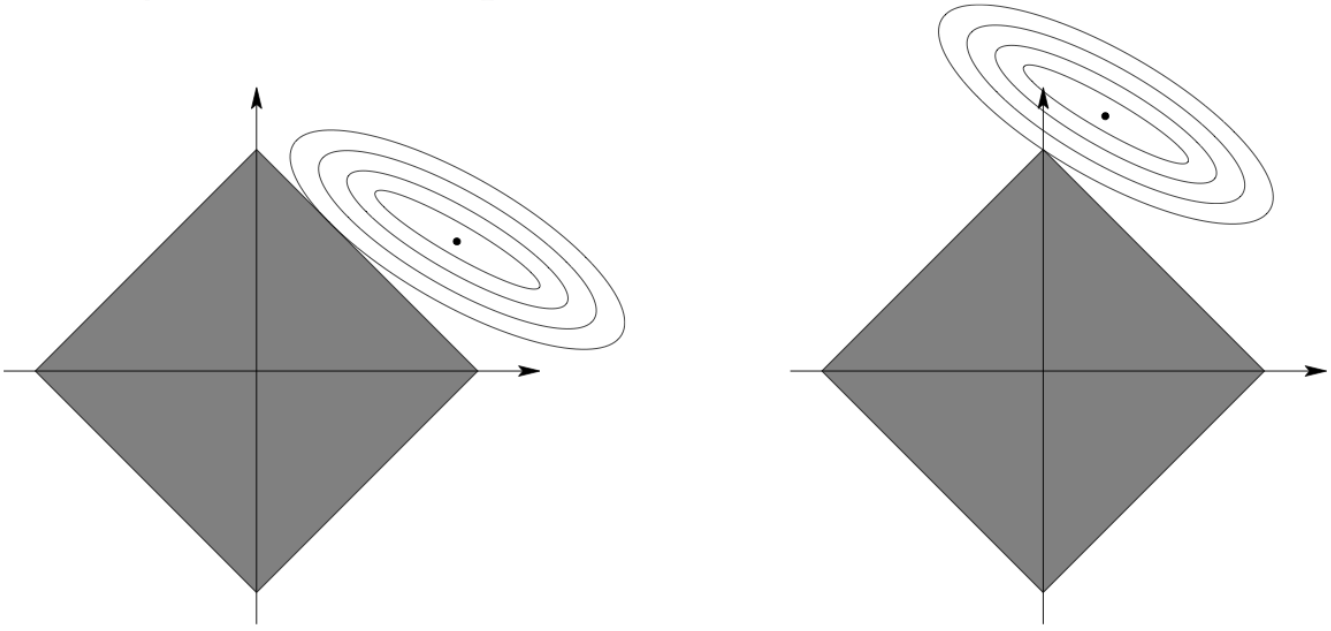
- No explicit solution, but this is just a quadratic program.
- LARS (Efron et al., 2004) provides a fast algorithm to compute the solution for all λ's simultaneously (regularization path)

LASSO regression example



Why LASSO leads to sparse solutions

Geometric interpretation with $p = 2$

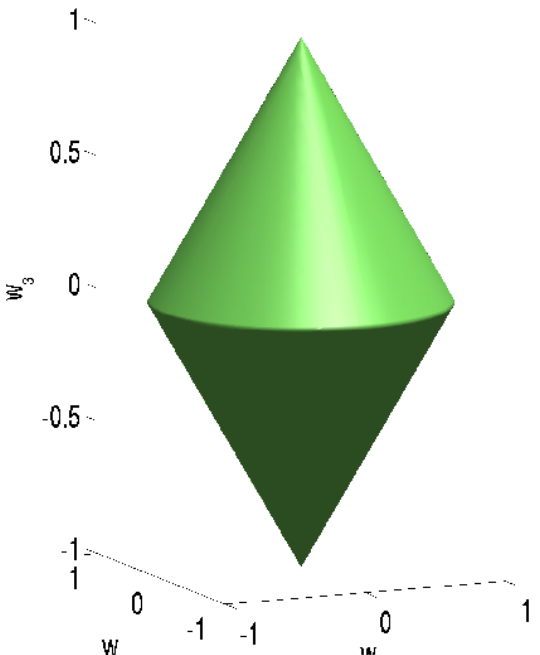


Generalization : Selecting pre-defined groups of variables

Group lasso (Yuan & Lin, 2006)

If groups of covariates are likely to be selected together, the ℓ_1/ℓ_2-norm induces sparse solutions *at the group level*:

$$\Omega_{group}(\beta) = \|\beta\|_{1,2} = \sum_g \|\beta_g\|_2$$

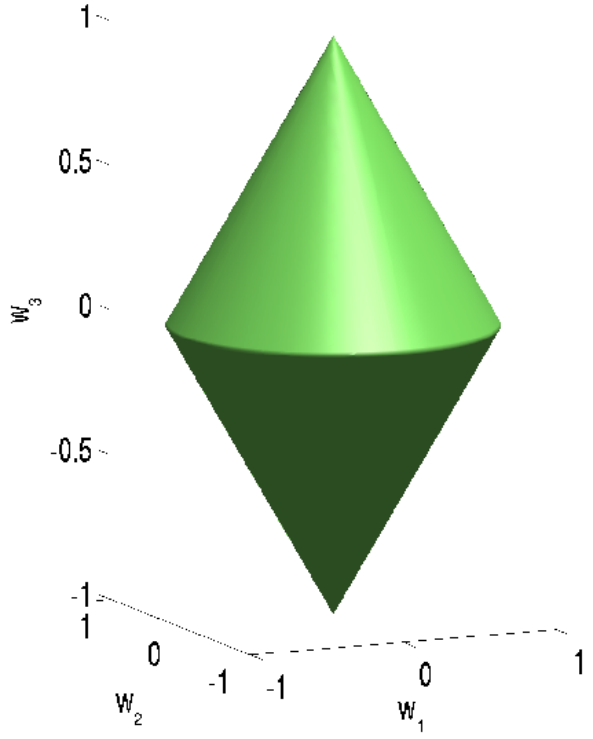


$$\Omega(\beta_1, \beta_2, \beta_3) = \|(\beta_1, \beta_2)\|_2 + \|\beta_3\|_2$$
$$= \sqrt{\beta_1^2 + \beta_2^2} + \sqrt{\beta_3^2}$$

Extension to other loss functions

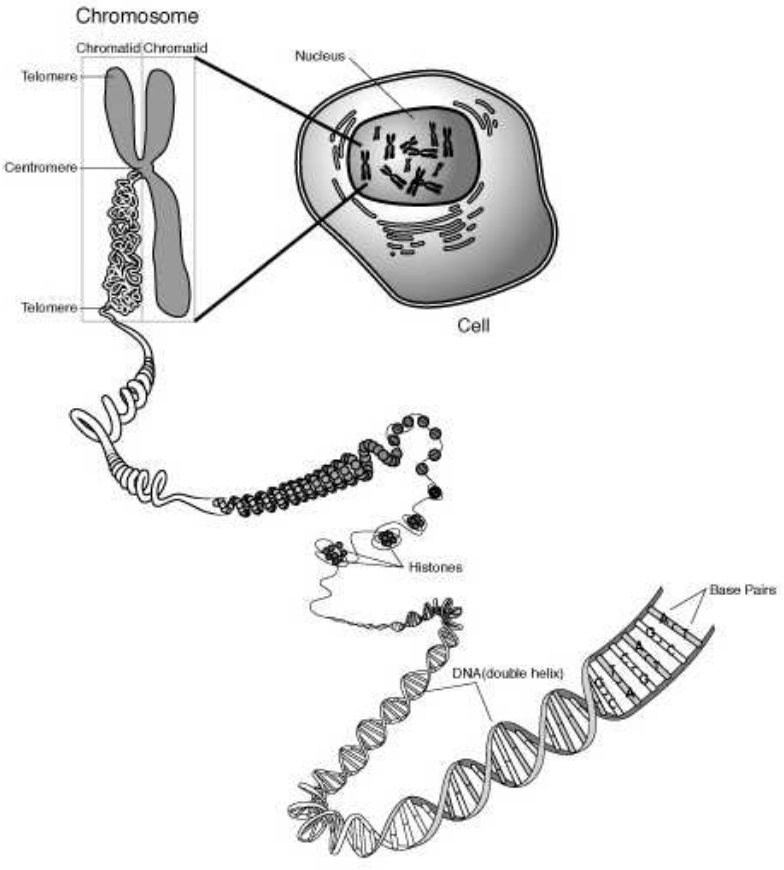
Of course we can learn sparse or group-sparse linear models with any different (smoothly convex) loss function:

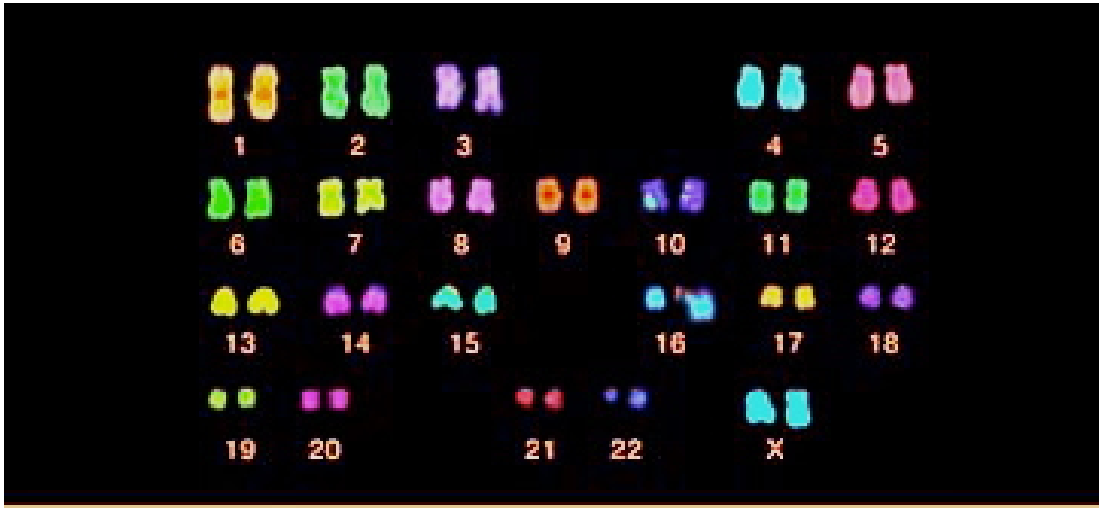
$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \ell\left(f_{\beta}\left(\mathbf{x}_i\right), \mathbf{y}_i\right) + \lambda \|\beta\|_1 \text{ or } \|\beta\|_{1,2}$$

