Kernel Methods in Machine Learning

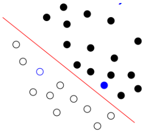
Julien Mairal & Jean-Philippe Vert

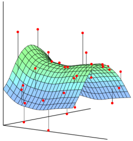
Romain Ménégaux (T.A.)

firstname.lastname@m4x.org

AMMI, May 2020

General learning framework





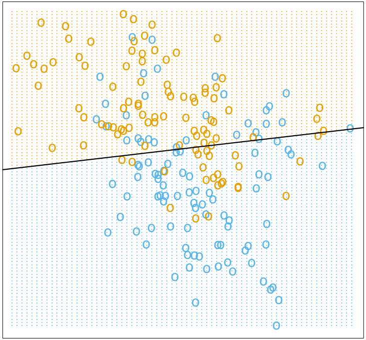
Input

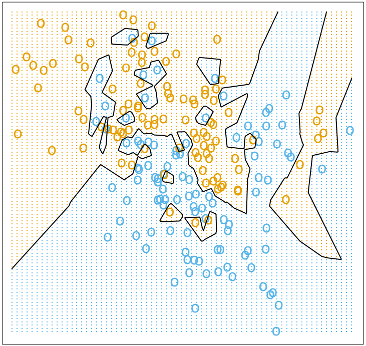
- $\mathcal{X}$ the space of patterns or data (typically, $\mathcal{X} = \mathbb{R}^p$)
- $\mathcal{Y}$ the space of response or labels
  - Classification or pattern recognition : $\mathcal{Y} = \{-1, 1\}$
  - Regression : $\mathcal{Y} = \mathbb{R}$
  - Structured output: $\mathcal{Y}$ general
- $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ 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 $f(x)$

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:

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

Important message

$$\text{Future prediction error} = \text{noise} + \text{bias}^2 + \text{variance}$$

- The "noise" part can not be avoided
- By choosing a learning model, we should consider both "bias" and "variance" if we want to make good predictions
- Intuitively, a more realistic, more complex model with more parameters to estimate has smaller bias but larger variance
- If variance dominates bias (eg, in high dimension), then having more complex, more realist models can hurt performance
- In other words, a wrong but simple model can work better than a more realistic but more complex model
- In many applications, domain experts (non-statisticians) often ignore the cost of complexity and prefer complex models, which can lead to disappointing results. You can help them!

Back to OLS

- Linear model with parameter $\beta \in \mathbb{R}^p$:

$$\forall x \in \mathbb{R}^p, \quad f_\beta(x) = \beta^\top x \quad \left(= \sum_{i=1}^{p} \beta_i x_i \right)$$

- Estimate $\hat{\beta}^{OLS}$ from training data to minimize the mean sum of squares (MSE):

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

Back to OLS (cont.)

- Let's use matrix notations:
  - $Y = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$ the vector of outcomes
  - $X = (x_1, \dots, x_n)^\top \in \mathbb{R}^{n \times p}$ the matrix (n rows=samples, p columns=features)
- We can rewrite MSE as

$$\mathrm{MSE}(\beta) = \frac{1}{n}(Y - X\beta)^\top (Y - X\beta)$$

- $\mathrm{MSE}(\beta)$ is a quadratic convex function; we minimize it by setting its gradient to 0:

$$\nabla_\beta \mathrm{MSE}(\beta) = \frac{2}{n} X^\top (X\beta - Y) = 0$$

- If $X^\top X$ is non-singular, the minimum is reached at

$$\hat{\beta}^{OLS} = \operatorname*{argmin}_\beta \mathrm{MSE}(\beta) = \left(X^\top X\right)^{-1} X^\top Y$$

Properties of OLS

Bias and variance of OLS

- Assume $Y = X\beta^* + \epsilon$, where $E\epsilon = 0$ and $E\epsilon\epsilon^\top = \sigma^2 I$.
- Then the least squares estimator

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

satisfies

$$\begin{cases} E\left(\hat{\beta}^{OLS}\right) = \beta^*, \\ Var(\hat{\beta}^{OLS}) = E\left(\hat{\beta}^{OLS} - \beta^*\right)\left(\hat{\beta}^{OLS} - \beta^*\right)^\top = \sigma^2 \left(X^\top X\right)^{-1}. \end{cases}$$

Proof: exercice

A solution: shrinkage estimators

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

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

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.:

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

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 minimum empirical risk, subject to some constraint:

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

for some penalty function $\Omega : \mathbb{R}^p \to \mathbb{R}^+$ and $C \geq 0$.

Equivalent formulation

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

is equivalent to

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

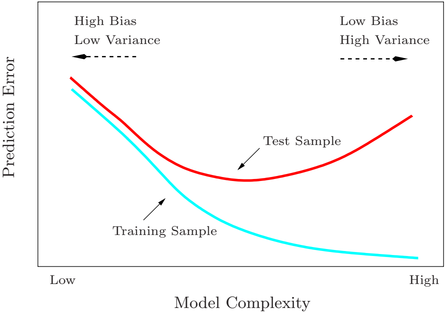
- There exists a (not necessarily unique) correspondance between C and λ such that the solutions to both problems are the same.
- If C increase, λ decreases
- The formulation with λ is often preferred to implement the algorithm
- Proof: using Lagrangian duality (only true under some assumptions, eg, R and Ω convex + Slater conditions, see later)

Choice of C or λ

- Choose a grid of values Λ for λ (or C)
- For each $\lambda \in \Lambda$ (or C) estimate the best model

$$\hat{\beta}_\lambda \in \underset{\beta}{\operatorname{argmin}}\ R(\beta) + \lambda\Omega(\beta)$$

- Select $\hat{\beta} = \hat{\beta}_{\hat{\lambda}}$ 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 modern machine learning involve models with many parameters (i.e., high dimension)
2. The total prediction 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 us to control the bias/variance trade-off
5. The choice of the penalty is where we can put prior knowledge to decrease bias
6. The parameter to control the bias-variance trade-off (C or λ) is typically chosen by cross-validation, to minimize the test error.

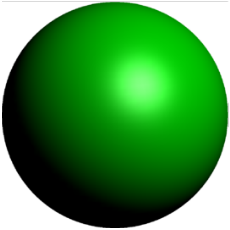
Outline

1. Learning in high dimension
2. Learning with ℓ_2 regularization
   - Ridge regression
   - Ridge logistic regression
   - Linear hard-margin SVM
   - Interlude: quick notes on constrained optimization
   - Back to hard-margin SVM
   - Soft-margin SVM
   - Large-margin classifiers
3. Learning with kernels
   - Kernel methods
   - Positive definite kernels and RKHS
   - Kernel examples
   - Multiple Kernel Learning (MKL)
4. Conclusion

Overview

- We focus on a simple penalty function: the squared Euclidean norm

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



- This will allow us to derive many state-of-the-art linear methods:
  - Ridge regression
  - Ridge logistic regression
  - SVM and large-margin classifiers
- This will allow us to extend these linear methods to nonlinear models, using kernels

Outline

Ridge regression (?)

