Support Vector Machine

Fabrice Rossi

SAMM
Université Paris 1 Panthéon Sorbonne

2018

Outline

From ERM to RLM

Empirical Risk Minimization

in the binary classification case $\mathcal{Y} = \{-1, 1\}$ with the binary loss function

- is computationally hard
- does not provide asymptotic universal consistency

From ERM to RLM

Empirical Risk Minimization

in the binary classification case $\mathcal{Y} = \{-1, 1\}$ with the binary loss function

- is computationally hard
- does not provide asymptotic universal consistency

Regularized Loss Minimization

- replace the binary loss function by a surrogate convex calibrated loss (function)
- add a (convex) regularization term
- typical example: Support Vector Machine

Outline

Linear Support Vector Machine

Kernelized SVM

Kernels

Support Vector Machine (linear case)

Hypotheses

- $\mathcal{Y} = \{-1, 1\}$
- $\mathcal{X} = \mathbb{R}^P$ (with extensions)
- binary loss $l_b(p, t) = \mathbf{1}_{p \neq t}$

RLM point of view

- class of models $\mathcal{G} = \left\{ \mathbf{x} \mapsto g_{\beta_0, \boldsymbol{\beta}}(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^T \mathbf{x} \right\}$
- surrogate loss: the hinge loss $l_{hinge}(p, t) = \max(0, 1 - pt)$ (convex and calibrated)
- regularization: $\mathbf{C}(g_{\beta_0, \boldsymbol{\beta}}) = \|\boldsymbol{\beta}\|^2$ (convex)

$$\widehat{(\beta_0, \boldsymbol{\beta})} = \arg\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{N} \sum_{i=1}^{N} \max\left(0, 1 - Y_i \left(\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i\right)\right) + \lambda \|\boldsymbol{\beta}\|^2$$

Ridge principle

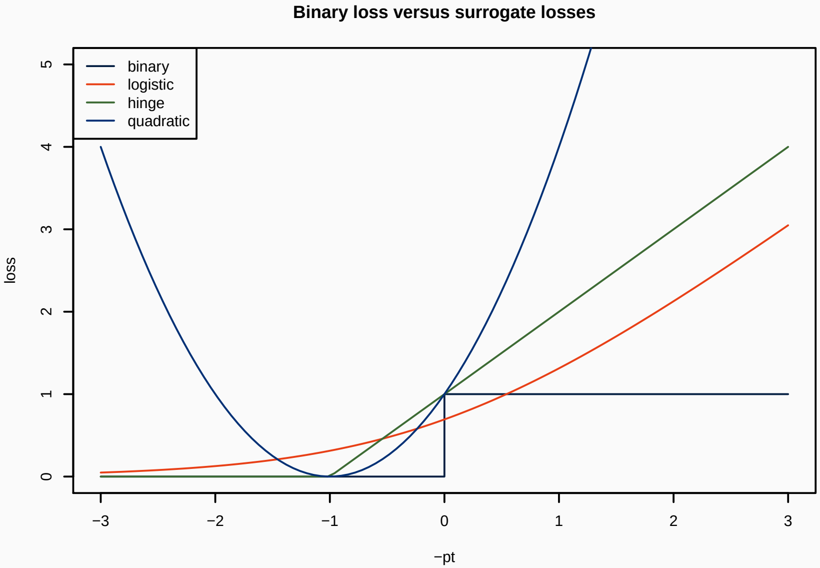
Linear function regularity

- if $f(\mathbf{x}) = \beta^T\mathbf{x}$, then $|f(\mathbf{x}) - f(\mathbf{x}')| \leq \|\beta\|\|\mathbf{x} - \mathbf{x}'\|$ (Cauchy-Schwarz inequality)
- thus $\|\beta\|$ measures the regularity of f

Ridge models

- linear models regularized using their **squared** natural regularity measure
- ridge regression:
  - $\mathcal{Y} = \mathbb{R}$
  - quadratic loss: $l_b(p, t) = (p - t)^2$, no surrogate loss needed
- ridge logistic regression:
  - $\mathcal{Y} = \{-1, 1\}$
  - surrogate logistic loss: $l_{logi}(p, t) = \log(1 + \exp(-pt))$ (convex and calibrated)

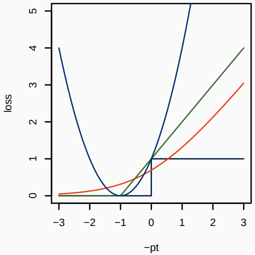
Why the hinge loss?