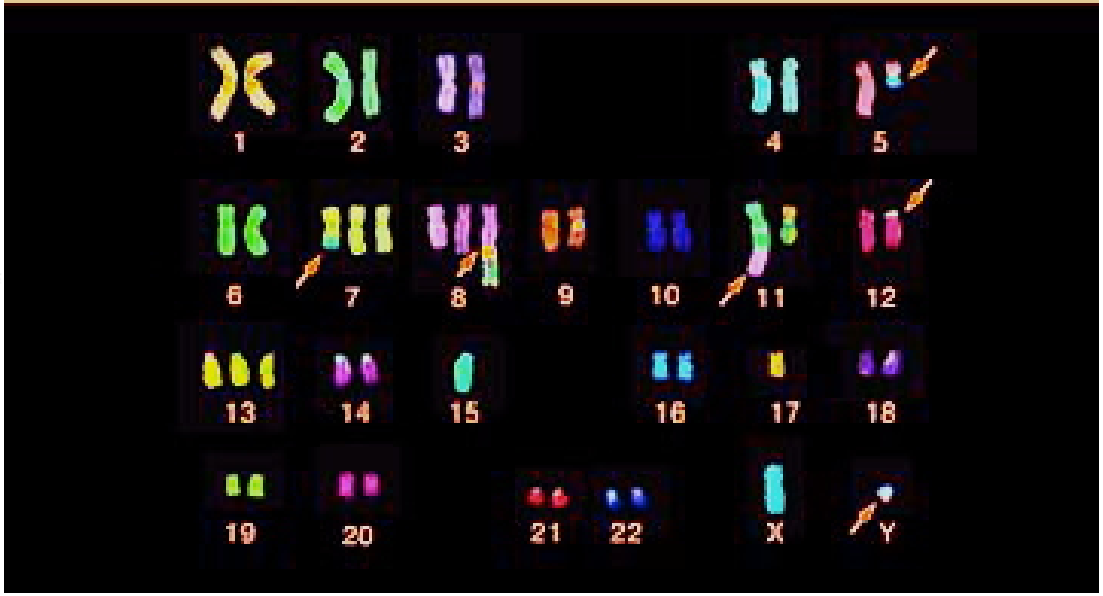


Outline

Chromosomic aberrations in cancer



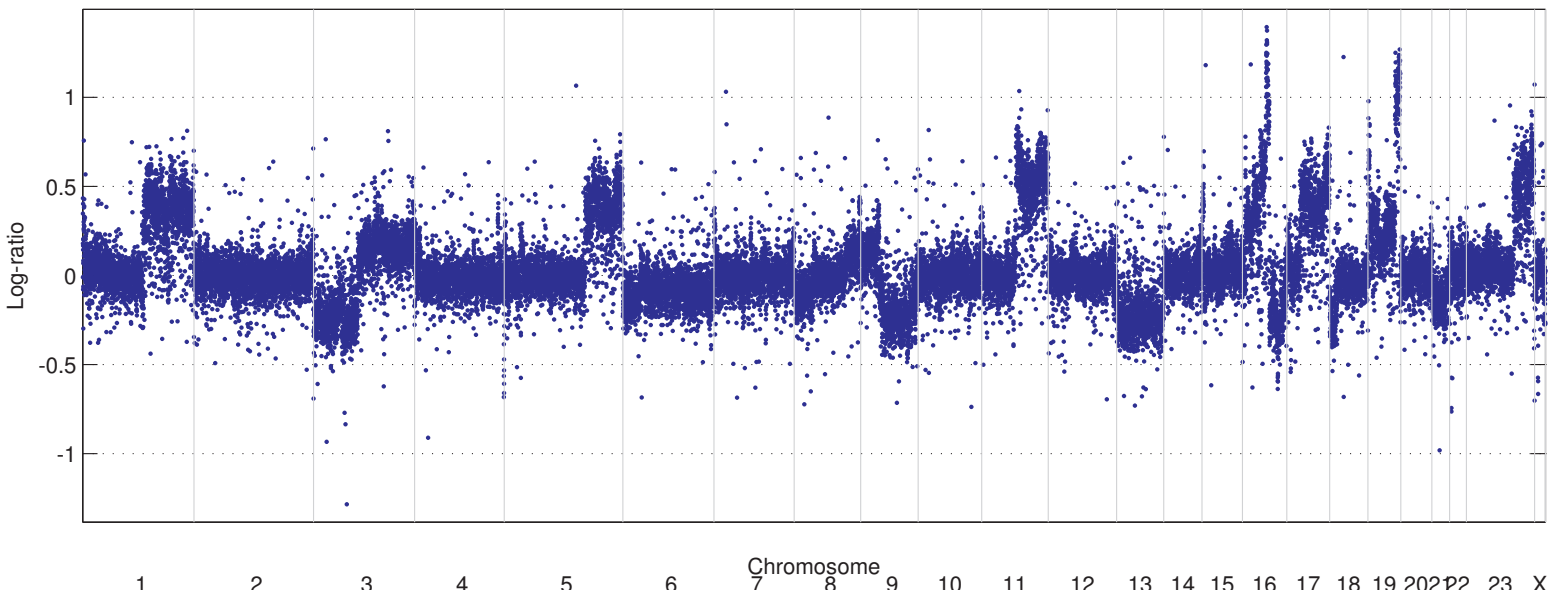




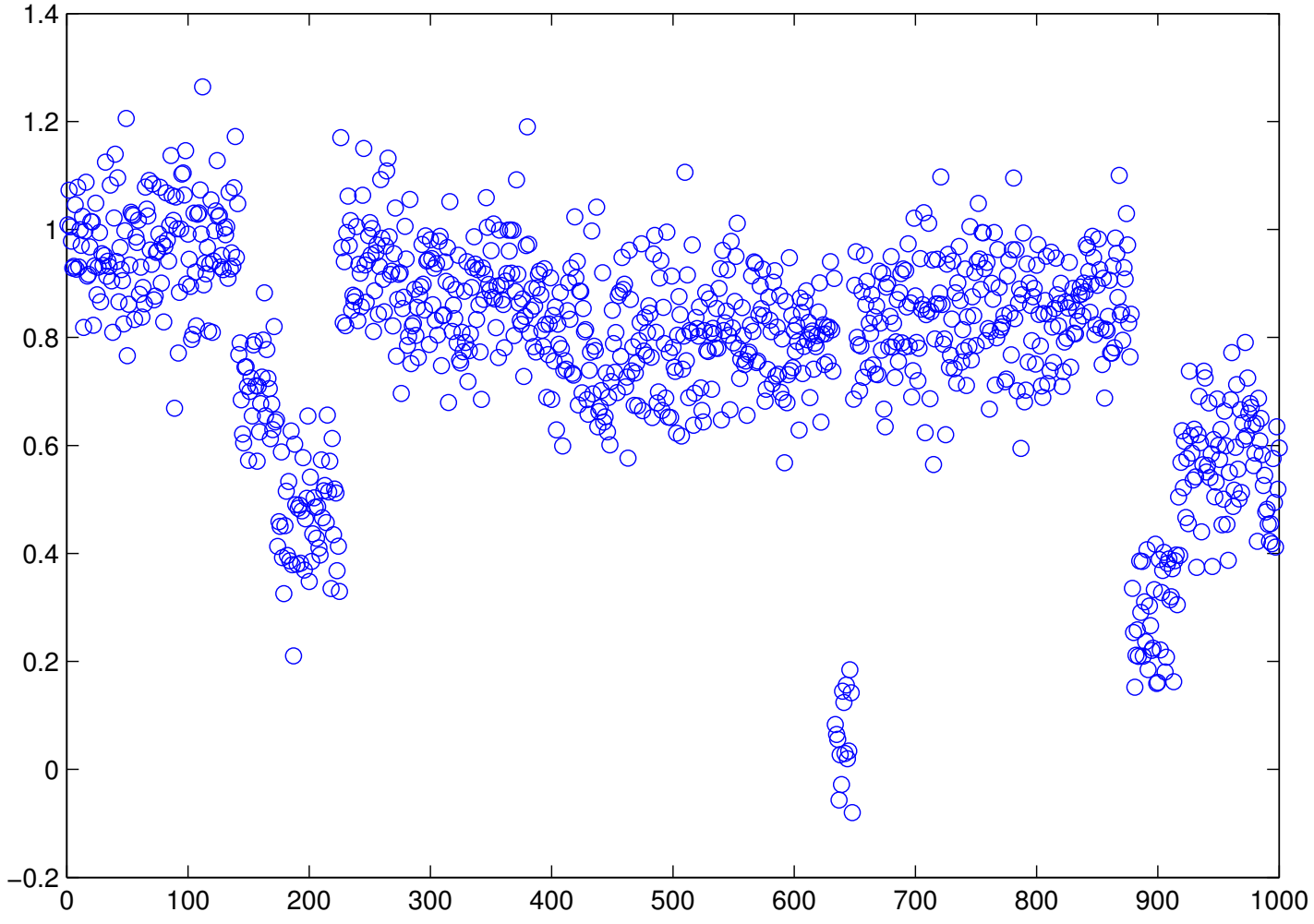
Comparative Genomic Hybridization (CGH)

Motivation

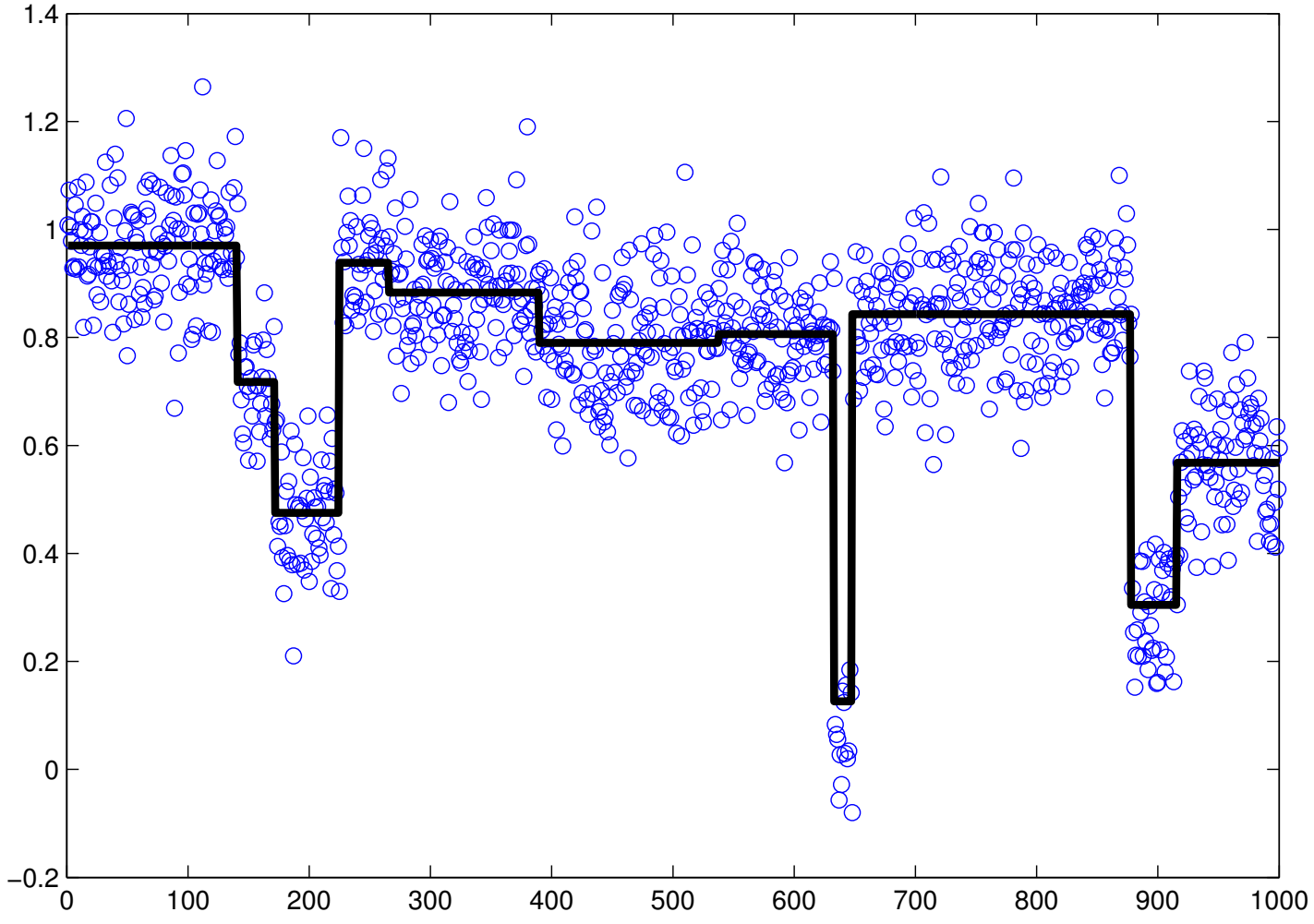
- Comparative genomic hybridization (CGH) data measure the DNA copy number along the genome
- Very useful, in particular in cancer research to observe systematically variants in DNA content



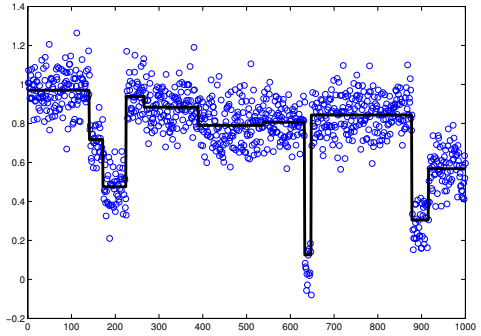
Where are the breakpoints?



Where are the breakpoints?



Optimal breakpoint detection

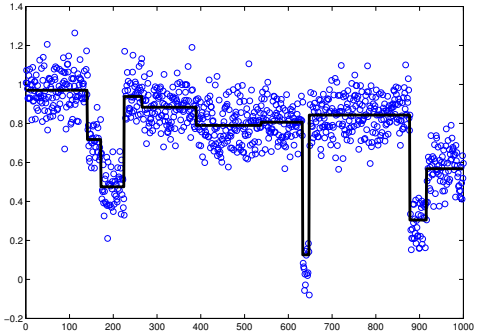


- Let $Y \in \mathbb{R}^p$ the signal. We search a smooth profile $\beta \in \mathbb{R}^p$ with at most k change-points by solving

$$\min_{\beta \in \mathbb{R}^p} \| Y - \beta \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} \mathbf{1}\left(\beta_{i+1} \neq \beta_i\right) \leq k$$

- This is an optimization problem over the $\binom{p}{k}$ partitions...
- Dynamic programming finds the solution in $O(p^2 k)$ in time and $O(p^2)$ in memory
- But: does not scale to $p = 10^6 \sim 10^9$...

Optimal breakpoint detection

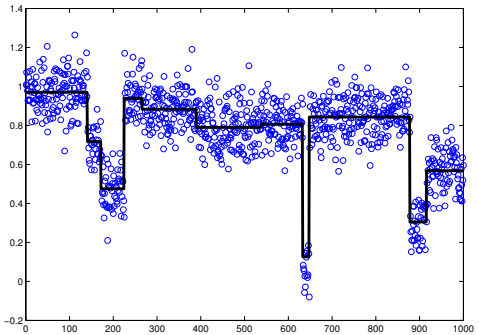


- Let $Y \in \mathbb{R}^p$ the signal. We search a smooth profile $\beta \in \mathbb{R}^p$ with at most k change-points by solving

$$\min_{\beta \in \mathbb{R}^p} \| Y - \beta \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} \mathbf{1}\left(\beta_{i+1} \neq \beta_i\right) \leq k$$

- This is an optimization problem over the $\binom{p}{k}$ partitions...
- Dynamic programming finds the solution in $O(p^2 k)$ in time and $O(p^2)$ in memory
- But: does not scale to $p = 10^6 \sim 10^9$...

Optimal breakpoint detection

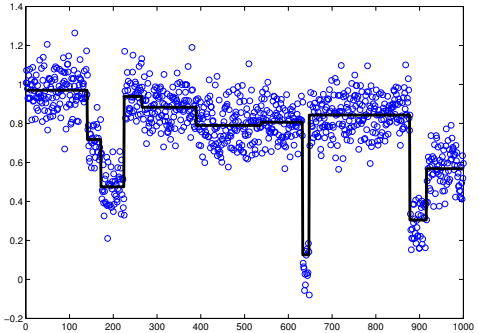


- Let $Y \in \mathbb{R}^p$ the signal. We search a smooth profile $\beta \in \mathbb{R}^p$ with at most k change-points by solving

$$\min_{\beta \in \mathbb{R}^p} \| Y - \beta \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} \mathbf{1}\left(\beta_{i+1} \neq \beta_i\right) \leq k$$

- This is an optimization problem over the $\binom{p}{k}$ partitions...
- Dynamic programming finds the solution in $O(p^2 k)$ in time and $O(p^2)$ in memory
- But: does not scale to $p = 10^6 \sim 10^9$...

Optimal breakpoint detection



- Let $Y \in \mathbb{R}^p$ the signal. We search a smooth profile $\beta \in \mathbb{R}^p$ with at most k change-points by solving

$$\min_{\beta \in \mathbb{R}^p} \| Y - \beta \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} \mathbf{1}\left(\beta_{i+1} \neq \beta_i\right) \leq k$$

- This is an optimization problem over the $\binom{p}{k}$ partitions...
- Dynamic programming finds the solution in $O(p^2 k)$ in time and $O(p^2)$ in memory
- But: does not scale to $p = 10^6 \sim 10^9$...

Promoting piecewise constant profiles

$$\Omega(\beta) = \|\beta\|_{TV} = \sum_{i=1}^{p-1} |\beta_{i+1} - \beta_i|$$

The total variation / variable fusion penalty

If $R(\beta)$ is convex and "smooth", the solution of

$$\min_{\beta \in \mathbb{R}^p} R(\beta) + \lambda \sum_{i=1}^{p-1} |\beta_{i+1} - \beta_i|$$

is usually piecewise constant (Rudin et al., 1992; Land and Friedman, 1996).