1. Consider the set of linear predictors:

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

Ridge regression (?)

1. Consider the set of linear predictors:

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

2. Consider the MSE as empirical risk:

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

Ridge regression (?)

1. Consider the set of linear predictors:

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

2. Consider the MSE as empirical risk:

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

3. Consider the squared Euclidean norm as a penalty:

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

Solution

- The penalized risk can be written in matrix form:

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

Solution

- The penalized risk can be written in matrix form:

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

- Unique minimizer (by setting the gradient to 0):

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

Performance of ridge regression

Lemma

Assume that:

- $Y = X\beta^* + \epsilon$, where $E\epsilon = 0$ and $E\epsilon\epsilon^\top = \sigma^2 I$.
- $X^\top X = nI_p$ (orthogonal design)

Then:

$$\begin{cases} bias\left(\hat{\beta}_\lambda^{\text{ridge}}\right) = E\left(\hat{\beta}_\lambda^{\text{ridge}}\right) - \beta^* = -\frac{\lambda}{1+\lambda}\beta^*\,, \\ Var(\hat{\beta}_\lambda^{\text{ridge}}) = \frac{\sigma^2}{n(1+\lambda)^2} I_p = \frac{1}{(1+\lambda)^2} Var(\hat{\beta}^{OLS})\,. \end{cases}$$

Proof: exercice

Performance of ridge regression

Corollary

For any $\lambda \geq 0$ let

$$f(\lambda) = E_{\mathcal{S},x_0}\left[bias^2\left(x_0^\top \hat{\beta}_\lambda^{\text{ridge}}\right) + Var\left(x_0^\top \hat{\beta}_\lambda^{\text{ridge}}\right)\right]$$

where $Ex_0 = 0, Ex_0x_0^\top = I_p$. Then $f(\lambda)$ is minimum for

$$\lambda^* = \frac{\sigma^2 p}{n\|\beta^*\|^2}$$

and

$$f(\lambda^*) = \frac{f(0)f(+\infty)}{f(0) + f(+\infty)} \leq \min\{f(0), f(\infty)\}$$

where

$$f(0) = \sigma^2 p/n\,, \quad f(\infty) = \|\beta^*\|^2$$

Proof: exercice

Limit cases

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

Corollary

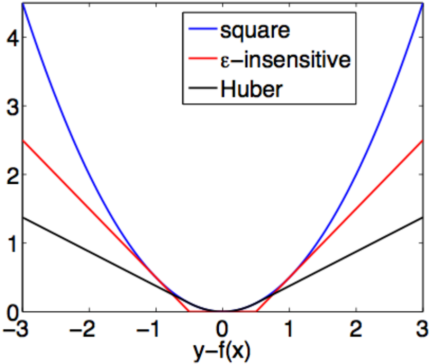
- As $\lambda \to 0$, $\hat{\beta}_\lambda^{\text{ridge}} \to \hat{\beta}^{\text{OLS}}$ (low bias, high variance).
- As $\lambda \to +\infty$, $\hat{\beta}_\lambda^{\text{ridge}} \to 0$ (high bias, low variance).

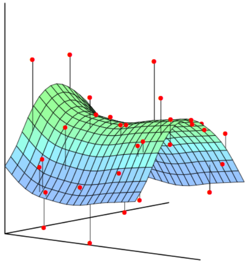
Generalization: ℓ_2-regularized learning

- A general ℓ_2-penalized estimator is of the form
$$\min_{\beta} \left\{ R(\beta) + \lambda \| \beta \|^2 \right\} , \tag{1}$$
where
$$R(\beta) = \frac{1}{n} \sum_{i=1}^{n} \ell(f_\beta(x_i), y_i)$$
for some general loss functions ℓ.
- Ridge regression corresponds to the particular loss
$$\ell(u, y) = (u - y)^2 .$$
- For general, convex losses, the problem (**??**) is strictly convex and has a unique global minimum, which can usually be found by numerical algorithms for convex optimization.

Losses for regression

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



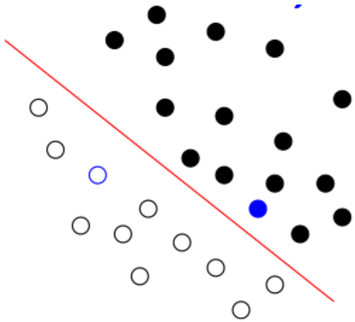


Outline

Binary classification

Setting

- $\mathcal{X} = \mathbb{R}^p$ set of inputs
- $\mathcal{Y} = \{-1, 1\}$ binary outputs
- $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ a training set in $(\mathcal{X} \times \mathcal{Y})^n$
- Goal: Estimate a function $f : \mathcal{X} \to \mathbb{R}$ to predict y by $sign(f(x))$



The 0/1 loss

- The 0/1 loss measures if a prediction is correct or not:

$$\ell_{0/1}(f(x), y)) = \mathbf{1}(yf(x) < 0) = \begin{cases} 0 & \text{if } y = sign(f(x)) \\ 1 & \text{otherwise.} \end{cases}$$

- It is them tempting to learn $f_\beta(x) = \beta^\top x$ by solving:

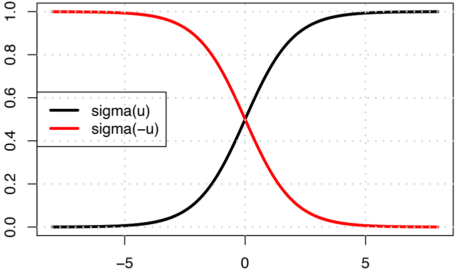
$$\min_{\beta \in \mathbb{R}^p} \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell_{0/1}\left(f_\beta\left(x_i\right), y_i\right)}_{\text{misclassification rate}} + \underbrace{\lambda \| \beta \|^2}_{\text{regularization}}$$

- However:
  - The problem is non-smooth, and typically NP-hard to solve
  - The regularization has no effect since the 0/1 loss is invariant by scaling of β
  - In fact, no function achieves the minimum when $\lambda > 0$ (*why?*)

The logistic loss

- An alternative is to define a probabilistic model of y parametrized by $f(x)$, e.g.:

$$\forall y \in \{-1, 1\}, \quad p(y \mid f(x)) = \frac{1}{1 + e^{-yf(x)}} = \sigma(yf(x))$$



- The logistic loss is the negative conditional likelihood:

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

Ridge logistic regression (?)