Proof:

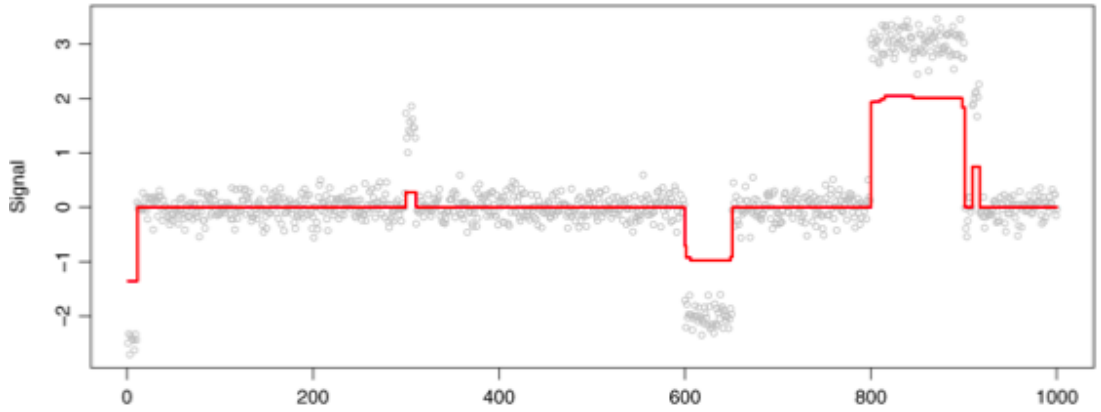
- Change of variable $u_i = \beta_{i+1} - \beta_i$, $u_0 = \beta_1$
- We obtain a Lasso problem in $u \in \mathbb{R}^{p-1}$
- u sparse means β piecewise constant

TV signal approximator

$$\min_{\beta \in \mathbb{R}^p} \| Y - \beta \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} | \beta_{i+1} - \beta_i | \leq \mu$$

Adding additional constraints does not change the change-points:

- $\sum_{i=1}^{p} | \beta_i | \leq \nu$ (Tibshirani et al., 2005; Tibshirani and Wang, 2008)
- $\sum_{i=1}^{p} \beta_i^2 \leq \nu$ (Mairal et al. 2010)



Solving TV signal approximator

$$\min_{\beta \in \mathbb{R}^p} \| Y - \beta \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} | \beta_{i+1} - \beta_i | \leq \mu$$

- QP with sparse linear constraints in $O(p^2)$ -> 135 min for $p = 10^5$ (Tibshirani and Wang, 2008)
- Coordinate descent-like method $O(p)$? -> 3s s for $p = 10^5$ (Friedman et al., 2007)
- For all μ with the LARS in $O(pK)$ (Harchaoui and Levy-Leduc, 2008)
- For all μ in $O(p \ln p)$ (Hoefling, 2009)
- For the first K change-points in $O(p \ln K)$ (Bleakley and V., 2010)

TV signal approximator as dichotomic segmentation

Algorithm 1 Greedy dichotomic segmentation

Require: k number of intervals, $\gamma(I)$ gain function to split an interval I into $I_L(I), I_R(I)$

1: I_0 represents the interval $[1, n]$
2: $\mathcal{P} = \{I_0\}$
3: **for** $i = 1$ to k **do**
4: $\quad I^* \leftarrow \arg\max_{I \in \mathcal{P}} \gamma(I^*)$
5: $\quad \mathcal{P} \leftarrow \mathcal{P} \setminus \{I^*\}$
6: $\quad \mathcal{P} \leftarrow \mathcal{P} \cup \{I_L(I^*), I_R(I^*)\}$
7: **end for**
8: **return** $\mathcal{P}$

Theorem (V. and Bleakley, 2010; see also Hoefling, 2009)

TV signal approximator performs "greedy" dichotomic segmentation

Apparently greedy algorithm finds the global optimum!

TV signal approximator as dichotomic segmentation

Algorithm 1 Greedy dichotomic segmentation

Require: k number of intervals, $\gamma(I)$ gain function to split an interval I into $I_L(I), I_R(I)$

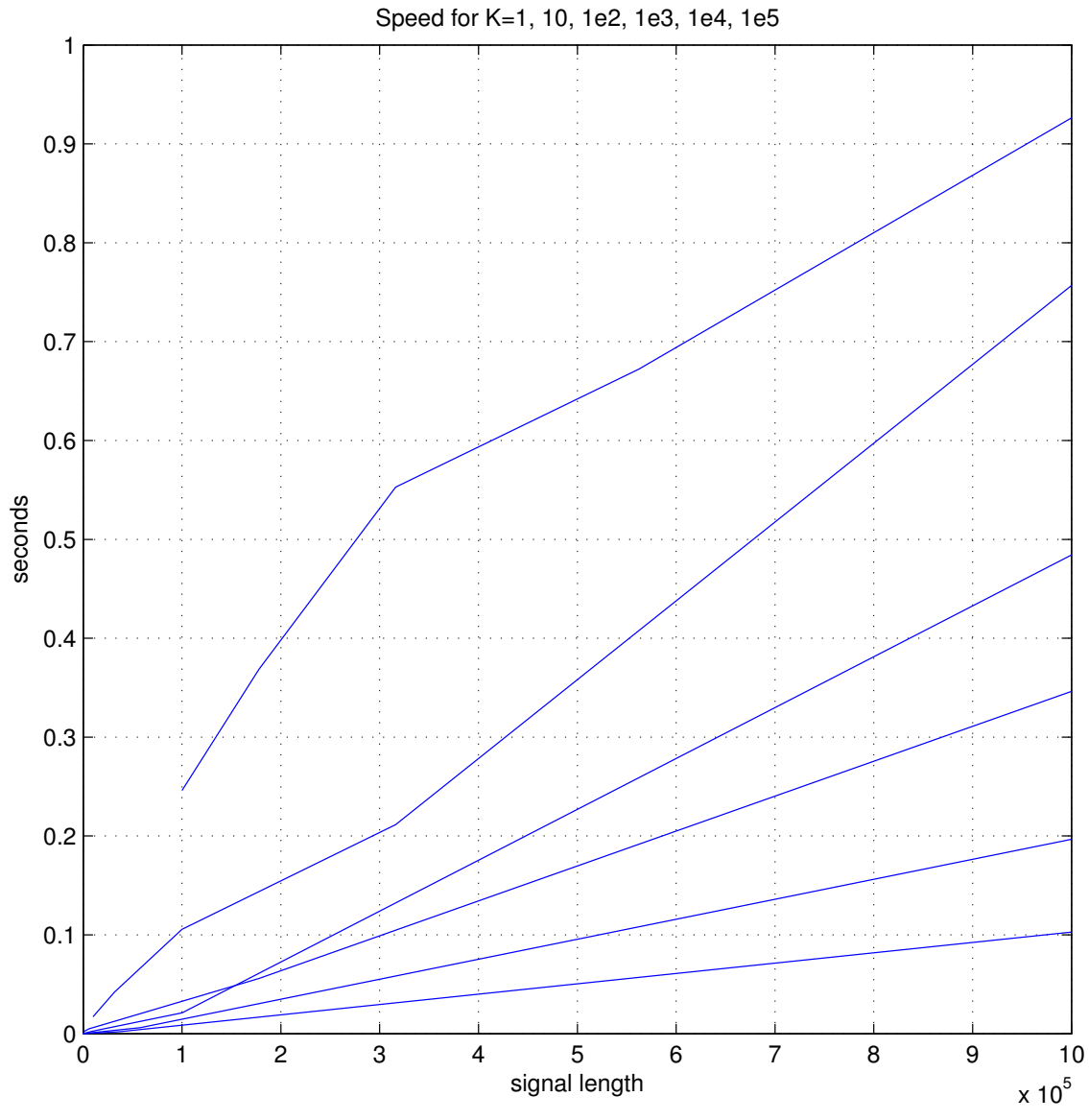
1: I_0 represents the interval $[1, n]$
2: $\mathcal{P} = \{I_0\}$
3: **for** $i = 1$ to k **do**
4: $\quad I^* \leftarrow \arg\max_{I \in \mathcal{P}} \gamma(I^*)$
5: $\quad \mathcal{P} \leftarrow \mathcal{P} \setminus \{I^*\}$
6: $\quad \mathcal{P} \leftarrow \mathcal{P} \cup \{I_L(I^*), I_R(I^*)\}$
7: **end for**
8: **return** $\mathcal{P}$

Theorem (V. and Bleakley, 2010; see also Hoefling, 2009)

TV signal approximator performs "greedy" dichotomic segmentation

Apparently greedy algorithm finds the global optimum!

Speed trial : 2 s. for $K = 100$, $p = 10^7$



BIOINFORMATICS APPLICATIONS NOTE

Vol. 27 no. 2 2011, pages 268–269
doi:10.1093/bioinformatics/btq635

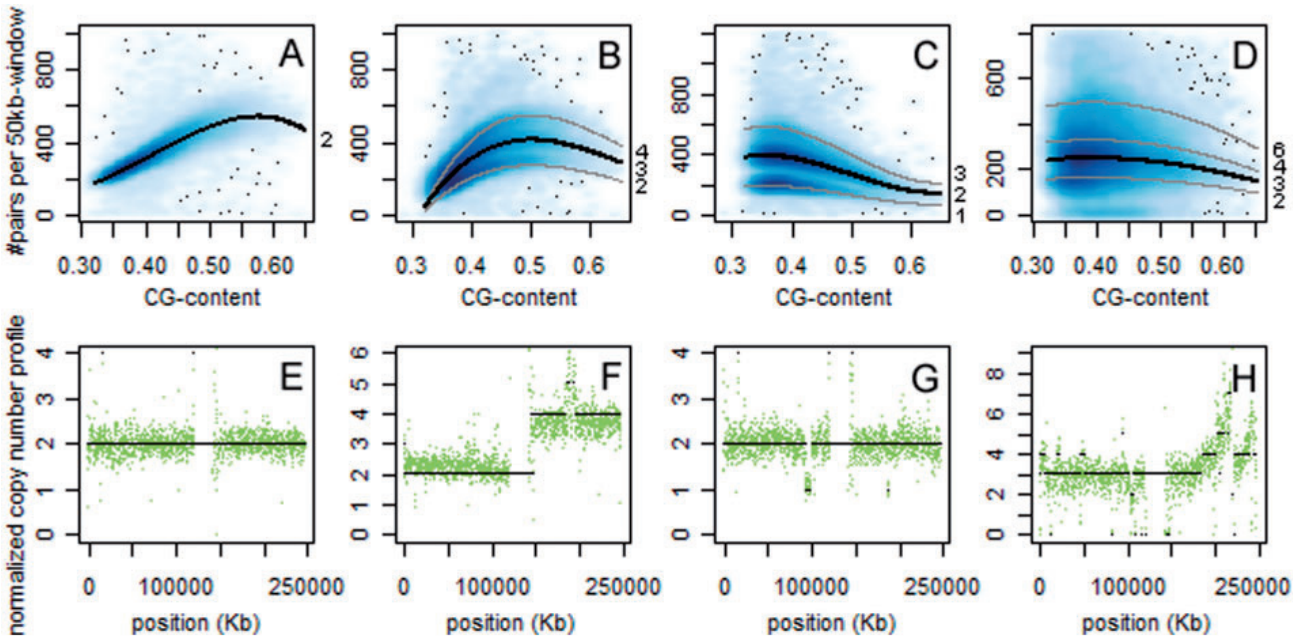
Genome analysis

Advance Access publication November 15, 2010

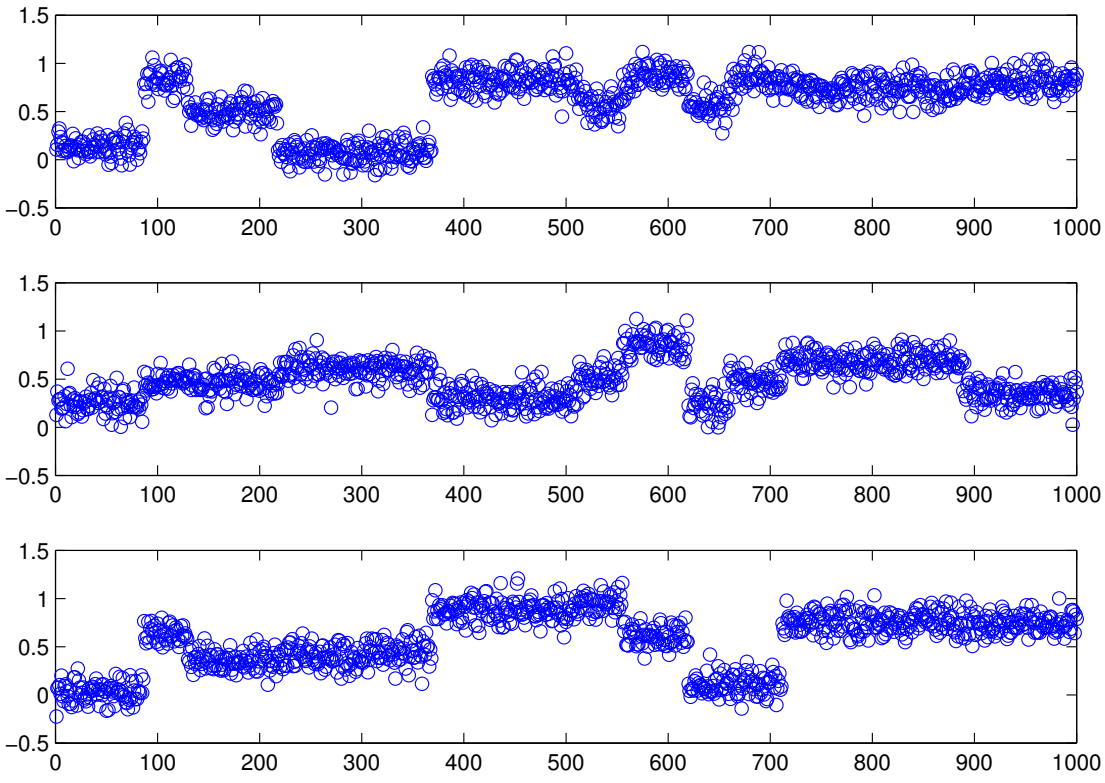
Control-free calling of copy number alterations in deep-sequencing data using GC-content normalization

Valentina Boeva[1,2,3,4,*], Andrei Zinovyev[1,2,3], Kevin Bleakley[1,2,3], Jean-Philippe Vert[1,2,3], Isabelle Janoueix-Lerosey[1,4], Olivier Delattre[1,4] and Emmanuel Barillot[1,2,3]

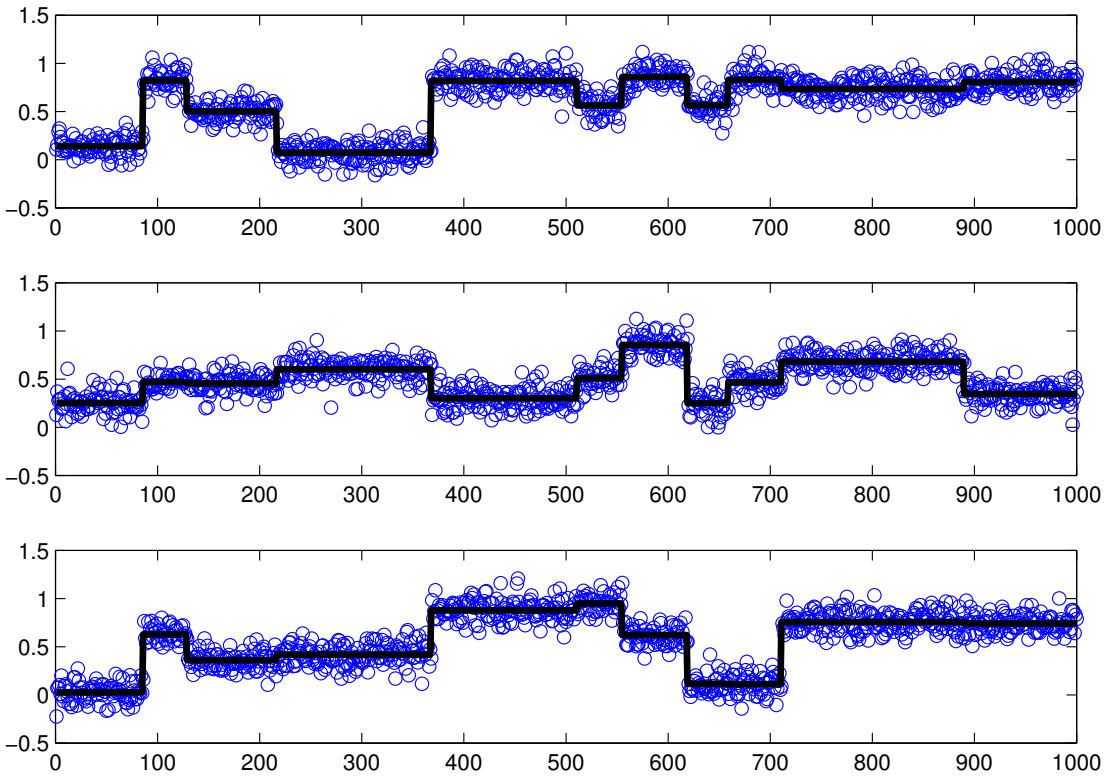
[1]Institut Curie, [2]INSERM, U900, Paris, F-75248, [3]Mines ParisTech, Fontainebleau, F-77300 and [4]INSERM, U830, Paris, F-75248 France



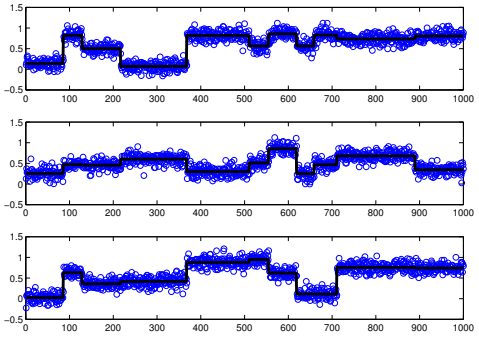
Extension 1: finding multiple change points shared by several profiles



Extension 1: finding multiple change points shared by several profiles



"Optimal" segmentation by dynamic programming



- Define the "optimal" piecewise constant approximation $\hat{U} \in \mathbb{R}^{p \times n}$ of Y as the solution of

$$\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} \mathbf{1}\left(U_{i+1,\bullet} \neq U_{i,\bullet}\right) \leq k$$

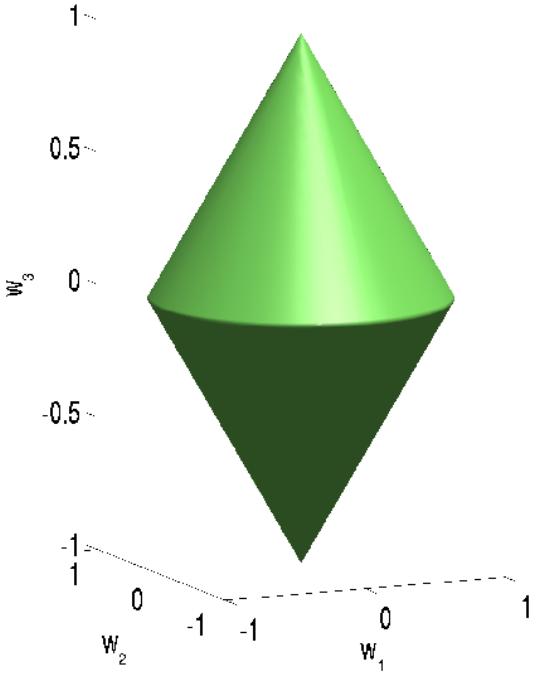
- DP finds the solution in $O(p^2 kn)$ in time and $O(p^2)$ in memory
- But: does not scale to $p = 10^6 \sim 10^9$...

Selecting pre-defined groups of variables

Group lasso (Yuan & Lin, 2006)

If groups of covariates are likely to be selected together, the ℓ_1/ℓ_2-norm induces sparse solutions *at the group level*:

$$\Omega_{group}(w) = \sum_g \|w_g\|_2$$



$$\begin{aligned}\Omega(w_1, w_2, w_3) &= \|(w_1, w_2)\|_2 + \|w_3\|_2 \\ &= \sqrt{w_1^2 + w_2^2} + \sqrt{w_3^2}\end{aligned}$$

GFLseg (Bleakley and V., 2011)

Replace

$$\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} \mathbf{1}\left(U_{i+1,\bullet} \neq U_{i,\bullet}\right) \leq k$$

by

$$\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} w_i \| U_{i+1,\bullet} - U_{i,\bullet} \| \leq \mu$$

GFLseg = Group Fused Lasso segmentation

Questions

- Practice: can we solve it efficiently?
- Theory: does it recover the correct segmentation?

GFLseg (Bleakley and V., 2011)

Replace

$$\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} \mathbf{1} \left(U_{i+1,\bullet} \neq U_{i,\bullet} \right) \leq k$$

by

$$\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} w_i \| U_{i+1,\bullet} - U_{i,\bullet} \| \leq \mu$$

GFLseg = Group Fused Lasso segmentation

Questions

- Practice: can we solve it efficiently?
- Theory: does it recover the correct segmentation?

TV approximator implementation

$$\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} w_i \| U_{i+1,\bullet} - U_{i,\bullet} \| \leq \mu$$

Theorem

The TV approximator can be solved efficiently:

- approximately with the group LARS in $O(npk)$ in time and $O(np)$ in memory
- exactly with a block coordinate descent + active set method in $O(np)$ in memory

Speed trial

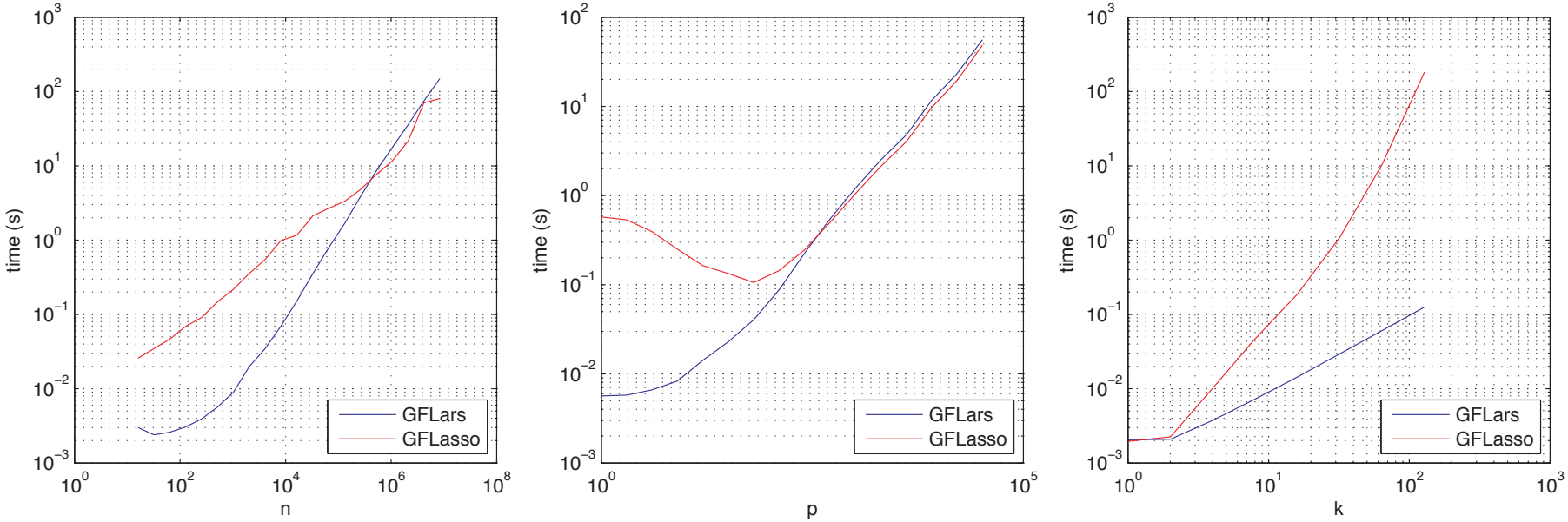
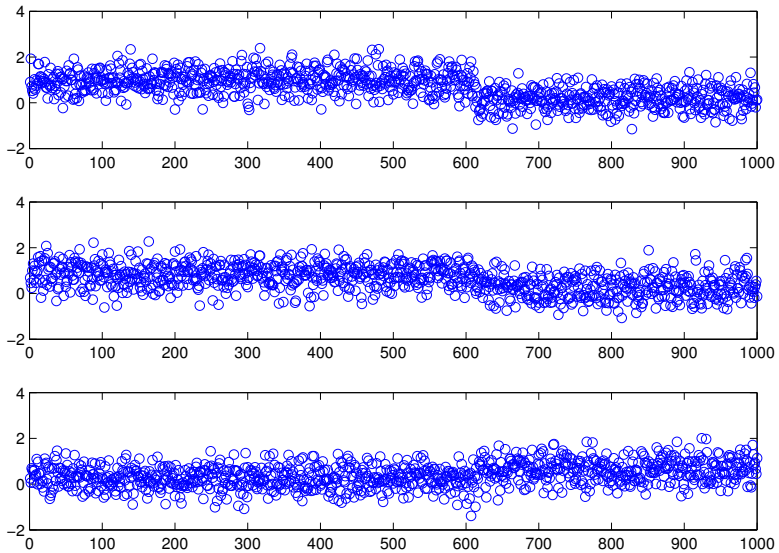


Figure 2: **Speed trials for group fused LARS (top row) and Lasso (bottom row).** *Left column:* varying n, with fixed $p = 10$ and $k = 10$; *center column:* varying p, with fixed $n = 1000$ and $k = 10$; *right column:* varying k, with fixed $n = 1000$ and $p = 10$. Figure axes are log-log. Results are averaged over 100 trials.

Consistency

Suppose a single change-point:

- at position $u = \alpha p$
- with increments $(\beta_i)_{i=1,\ldots,n}$ s.t. $\bar{\beta}^2 = \lim_{k\to\infty} \frac{1}{n} \sum_{i=1}^{n} \beta_i^2$
- corrupted by i.i.d. Gaussian noise of variance σ^2



Does the TV approximator correctly estimate the first change-point as p increases?

Consistency of the weighted TV approximator

$$\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} w_i \| U_{i+1,\bullet} - U_{i,\bullet} \| \leq \mu$$

Theorem

The weighted TV approximator with weights

$$\forall i \in [1, p-1] \ , \quad w_i = \sqrt{\frac{i(p-i)}{p}}$$

correctly finds the first change-point with probability tending to 1 *as* $n \to +\infty$.

- we see the benefit of increasing n
- we see the benefit of adding weights to the TV penalty

Consistency for a single change-point

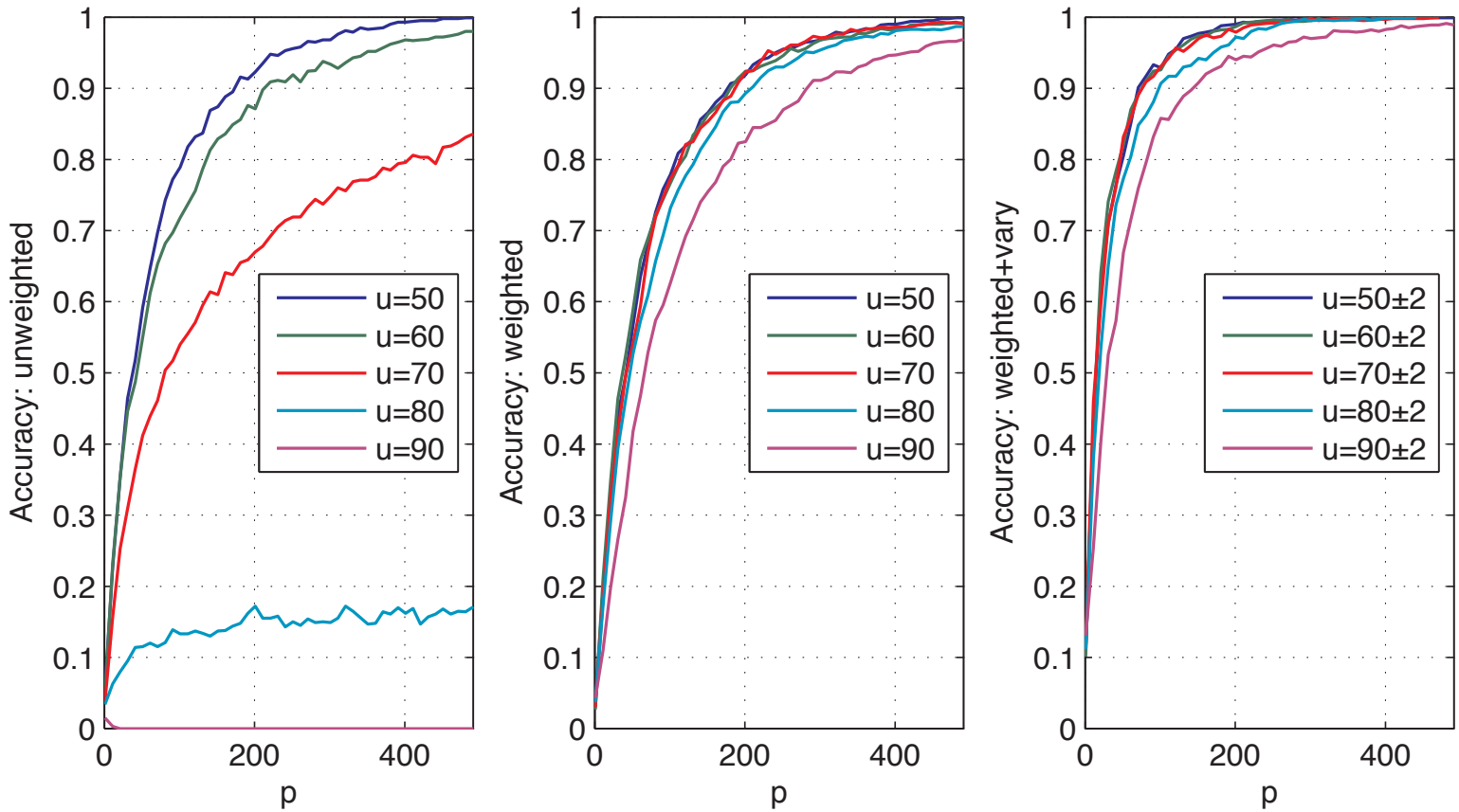


Figure 3: **Single change-point accuracy for the group fused Lasso.** Accuracy as a function of the number of profiles p when the change-point is placed in a variety of positions $u = 50$ to $u = 90$ (left and centre plots, resp. unweighted and weighted group fused Lasso), or: $u = 50 \pm 2$ to $u = 90 \pm 2$ (right plot, weighted with varying change-point location), for a signal of length 100.