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

- Can be interpreted as a regularized conditional maximum likelihood estimator
- No explicit solution, but smooth convex optimization problem that can be solved numerically

Solving ridge logistic regression

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

No explicit solution, but convex problem with:

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

Outline

Loss functions for classifications

We already saw 3 loss functions for binary classification problems

- The 0/1 loss $\ell_{0/1}(f(x), y) = \mathbf{1}(yf(x) < 0)$
- The logistic loss $\ell_{logistic}(f(x), y) = \ln\left(1 + e^{-yf(x)}\right)$
- The hinge loss $\ell_{hinge}(f(x), y) = \max(0, 1 - yf(x))$

Definition

In binary classification ($\mathcal{Y} = \{-1, 1\}$), the margin of the function f for a pair (x, y) is:

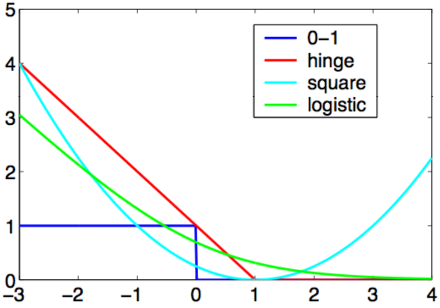
$$yf(x).$$

In all cases the loss is a decreasing function of the margin, i.e.,

$$\ell(f(x), y) = \varphi(yf(x)), \quad \text{with } \varphi \text{ non-increasing}$$

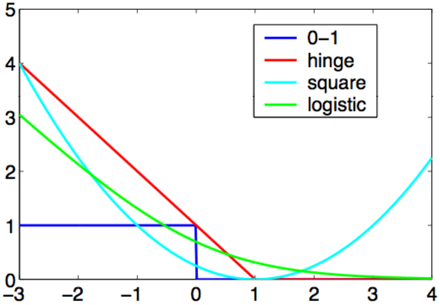
What about other similar loss functions?

Loss function examples



Method	$\varphi(u)$
Logistic regression	$\log\left(1+e^{-u}\right)$
Support vector machine (1-SVM)	$\max\left(1-u,0\right)$
Support vector machine (2-SVM)	$\max\left(1-u,0\right)^2$
Boosting	e^{-u}

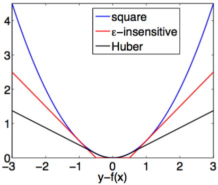
Summary: large margin classifiers

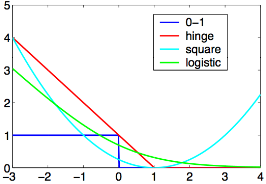


$$\min_{f \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \varphi\left(y_i f\left(x_i\right)\right) + \lambda \| f \|_{\mathcal{H}}^2 \right\}$$

- φ calibrated (e.g., decreasing, $\varphi'(0) < 0$) $\implies$ good proxy for classification error
- φ convex + representer theorem $\implies$ efficient algorithms
- φ smooth (Lipschitz) + ℓ_2 regularization $\implies$ good learning ability

Summary: ℓ_2-regularized linear methods



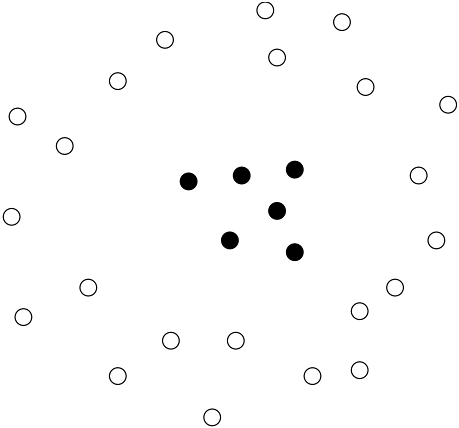


$$f_\beta(x) = \beta^\top x\,, \quad \min_\beta \left\{ \frac{1}{n}\sum_{i=1}^{n} \ell(f_\beta(x_i), y_i) + \lambda \|\beta\|_2^2 \right\}$$

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

Outline

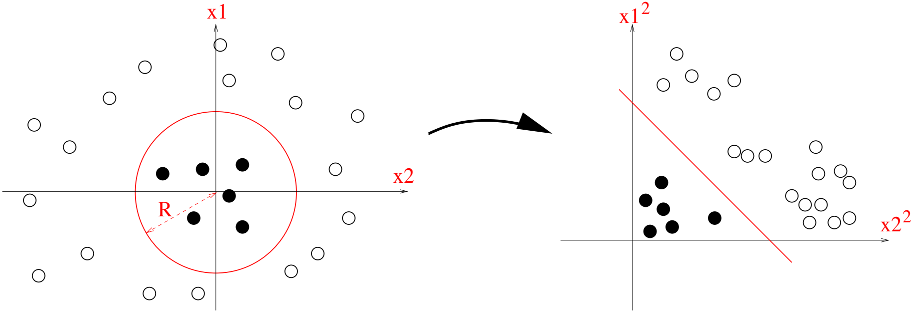
Motivation



- Sometimes linear models are not interesting...
- Kernels will allow to solve nonlinear problems with linear methods!

Outline

"Linear" depends on the representation you choose



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

$$f(x) = x_1^2 + x_2^2 - R^2 = \beta^\top \Phi(x) + b$$

with $\beta = (1,1)^\top$ and $b = -R^2$

Kernel = inner product in the feature space

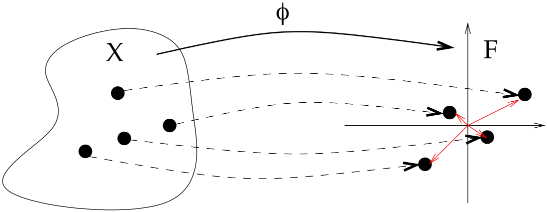
Definition

For a given mapping

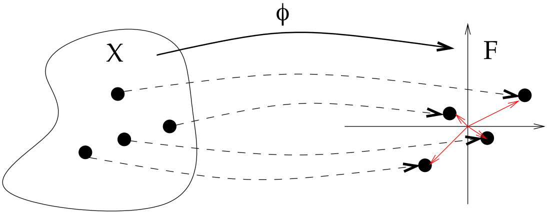
$$\Phi : \mathcal{X} \mapsto \mathcal{H}$$

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

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



Example

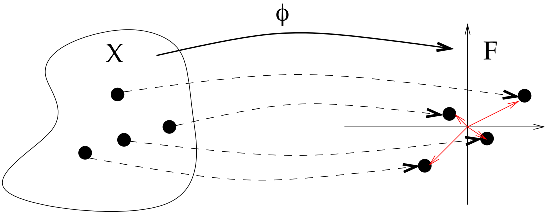


Let $\mathcal{X} = \mathcal{H} = \mathbb{R}^2$ and for $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ let $\Phi(x) = \begin{pmatrix} x_1^2 \\ x_2^2 \end{pmatrix}$

Then the kernel is:

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

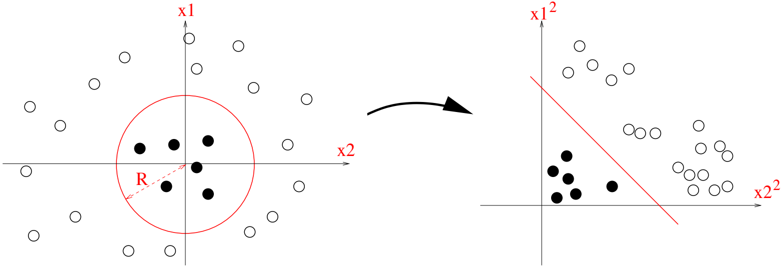
The kernel tricks



2 tricks

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

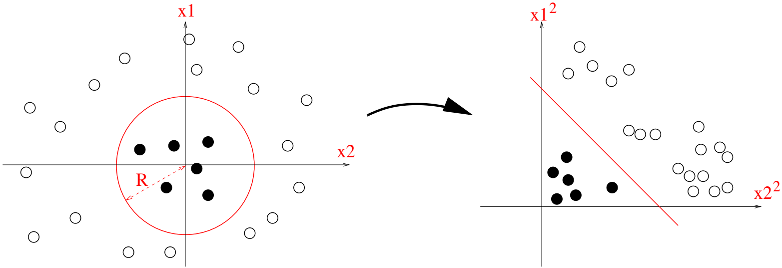
Trick 2 illustration: polynomial kernel



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

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

Trick 2 illustration: polynomial kernel



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

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

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

More generally: trick 1 for ℓ_2-regularized linear models

Representer theorem

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

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

can be expanded as

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

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

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

Representer theorem: proof

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

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

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

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

Example: kernel ridge regression

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

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

can be expanded as:

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

Example: kernel ridge regression

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

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

- Moreover,

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

Example: kernel ridge regression

- The problem is therefore equivalent to:

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

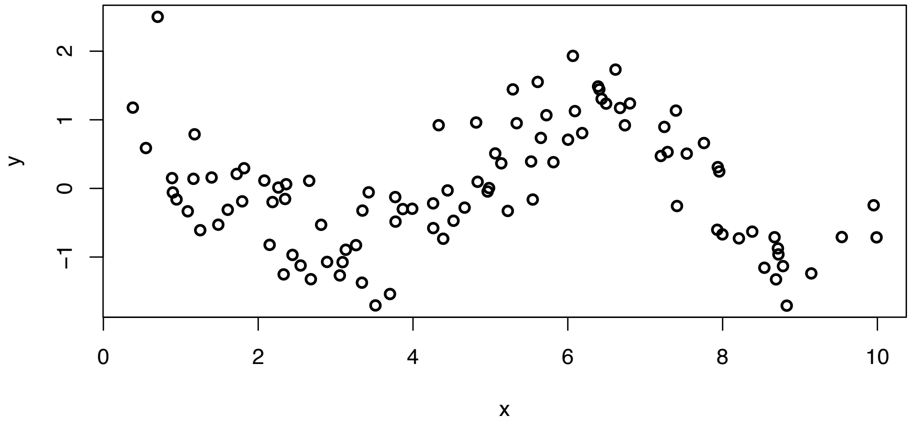
- This is a convex and differentiable function of α. Its minimum can therefore be found by setting the gradient in α to zero:

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

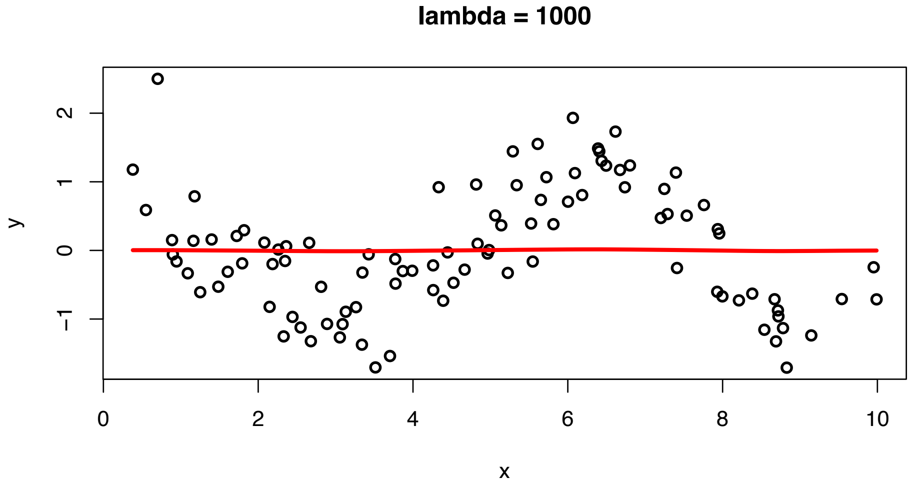
- For $\lambda > 0$, $K + \lambda n I$ is invertible (because K is positive semidefinite) so one solution is to take:

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

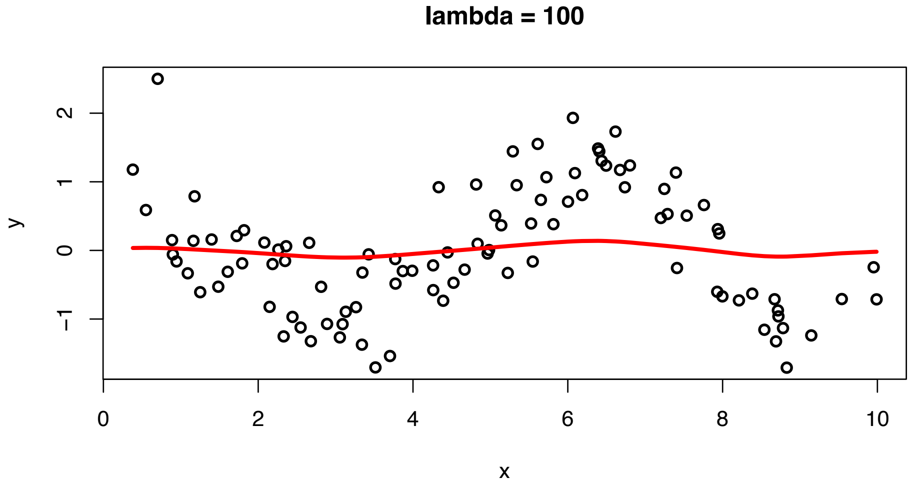
Example (KRR with Gaussian RBF kernel)



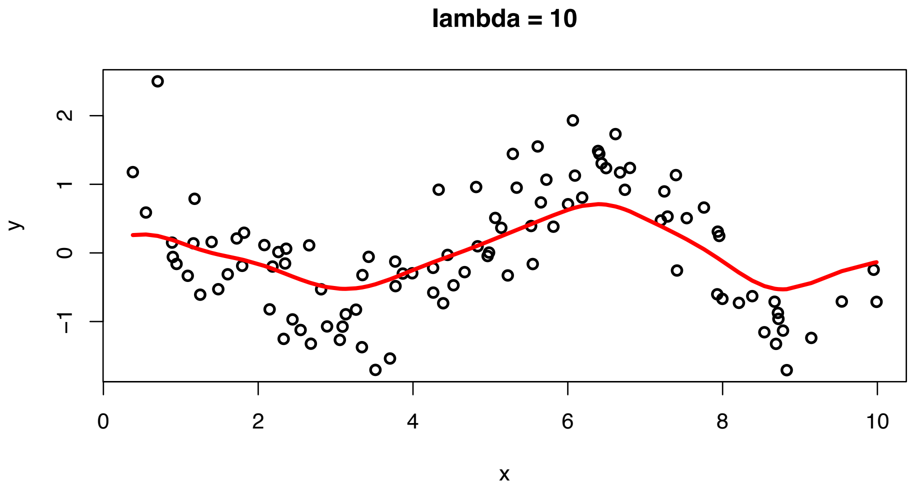
Example (KRR with Gaussian RBF kernel)