Estimation of several change-points

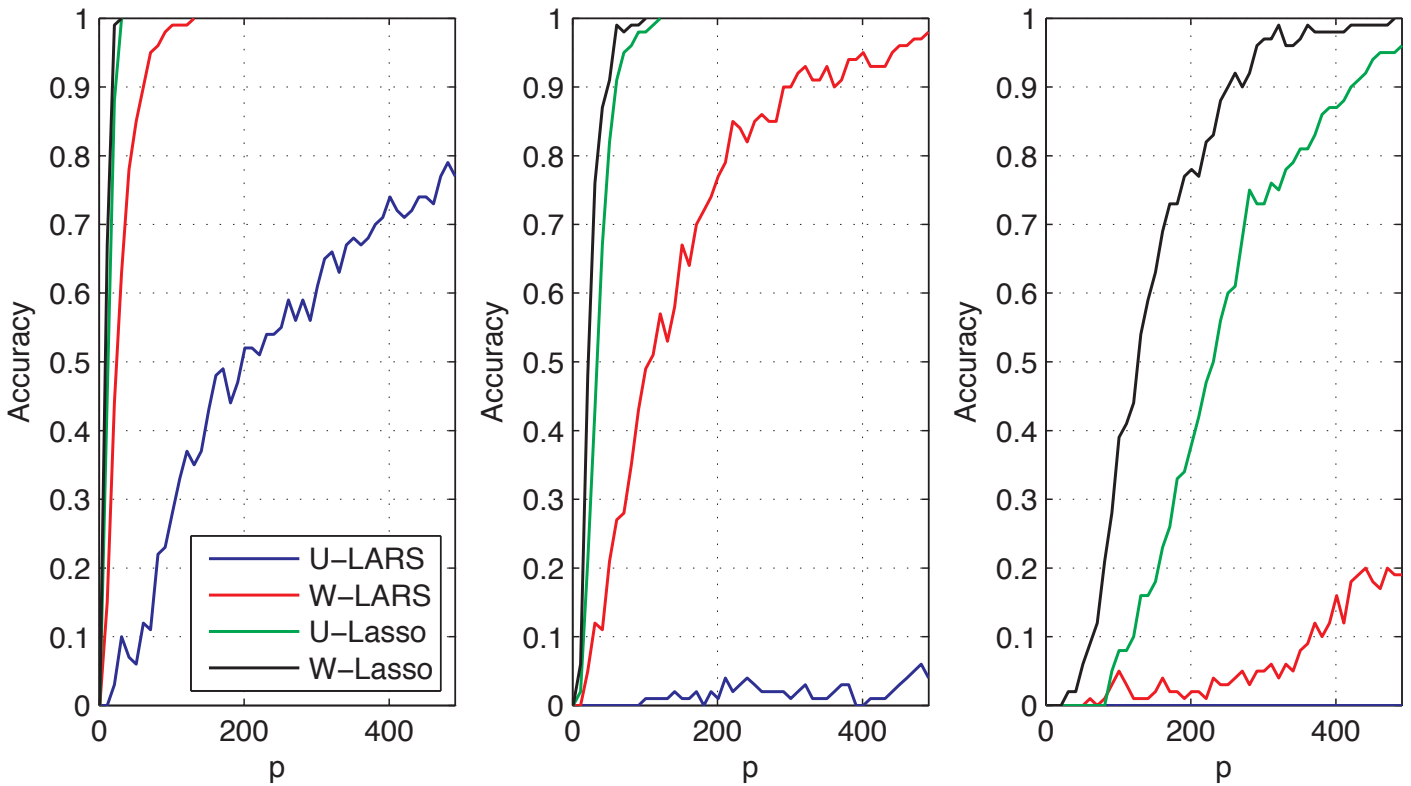
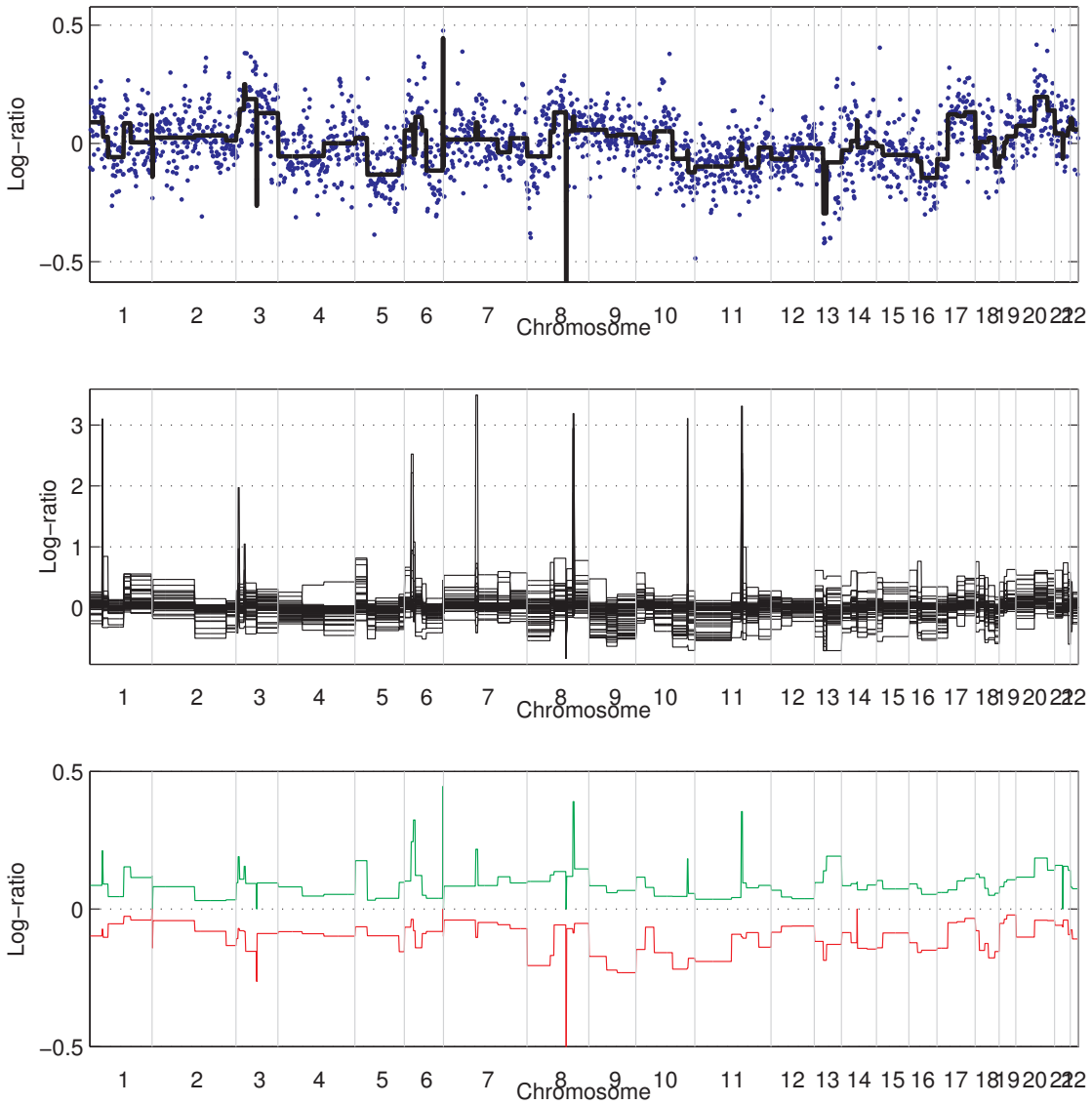
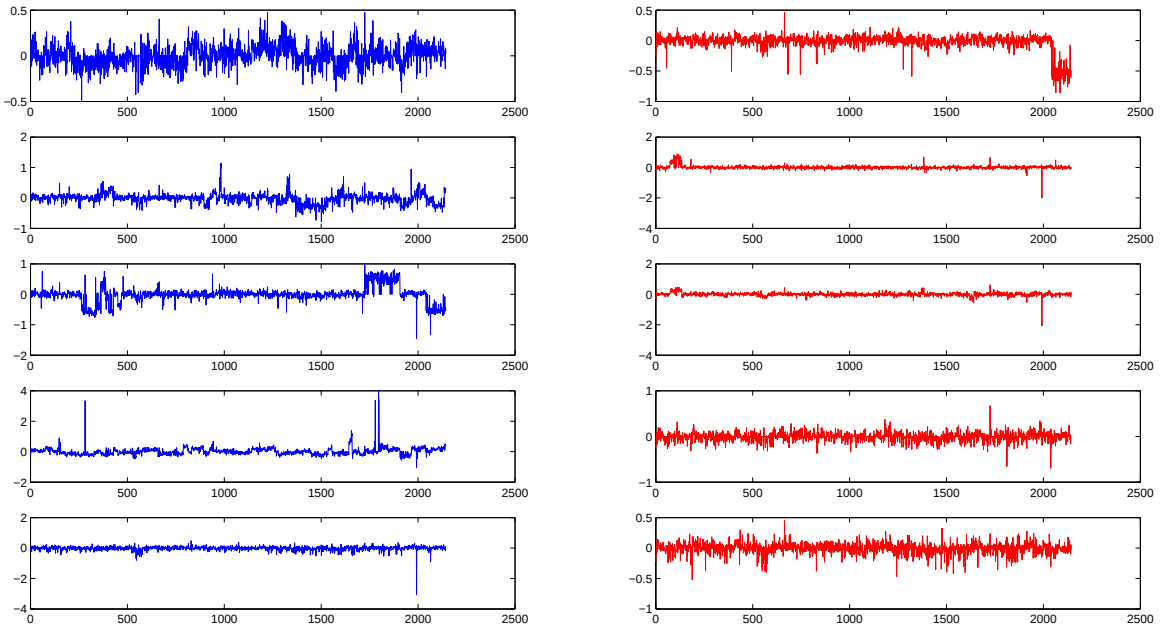


Figure 4: **Multiple change-point accuracy.** Accuracy as a function of the number of profiles p when change-points are placed at the nine positions $\{10, 20, \ldots, 90\}$ and the variance σ^2 of the centered Gaussian noise is either 0.05 (left), 0.2 (center) and 1 (right). The profile length is 100.

Application: detection of frequent abnormalities



Extension 2: Supervised classification of genomic profiles

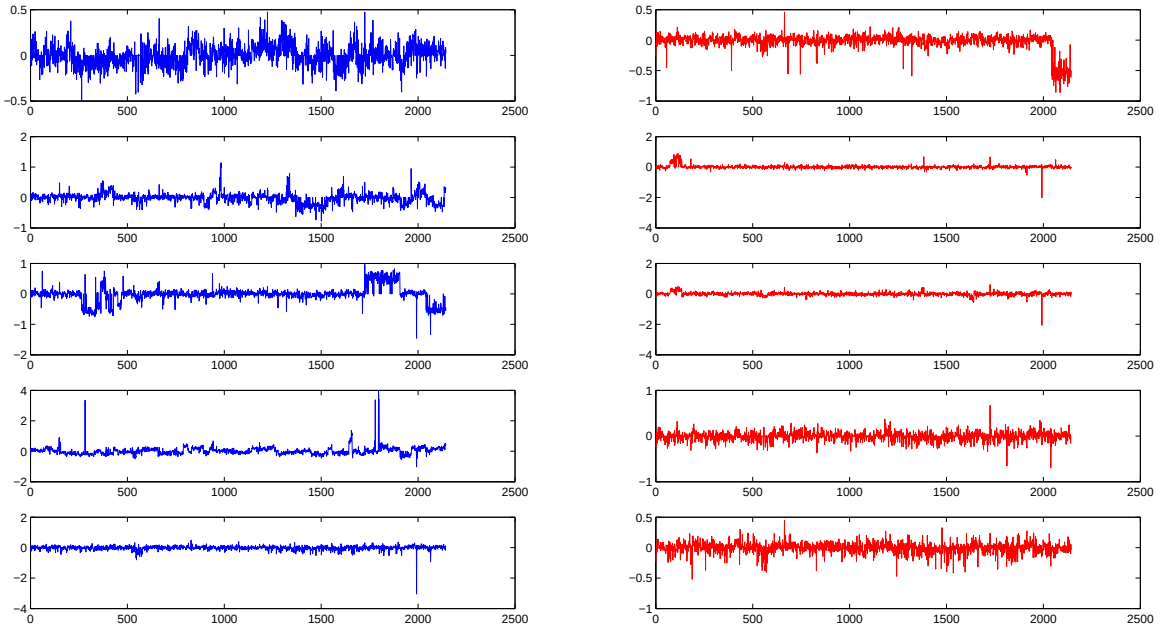


- $x_1, \ldots, x_n \in \mathbb{R}^p$ the n profiles of length p
- $y_1, \ldots, y_n \in [-1, 1]$ the labels
- We want to learn a function $f : \mathbb{R}^p \to [-1, 1]$

Prior knowledge

We expect β to be

- sparse : not all positions should be discriminative, and we want to identify the predictive region (presence of oncogenes or tumor suppressor genes?)
- piecewise constant : within a selected region, all probes should contribute equally



Fused lasso for supervised classification (Rapaport et al., 2008)

$$\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} \ell\left(y_i, \beta^\top x_i\right) + \lambda_1 \sum_{i=1}^{p} |\beta_i| + \lambda_2 \sum_{i=1}^{p-1} |\beta_{i+1} - \beta_i| \,.$$

where ℓ is, e.g., the hinge loss $\ell(y, t) = max(1 - yt, 0)$.