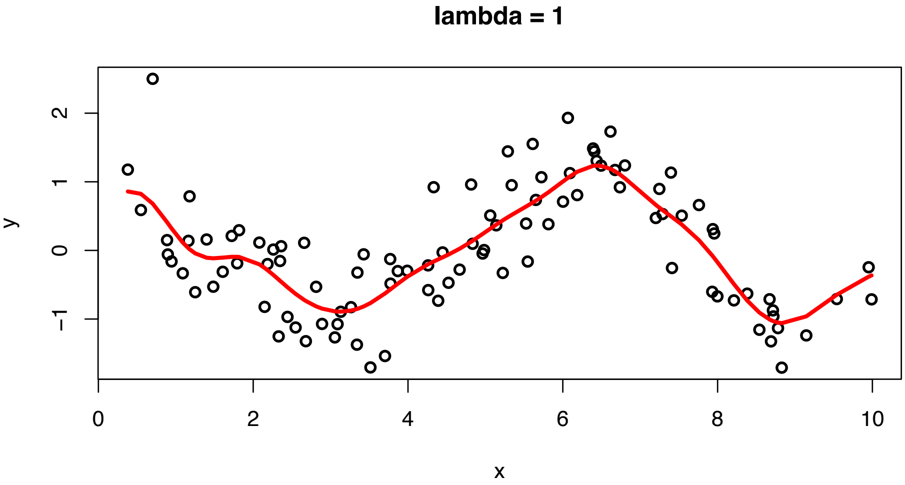
Example (KRR with Gaussian RBF kernel)



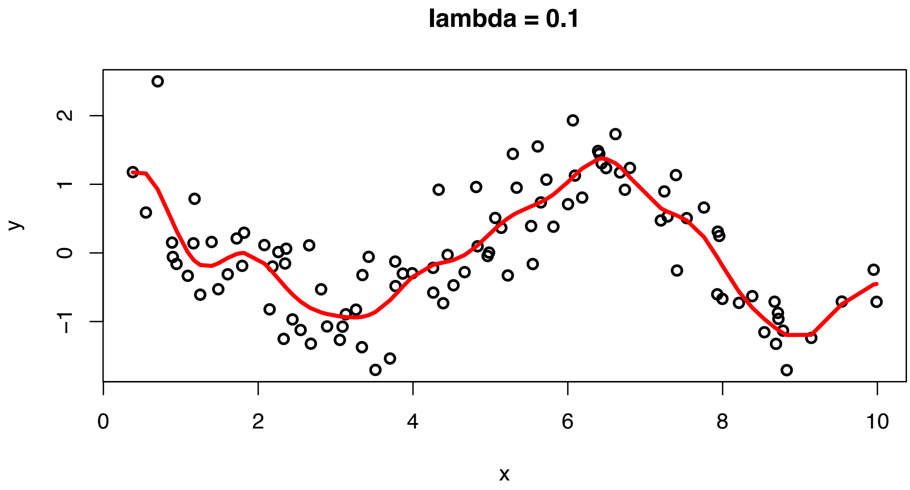
Example (KRR with Gaussian RBF kernel)



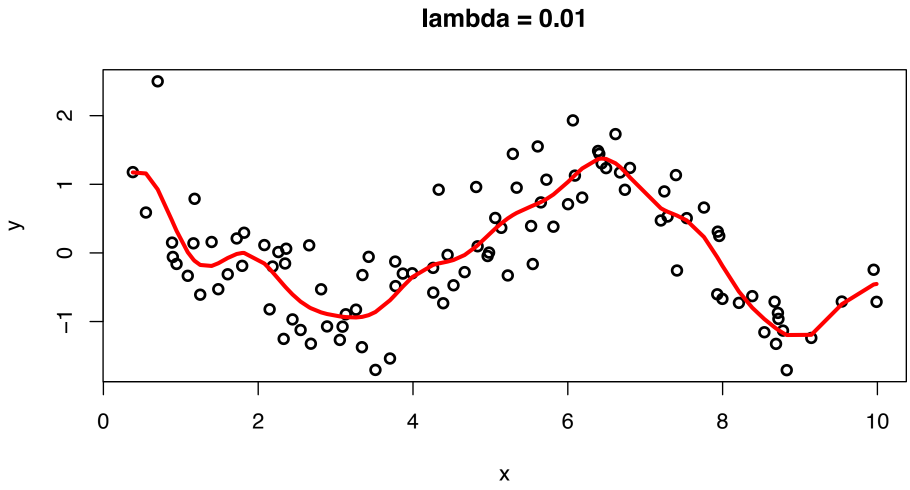
Example (KRR with Gaussian RBF kernel)



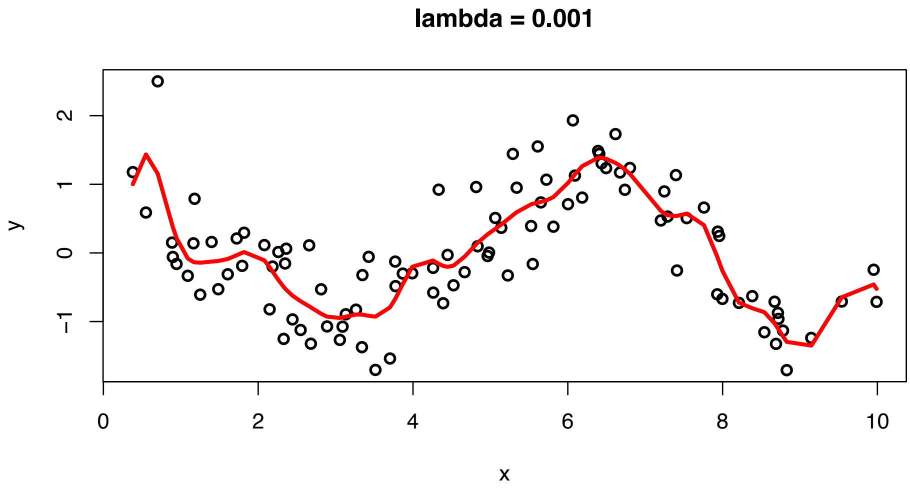
Example (KRR with Gaussian RBF kernel)



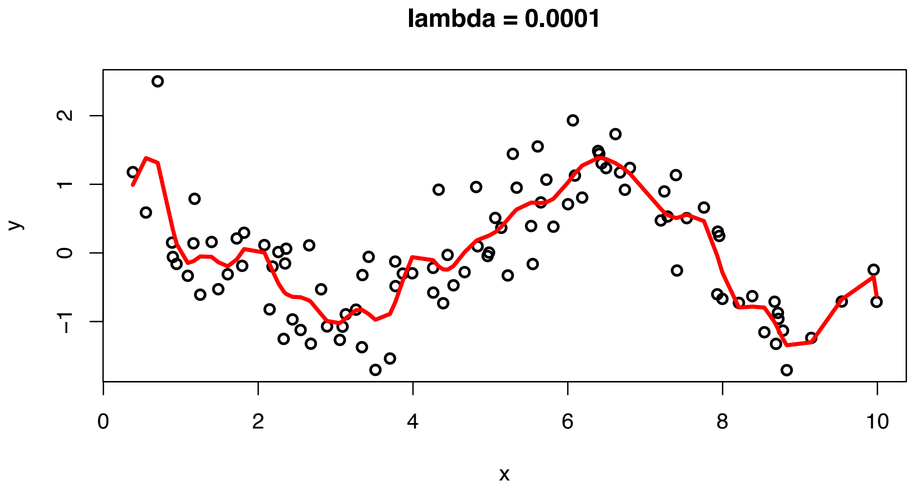
Example (KRR with Gaussian RBF kernel)



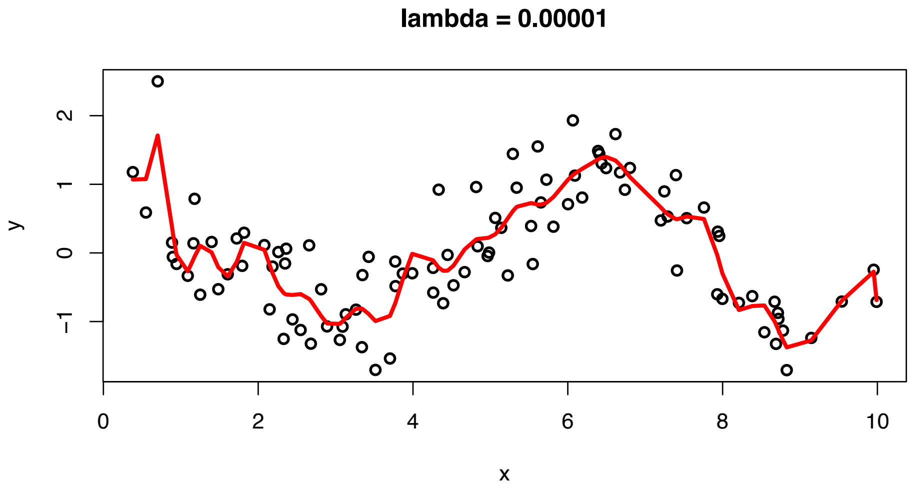
Example (KRR with Gaussian RBF kernel)



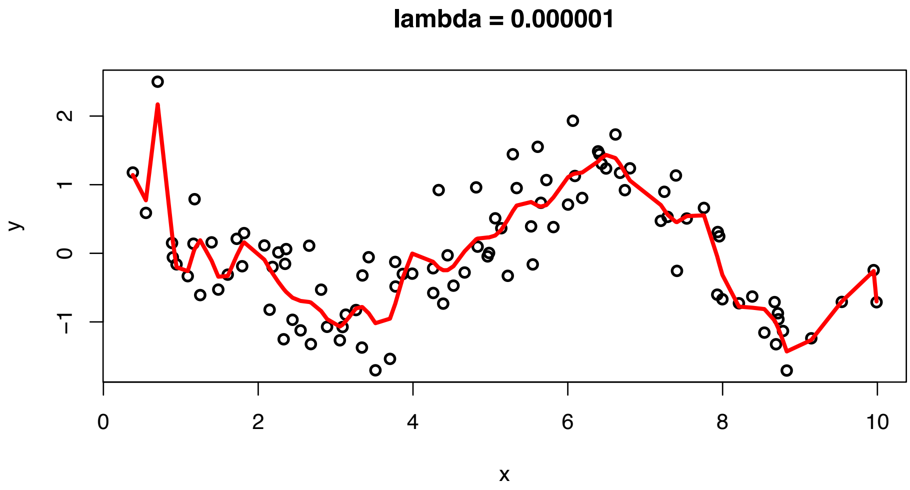
Example (KRR with Gaussian RBF kernel)



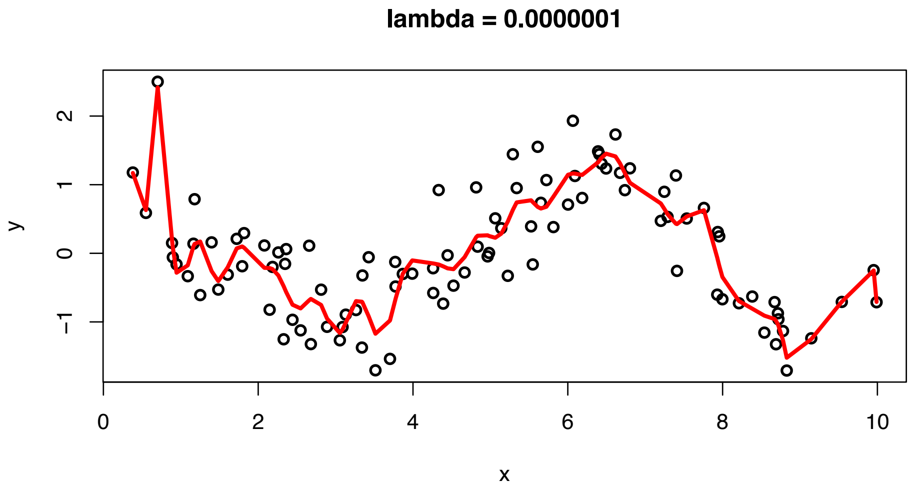
Example (KRR with Gaussian RBF kernel)



Example (KRR with Gaussian RBF kernel)



Example (KRR with Gaussian RBF kernel)



Remark: uniqueness of the solution

Let us find *all* α's that solve

$$K\left[(K+\lambda n I)\,\alpha - Y\right]=0$$

- 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:

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

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

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

  therefore $\beta = \beta'$. KRR has a unique solution β, which can possibly be expressed by several α's if K is singular.

Comparison with "standard" ridge regression

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

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

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

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

- Oups... which one is correct?