Implementation

- When ℓ is the hinge loss (fused SVM), this is a linear program -> up to $p = 10^3 \sim 10^4$
- When ℓ is convex and smooth (logistic, quadratic), efficient implementation with proximal methods -> up to $p = 10^8 \sim 10^9$

Fused lasso for supervised classification (Rapaport et al., 2008)

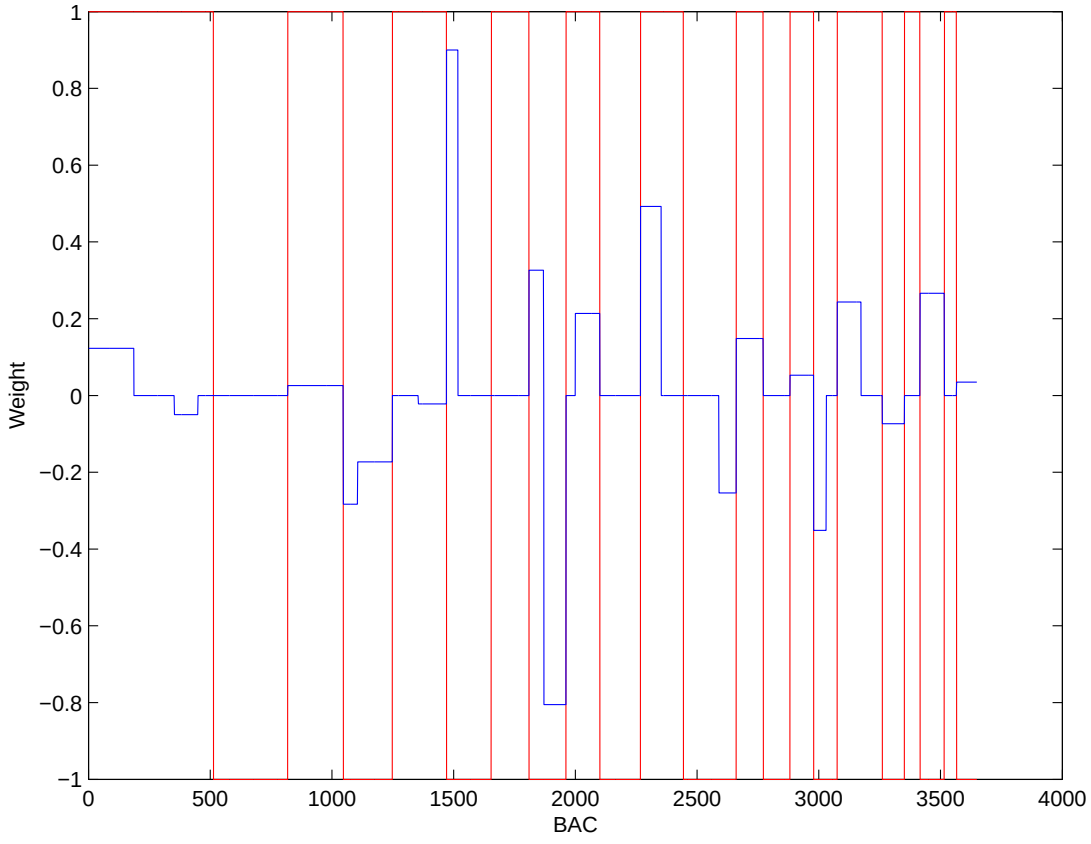
$$\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} \ell\left(y_i, \beta^\top x_i\right) + \lambda_1 \sum_{i=1}^{p} |\beta_i| + \lambda_2 \sum_{i=1}^{p-1} |\beta_{i+1} - \beta_i| \,.$$

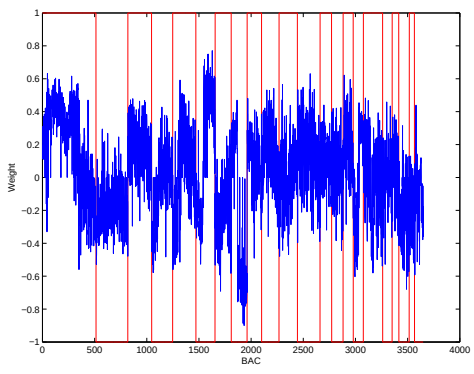
where ℓ is, e.g., the hinge loss $\ell(y, t) = max(1 - yt, 0)$.

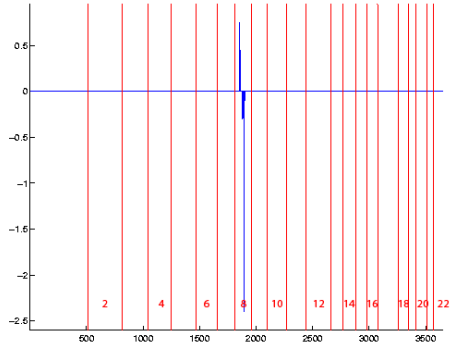
Implementation

- When ℓ is the hinge loss (fused SVM), this is a linear program -> up to $p = 10^3 \sim 10^4$
- When ℓ is convex and smooth (logistic, quadratic), efficient implementation with proximal methods -> up to $p = 10^8 \sim 10^9$

Example: predicting metastasis in melanoma





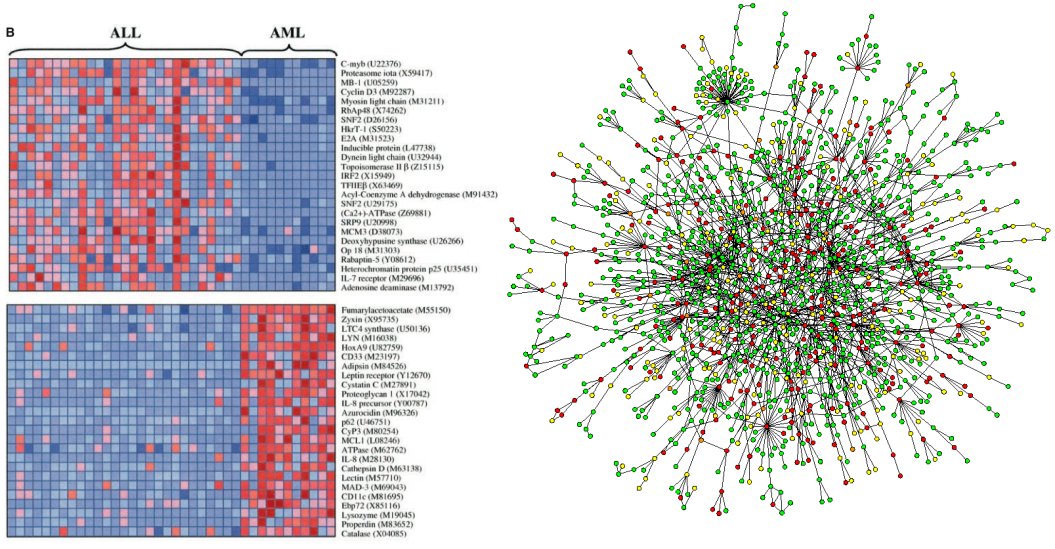


Outline

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

$$\min_{\beta} R(\beta) + \lambda \Omega_G(\beta)$$

Hypothesis

We would like to design penalties $\Omega_G(\beta)$ to promote one of the following hypothesis:

- Hypothesis 1: genes near each other on the graph should have similar weights (but we do not try to select only a few genes), i.e., the classifier should be smooth on the graph
- Hypothesis 2: genes selected in the signature should be connected to each other, or be in a few known functional groups, without necessarily having similar weights.

Graph based penalty with kernels

Prior hypothesis

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

Network kernel (Rapaport et al., 2007)

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

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

Graph based penalty with kernels

Prior hypothesis

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

Network kernel (Rapaport et al., 2007)

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

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

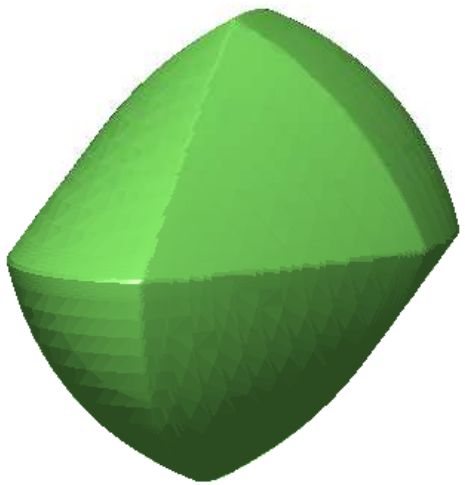
Other penalties without kernels

- Gene selection + Piecewise constant on the graph

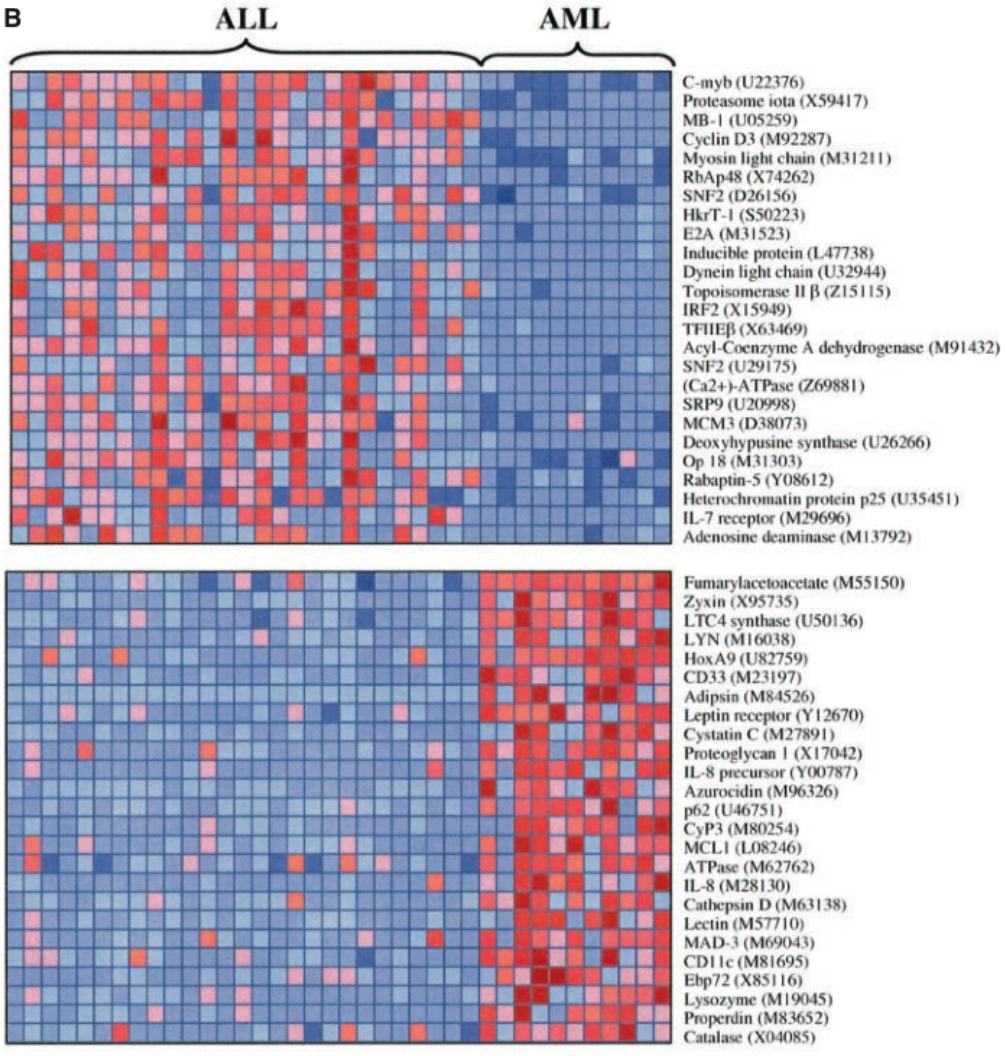
$$\Omega(\beta) = \sum_{i \sim j} |\beta_i - \beta_j| + \sum_{i=1}^{p} |\beta_i|$$

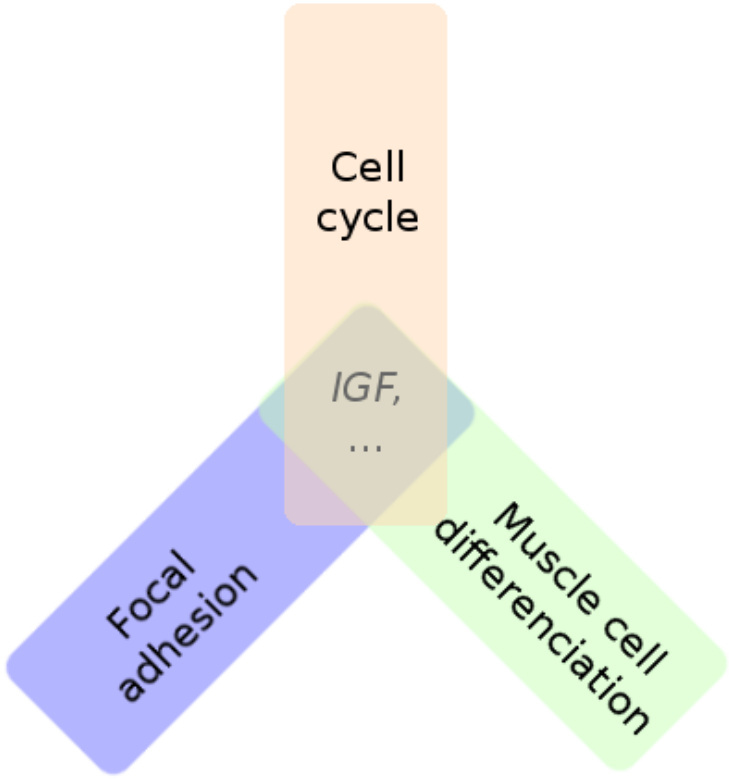
- Gene selection + smooth on the graph

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



How to select jointly genes belonging to predefined pathways?