Comparison with "standard" ridge regression

Matrix inversion lemma

For any matrices B and C, and $\gamma > 0$ the following holds (when it makes sense):

$$B\left(CB + \gamma I\right)^{-1} = \left(BC + \gamma I\right)^{-1} B$$

We deduce that (of course...):

$$\hat{\beta} = \underbrace{\left(X^\top X + n\lambda I\right)^{-1}}_{p \times p} X^\top Y = X^\top \underbrace{\left(XX^\top + \lambda n I\right)^{-1}}_{n \times n} Y = \tilde{\beta}$$

Computationally, inverting the matrix is the expensive part, which suggest to implement:

- KRR when $p > n$ (high dimension)
- RR when $p < n$ (many points)

Generalization

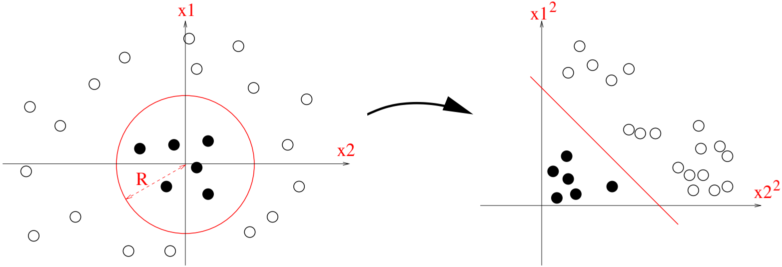
- We learn the function $f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)$ by solving in α the following optimization problem, with adequate loss function ℓ:

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

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

Outline

Remember: polynomial kernel



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

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

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

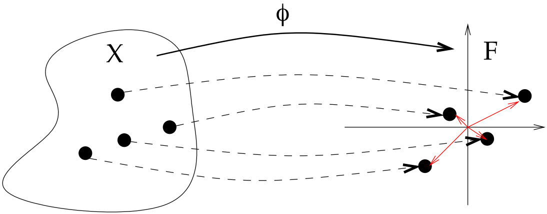
Definition

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

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

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

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



Reminder ...

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

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 $\mathbb{R}_+$)

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

Example: SVM with a Gaussian kernel

- Training:

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

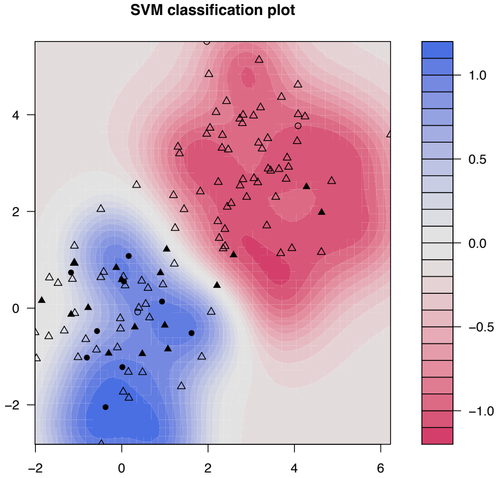
$$\text{s.t. } 0 \leq \alpha_i \leq C, \quad \text{and } \sum_{i=1}^{n} \alpha_i y_i = 0.$$

- Prediction

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

Example: SVM with a Gaussian kernel

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



Positive Definite (p.d.) functions

Definition

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

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

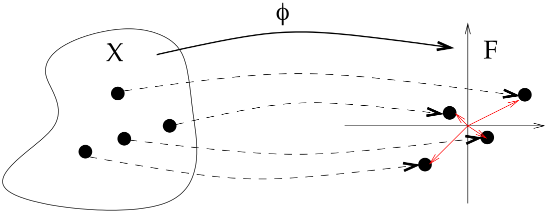
and which satisfies, for all $N \in \mathbb{N}$, $(x_1, x_2, \ldots, x_N) \in \mathcal{X}^N$ et $(a_1, a_2, \ldots, a_N) \in \mathbb{R}^N$:

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

Kernels are p.d. functions

Theorem (Aronszajn, 1950)

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



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

Let

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

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

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

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

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

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

with

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

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

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

RKHS

Definition

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

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

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

An equivalent definition of RKHS

Theorem

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

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

is continuous.

An equivalent definition of RKHS

Theorem

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

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

is continuous.

Corollary

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

Proof

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

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

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

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

Proof (Converse)

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

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

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

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

Unicity of r.k. and RKHS

Theorem

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

Unicity of r.k. and RKHS

Theorem

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

Consequence

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

Proof

If a r.k. exists then it is unique

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

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

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

Proof

If a r.k. exists then it is unique

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

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

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

The RKHS of a r.k. K is unique

Left as exercice.

An important result

Theorem

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

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

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

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

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

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

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

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

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

let:

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

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

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

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

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

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

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

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

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

  In particular Cauchy-Schwarz is valid with $\langle ., . \rangle_{\mathcal{H}_0}$.

- By Cauchy-Schwarz we deduce that $\forall x \in \mathcal{X}$:

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

  therefore $\| f \|_{\mathcal{H}_0} = 0 \implies f = 0$.

- $\mathcal{H}_0$ is therefore a pre-Hilbert space endowed with the inner product $\langle ., . \rangle_{\mathcal{H}_0}$.

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

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

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

  Therefore for any x the sequence $(f_n(x))_{n\geq 0}$ is Cauchy in $\mathbb{R}$ and has therefore a limit.

- If we add to $\mathcal{H}_0$ the functions defined as the pointwise limits of Cauchy sequences, then the space becomes complete and is therefore a Hilbert space, with K as r.k. (up to a few technicalities, left as exercice). $\square$

Application: back to Aronszajn's theorem

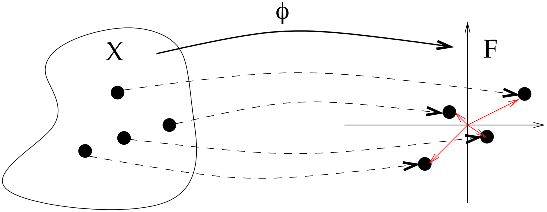
Theorem (Aronszajn, 1950)

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

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

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

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



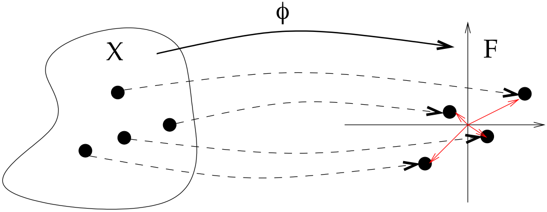
Proof of Aronzsajn's theorem: p.d. $\implies$ kernel

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

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

- By the reproducing property we have:

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



RKHS of the linear kernel

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

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

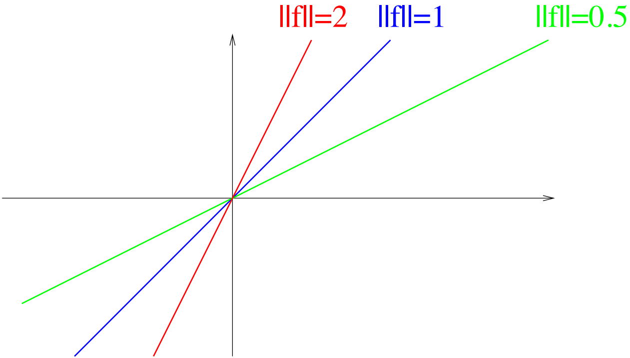
  with $w = \sum_i a_i x_i$.
- The RKHS is therefore the set of linear forms endowed with the following inner product:

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

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

RKHS of the linear kernel (cont.)

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



ℓ_2-regularized methods in RKHS

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

is equivalent to

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

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

Smoothness functional

A simple inequality

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

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

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

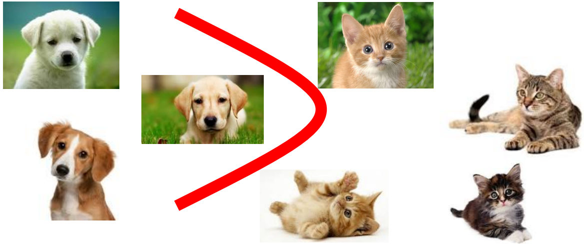
Important message

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

Kernels and RKHS : Summary for supervised learning

The goal is to learn a **prediction function** $f : \mathcal{X} \to \mathcal{Y}$ given labeled training data $(x_i, y_i)_{i=1,\dots,n}$ with x_i in $\mathcal{X}$, and y_i in $\mathcal{Y}$:

$$\min_{f \in \mathcal{F}} \quad \underbrace{\frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \Omega(f)}_{\text{regularization}} .$$



Kernels and RKHS : Summary for supervised learning

The goal is to learn a **prediction function** $f : \mathcal{X} \to \mathcal{Y}$ given labeled training data $(x_i, y_i)_{i=1,\dots,n}$ with x_i in $\mathcal{X}$, and y_i in $\mathcal{Y}$:

$$\min_{f \in \mathcal{F}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \Omega(f)}_{\text{regularization}} .$$

The labels y_i are in

- $\{-1, +1\}$ for **binary** classification problems.
- $\{1, \dots, K\}$ for **multi-class** classification problems.
- $\mathbb{R}$ for **regression** problems.
- $\mathbb{R}^k$ for **multivariate regression** problems.

Kernels and RKHS : Summary for supervised learning

The goal is to learn a **prediction function** $f : \mathcal{X} \rightarrow \mathcal{Y}$ given labeled training data $(x_i, y_i)_{i=1,\ldots,n}$ with x_i in $\mathcal{X}$, and y_i in $\mathcal{Y}$:

$$\min_{f \in \mathcal{F}} \underbrace{\frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \Omega(f)}_{\text{regularization}} .$$

Example with linear models: logistic regression, SVMs, *etc.*

- assume there exists a linear relation between y and features x in $\mathbb{R}^p$.
- $f(x) = w^\top x + b$ is parametrized by w, b in $\mathbb{R}^{p+1}$;
- L is often a **convex** loss function;
- $\Omega(f)$ is often the squared ℓ_2-norm $\|w\|^2$.

Kernels and RKHS : Summary for supervised learning

The goal is to learn a **prediction function** $f : \mathcal{X} \rightarrow \mathcal{Y}$ given labeled training data $(x_i, y_i)_{i=1,\ldots,n}$ with x_i in $\mathcal{X}$, and y_i in $\mathcal{Y}$:

$$\min_{f \in \mathcal{F}} \underbrace{\frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i))}_{\text{empirical risk, data fit}} + \underbrace{\lambda \Omega(f)}_{\text{regularization}} .$$

Remark about multilayer neural networks

- The "neural network" space $\mathcal{F}$ is explicitly parametrized by:

$$f(\mathbf{x}) = \sigma_k(\mathbf{A}_k \sigma_{k-1}(\mathbf{A}_{k-1} \ldots \sigma_2(\mathbf{A}_2 \sigma_1(\mathbf{A}_1 \mathbf{x})) \ldots)).$$

- Linear operations are either unconstrained (fully connected) or involve parameter sharing (e.g., convolutions).
- Finding the optimal $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k$ yields a **non-convex** optimization problem.

Kernels and RKHS : Summary for supervised learning

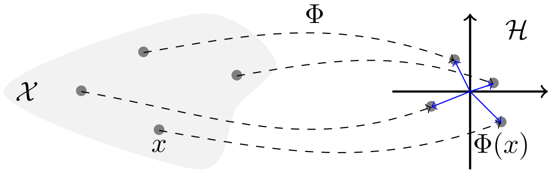
A classical kernel formulation for supervised learning

$$\min_{f \in \mathcal{H}} \quad \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) \; + \; \lambda \|f\|_{\mathcal{H}}^2.$$

- **map** data x in $\mathcal{X}$ to a Hilbert space and work with **linear forms**:

$$\Phi : \mathcal{X} \to \mathcal{H} \qquad \text{and} \qquad f(x) = \langle \Phi(x), f \rangle_{\mathcal{H}}.$$

- This is done implicitly with a positive definite kernel $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$!



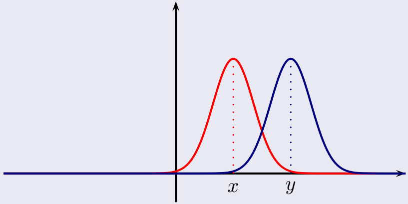
Kernels and RKHS : Summary for supervised learning

What does it mean to map a data point to a function?

Ex: if x, y in $\mathbb{R}$ and $K(x,y) = e^{-\frac{1}{\sigma^2}(x-y)^2}$ is the Gaussian kernel,

$$\Phi(x) : t \mapsto e^{-\frac{1}{\sigma^2}(x-t)^2}$$

$$\Phi(y) : t \mapsto e^{-\frac{1}{\sigma^2}(y-t)^2}$$



- Data points are mapped to Gaussian functions living in a Hilbert space $\mathcal{H}$.
- But $\mathcal{H}$ is much richer and contains much more than Gaussian functions!
- Prediction functions f live in $\mathcal{H}$: $f(x) = \langle f, \Phi(x) \rangle$.

Kernels and RKHS : Summary

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

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

Outline

Kernel examples

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

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

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

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

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

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

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

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

Exercice

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

How to choose or make a kernel?

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

Outline

In one slide...

- Learning in high dimension requires regularization, e.g., by ℓ_2 penalty for linear methods
- Kernels allow to transform any ℓ_2-regularized linear models into a nonlinear model, thanks to the kernel trick
- There exists many kernels, which correspond to different feature spaces (of finite or infinite dimensions)
- We can combine and learn kernels, e.g., for integration of heterogeneous data
- Hot research topics
  - Large-scale ML with kernels
  - Deep kernel methods

In one slide...

- Learning in high dimension requires regularization, e.g., by ℓ_2 penalty for linear methods
- Kernels allow to transform any ℓ_2-regularized linear models into a nonlinear model, thanks to the kernel trick
- There exists many kernels, which correspond to different feature spaces (of finite or infinite dimensions)
- We can combine and learn kernels, e.g., for integration of heterogeneous data
- Hot research topics
  - Large-scale ML with kernels
  - Deep kernel methods

MURAKOZE