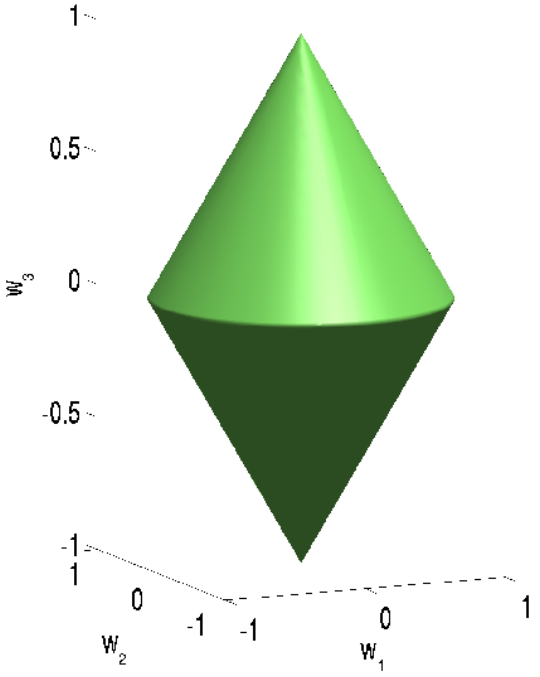


Selecting pre-defined groups of variables

Group lasso (Yuan & Lin, 2006)

If groups of covariates are likely to be selected together, the ℓ_1/ℓ_2-norm induces sparse solutions *at the group level*:

$$\Omega_{group}(w) = \sum_g \|w_g\|_2$$

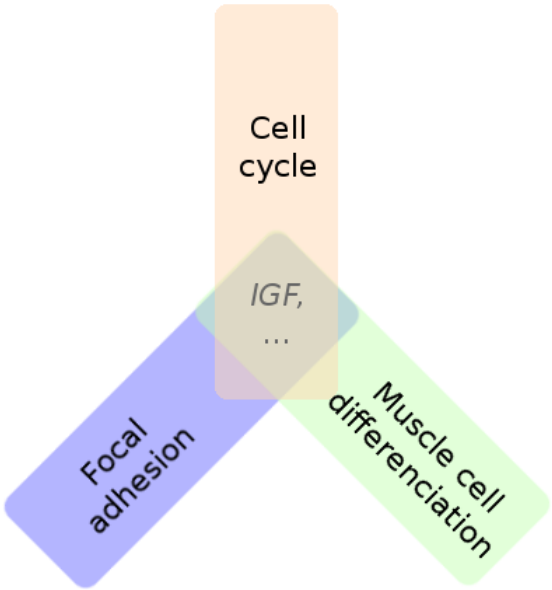


$$\Omega(w_1, w_2, w_3) = \|(w_1, w_2)\|_2 + \|w_3\|_2$$

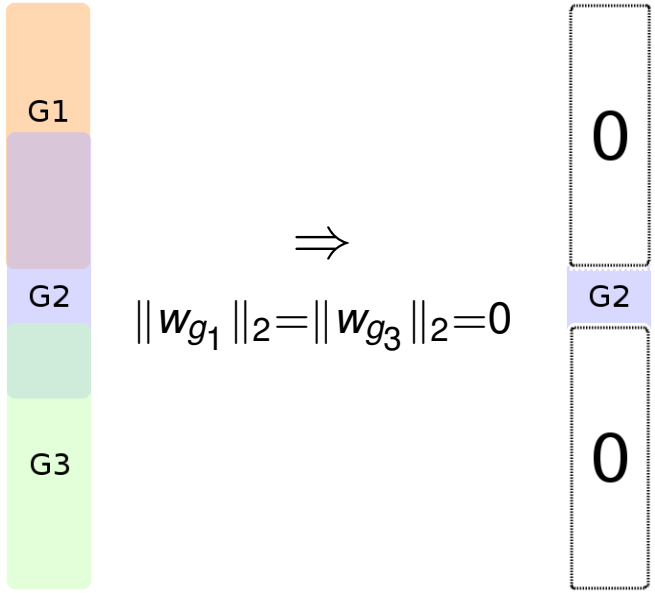
What if a gene belongs to several groups?

Issue of using the group-lasso

- $\Omega_{group}(w) = \sum_g \|w_g\|_2$ sets groups to 0.
- One variable is selected $\Leftrightarrow$ all the groups to which it belongs are selected.



IGF selection $\Rightarrow$ selection of unwanted groups



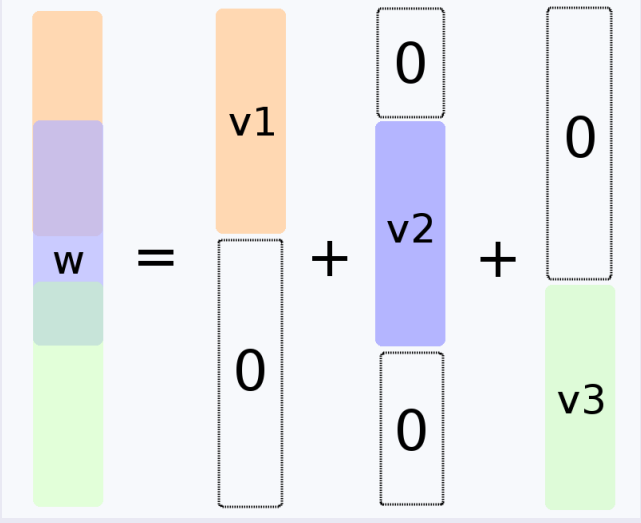
Removal of *any* group containing a gene $\Rightarrow$ the weight of the gene is 0.

Latent group lasso (Jacob et al., 2009)

An idea

Introduce latent variables v_g:

$$\begin{cases} \min\limits_{w,v} L(w) + \lambda \sum\limits_{g \in \mathcal{G}} \|v_g\|_2 \\ w = \sum_{g \in \mathcal{G}} v_g \\ \text{supp}\,(v_g) \subseteq g. \end{cases}$$



Properties

- Resulting support is a *union* of groups in $\mathcal{G}$.
- Possible to select one variable without selecting all the groups containing it.
- Equivalent to group lasso when there is no overlap

A new norm

Overlap norm

$$\begin{cases} \min\limits_{w,v} L(w) + \lambda \sum\limits_{g \in \mathcal{G}} \|v_g\|_2 \\ w = \sum_{g \in \mathcal{G}} v_g \\ \text{supp}\,(v_g) \subseteq g. \end{cases} = \min_{w} L(w) + \lambda \Omega_{overlap}(w)$$

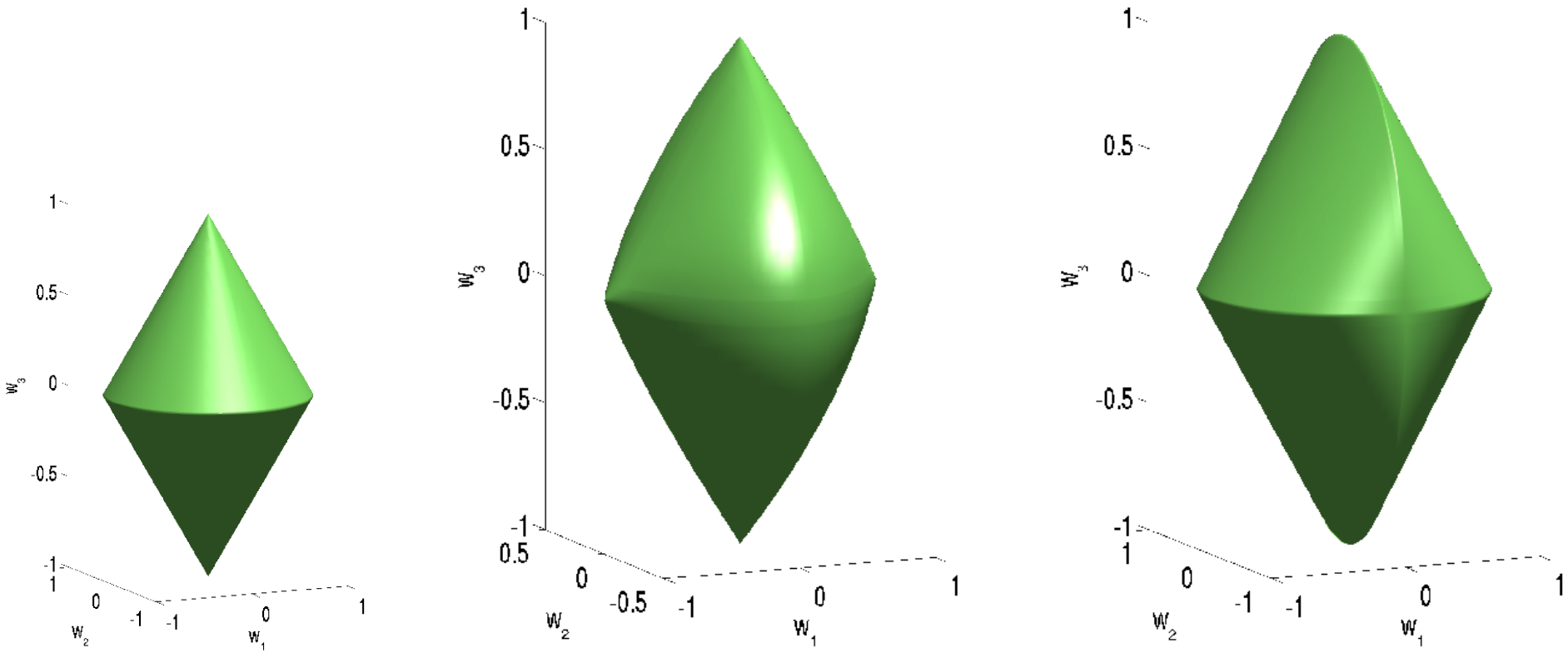
with

$$\Omega_{overlap}(w) \triangleq \begin{cases} \min\limits_{v} \sum\limits_{g \in \mathcal{G}} \|v_g\|_2 \\ w = \sum_{g \in \mathcal{G}} v_g \\ \text{supp}\,(v_g) \subseteq g. \end{cases} \quad (*)$$

Property

- $\Omega_{overlap}(w)$ is a norm of w.
- $\Omega_{overlap}(.)$ associates to w a specific (not necessarily unique) decomposition $(v_g)_{g \in \mathcal{G}}$ which is the argmin of $(*)$.

Overlap and group unity balls



Balls for $\Omega^{\mathcal{G}}_{\text{group}}(\cdot)$ (middle) and $\Omega^{\mathcal{G}}_{\text{overlap}}(\cdot)$ (right) for the groups $\mathcal{G} = \{\{1,2\},\{2,3\}\}$ where w_2 is represented as the vertical coordinate. Left: group-lasso ($\mathcal{G} = \{\{1,2\},\{3\}\}$), for comparison.

Theoretical results

Consistency in group support (Jacob et al., 2009)

- Let $\bar{w}$ be the true parameter vector.
- Assume that there exists a unique decomposition $\bar{v}_g$ such that $\bar{w} = \sum_g \bar{v}_g$ and $\Omega^{\mathcal{G}}_{\text{overlap}}(\bar{w}) = \sum \|\bar{v}_g\|_2$.
- Consider the regularized empirical risk minimization problem $L(w) + \lambda \Omega^{\mathcal{G}}_{\text{overlap}}(w)$.

Then

- under appropriate mutual incoherence conditions on X,
- as $n \to \infty$,
- with very high probability,

the optimal solution $\hat{w}$ admits a unique decomposition $(\hat{v}_g)_{g \in \mathcal{G}}$ such that

$$\left\{ g \in \mathcal{G} | \hat{v}_g \neq 0 \right\} = \left\{ g \in \mathcal{G} | \bar{v}_g \neq 0 \right\}.$$

Theoretical results

Consistency in group support (Jacob et al., 2009)

- Let $\bar{w}$ be the true parameter vector.
- Assume that there exists a unique decomposition $\bar{v}_g$ such that $\bar{w} = \sum_g \bar{v}_g$ and $\Omega^{\mathcal{G}}_{\text{overlap}}(\bar{w}) = \sum \|\bar{v}_g\|_2$.
- Consider the regularized empirical risk minimization problem $L(w) + \lambda \Omega^{\mathcal{G}}_{\text{overlap}}(w)$.

Then

- under appropriate mutual incoherence conditions on X,
- as $n \to \infty$,
- with very high probability,

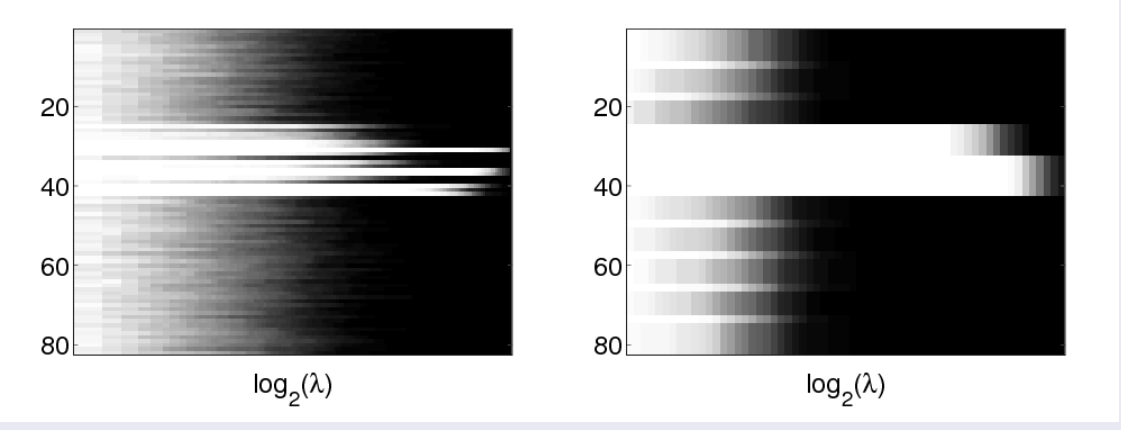
the optimal solution $\hat{w}$ admits a unique decomposition $(\hat{v}_g)_{g \in \mathcal{G}}$ such that

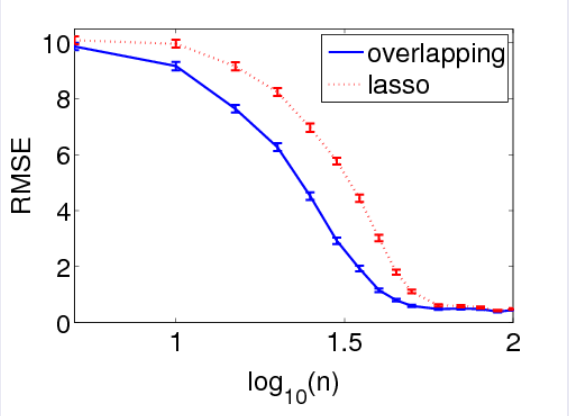
$$\left\{ g \in \mathcal{G} | \hat{v}_g \neq 0 \right\} = \left\{ g \in \mathcal{G} | \bar{v}_g \neq 0 \right\}.$$

Experiments

Synthetic data: overlapping groups

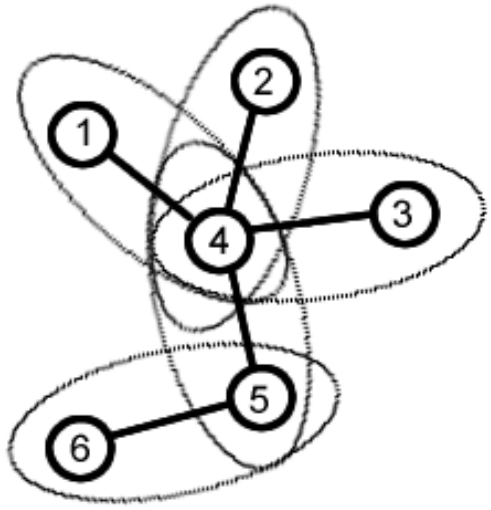
- 10 groups of 10 variables with 2 variables of overlap between two successive groups :$\{1, \ldots, 10\}, \{9, \ldots, 18\}, \ldots, \{73, \ldots, 82\}$.
- Support: union of 4*th* and 5*th* groups.
- Learn from 100 training points.





Frequency of selection of each variable with the lasso (left) and $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ (middle), comparison of the RMSE of both methods (right).

Graph lasso



Two solutions

$$\Omega_{intersection}(\beta) = \sum_{i \sim j} \sqrt{\beta_i^2 + \beta_j^2}\,,$$

$$\Omega_{union}(\beta) = \sup_{\alpha \in \mathbb{R}^p : \forall i \sim j, \|\alpha_i^2 + \alpha_j^2\| \leq 1} \alpha^\top \beta\,.$$

Graph lasso vs kernel on graph

- Graph lasso:

$$\Omega_{\text{graph lasso}}(w) = \sum_{i \sim j} \sqrt{w_i^2 + w_j^2}\,.$$

  constrains the sparsity, not the values
- Graph kernel

$$\Omega_{\text{graph kernel}}(w) = \sum_{i \sim j} (w_i - w_j)^2\,.$$

  constrains the values (smoothness), not the sparsity

Preliminary results

Breast cancer data

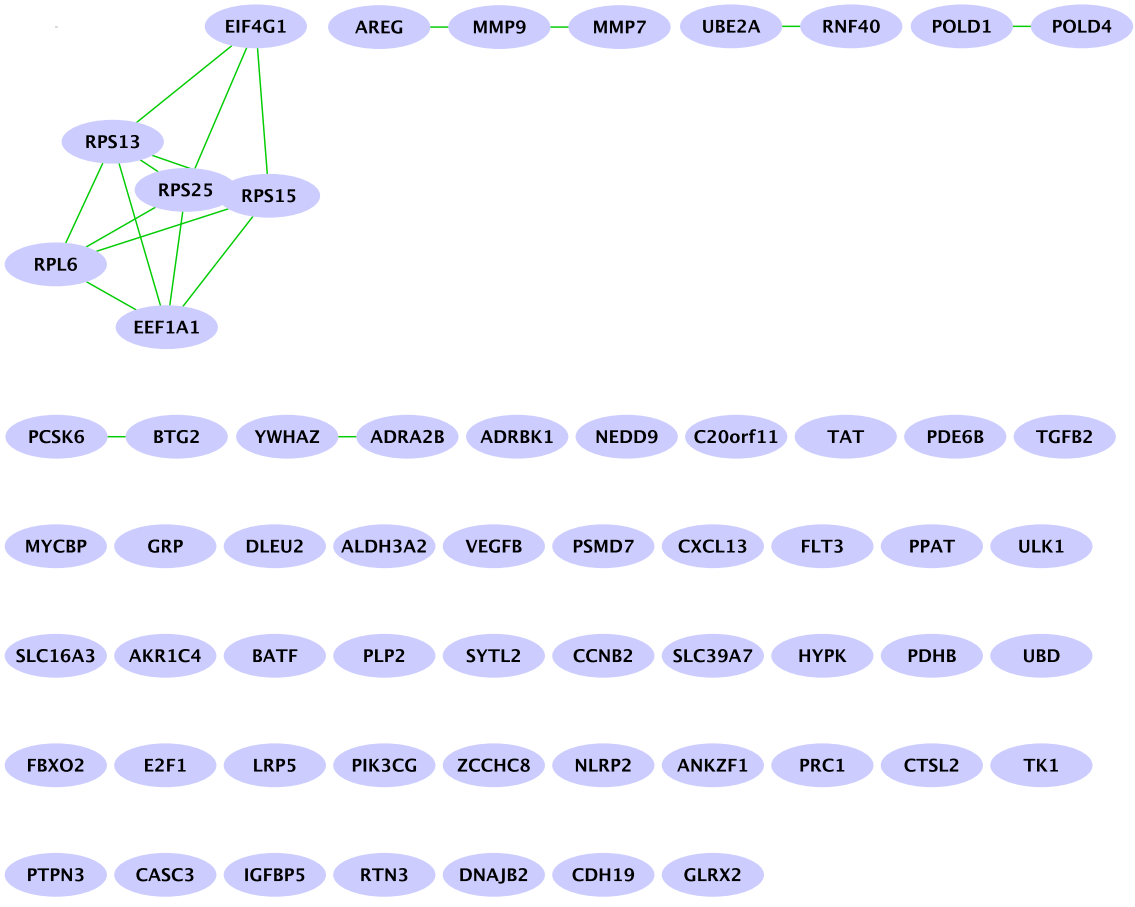
- Gene expression data for 8,141 genes in 295 breast cancer tumors.
- Canonical pathways from MSigDB containing 639 groups of genes, 637 of which involve genes from our study.

METHOD	ℓ_1	$\Omega^{\mathcal{G}}_{\text{OVERLAP}}(.)$
ERROR	0.38 ± 0.04	0.36 ± 0.03
MEAN ♯ PATH.	130	30

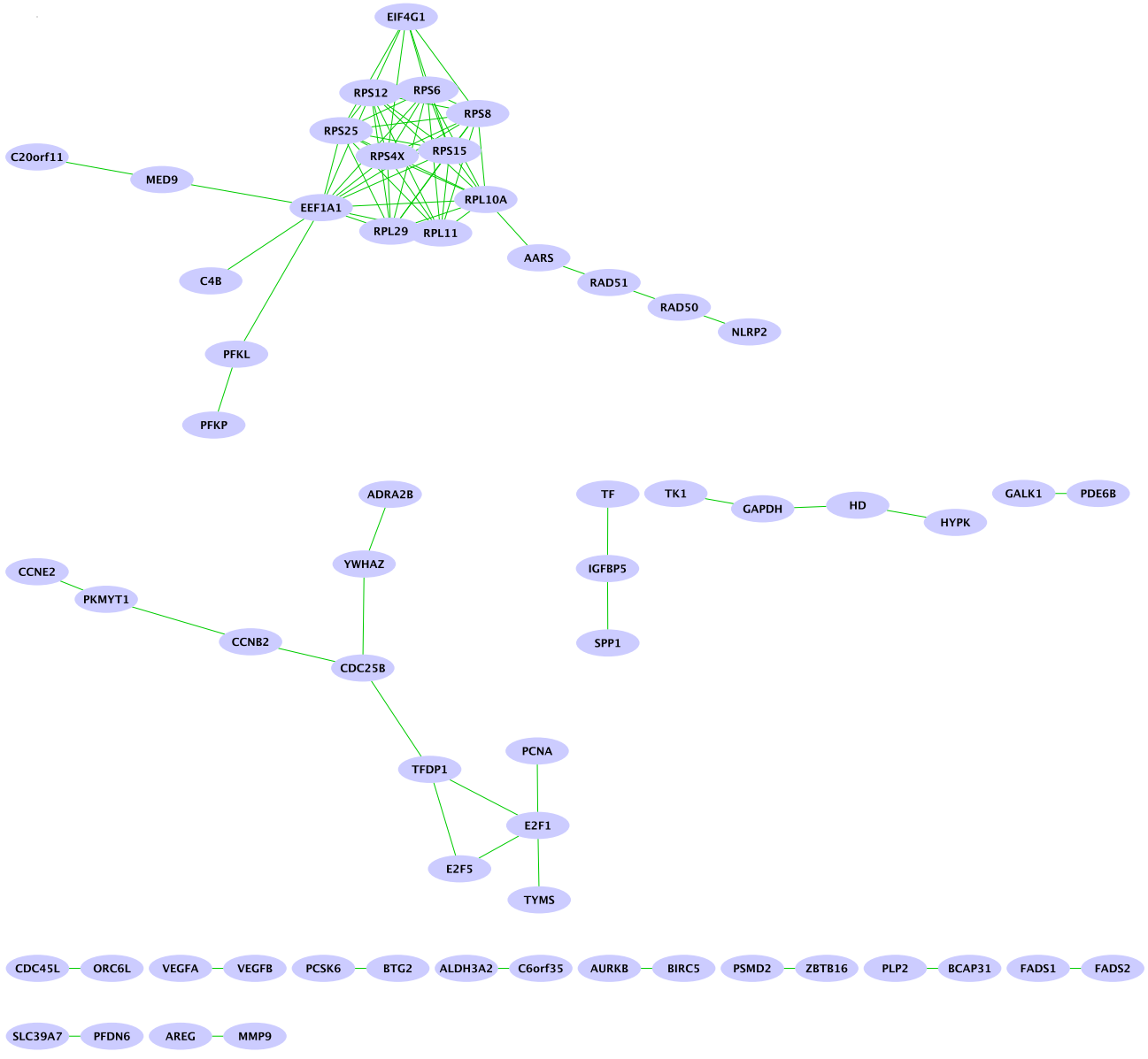
- Graph on the genes.

METHOD	ℓ_1	$\Omega_{graph}(.)$
ERROR	0.39 ± 0.04	0.36 ± 0.01
AV. SIZE C.C.	1.03	1.30

Lasso signature



Graph Lasso signature



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.

V. Boeva, A. Zinovyev, K. Bleakley, J.-P. Vert, I. Janoueix-Lerosey, O. Delattre, and E. Barillot. Control-free calling of copy number alterations in deep-sequencing data using GC-content normalization. *Bioinformatics*, 27(2):268–269, Jan 2011. doi: 10.1093/bioinformatics/btq635. URL http://dx.doi.org/10.1093/bioinformatics/btq635.

S. S. Chen, D. L. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998. doi: 10.1137/S1064827596304010. URL http://dx.doi.org/10.1137/S1064827596304010.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Stat.*, 32(2): 407–499, 2004. doi: 10.1214/009053604000000067. URL http://dx.doi.org/10.1214/009053604000000067.

J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Ann. Appl. Statist.*, 1(1):302–332, 2007. doi: 10.1214/07-AOAS131. URL http://dx.doi.org/10.1214/07-AOAS131.

G. Furnival and R. Wilson. Regressions by leaps and bounds. *Technometrics*, 16(4):499–511, 1974. URL http://www.jstor.org/stable/1267601.

References (cont.)

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 C. Levy-Leduc. Multiple change-point estimation with a total variation penalty. *J. Am. Stat. Assoc.*, 105(492):1480–1493, 2010. doi: 10.1198/jasa.2010.tm09181. URL http://dx.doi.org/10.1198/jasa.2010.tm09181.

H. Hoefling. A path algorithm for the Fused Lasso Signal Approximator. Technical Report 0910.0526v1, arXiv, Oct. 2009. URL http://arxiv.org/abs/0910.0526.

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.

References (cont.)

S. R. Land and J. H. Friedman. Variable fusion: A new adaptive signal regression method. Technical Report 656, Department of Statistics, Carnegie Mellon University Pittsburgh, 1997. URL http://www.stat.cmu.edu/tr/tr656/tr656.html.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, 2010. URL http://jmlr.csail.mit.edu/papers/v11/mairal10a.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.

F. Rapaport, E. Barillot, and J.-P. Vert. Classification of arrayCGH data using fused SVM. *Bioinformatics*, 24(13):i375–i382, Jul 2008. doi: 10.1093/bioinformatics/btn188. URL http://dx.doi.org/10.1093/bioinformatics/btn188.

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992. doi: 10.1016/0167-2789(92)90242-F. URL http://dx.doi.org/10.1016/0167-2789(92)90242-F.

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.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B*, 58(1): 267–288, 1996. URL http://www.jstor.org/stable/2346178.

References (cont.)

R. Tibshirani and P. Wang. Spatial smoothing and hot spot detection for cgh data using the fused lasso. *Biostatistics (Oxford, England)*, 9(1):18–29, January 2008. ISSN 1465-4644. doi: 10.1093/biostatistics/kxm013. URL http://dx.doi.org/10.1093/biostatistics/kxm013.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 67(1):91–108, 2005. URL http://ideas.repec.org/a/bla/jorssb/v67y2005i1p91-108.html.

J.-P. Vert and K. Bleakley. Fast detection of multiple change-points shared by many signals using group LARS. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Adv. Neural. Inform. Process Syst.*, volume 22, pages 2343–2352, 2010. URL http://books.nips.cc/papers/files/nips23/NIPS2010_1131.pdf.

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.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B*, 68(1):49–67, 2006. doi: 10.1111/j.1467-9868.2005.00532.x. URL http://dx.doi.org/10.1111/j.1467-9868.2005.00532.x.