Why the hinge loss?

The hinge loss

- is convex and calibrated
- is an upper bound of the binary loss
- does not penalize (at all) very good decision ($pt \geq 1$)
- does not overemphasize errors



Robust decisions

- the actual decision model is $f(\mathbf{x}) = \mathrm{sign}(g(\mathbf{x}))$
- if $g(\mathbf{X}_i)Y_i \geq 0$ the decision is correct, but it could be influenced by noise, especially is $g(\mathbf{X}_i)Y_i \simeq 0$
- confidence in the decision increases with $g(\mathbf{X}_i)Y_i$
- the hinge loss penalizes decisions that are not "robust" enough in term of the "margin" $g(\mathbf{X}_i)Y_i$

Geometrical point of view

Definition (Linear separation)

A data set $\mathcal{D}$ is **linearly separable** if there is a linear model given by $g_{\beta_0,\boldsymbol{\beta}}(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^T\mathbf{x}$ such tat $\forall 1 \leq i \leq N \; \mathrm{sign}(g(\mathbf{X}_i)) = Y_i$

Definition (Geometrical margin)

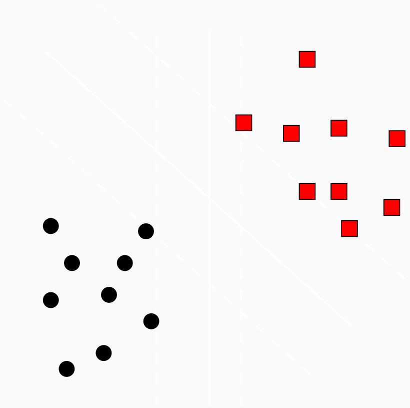
Let $\mathcal{D}$ be a linearly separable data set. The **geometrical margin** of a separating linear model given by $g_{\beta_0,\boldsymbol{\beta}}$ is

$$\mathcal{M}(g_{\beta_0,\boldsymbol{\beta}}) = \min_{1\leq i\leq N} \frac{|\beta_0 + \boldsymbol{\beta}^T\mathbf{X}_i|}{\|\boldsymbol{\beta}\|}.$$

Interpretation

The geometrical margin is the smallest distance between points in the data set and the separating hyperplane associated to the model.

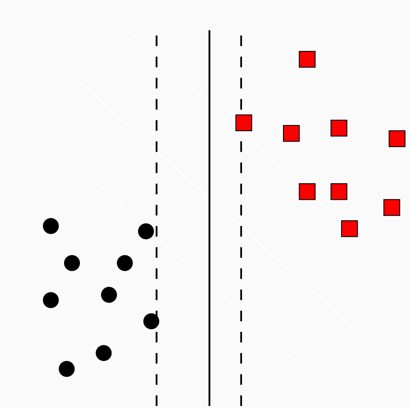
Maximal geometrical margin



Linearly separable data

- many linear classifiers

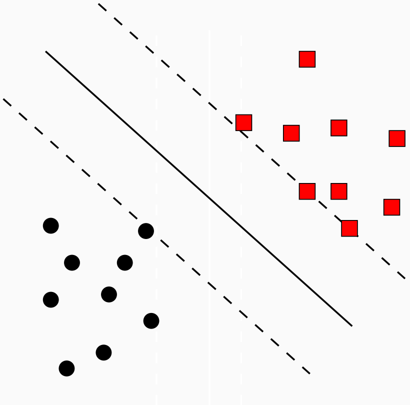
Maximal geometrical margin



Linearly separable data

- many linear classifiers
- if some data are close to the separator, the margin is small $\Rightarrow$ low robustness

Maximal geometrical margin



Linearly separable data

- many linear classifiers
- if some data are close to the separator, the margin is small $\Rightarrow$ low robustness
- choose the classifier by maximizing the geometrical margin

Optimization problem

Geometrical margin maximization

the problem

$$\max_{\beta_0,\boldsymbol{\beta}} \min_{1\leq i\leq N} \frac{|\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i|}{\|\boldsymbol{\beta}\|}$$

is equivalent to

$$(\mathcal{P}_0) \quad \begin{array}{ll} \underset{\beta_0,\boldsymbol{\beta}}{\text{maximize}} & \min_{1\leq i\leq N} \frac{Y_i(\beta_0+\boldsymbol{\beta}^T\mathbf{X}_i)}{\|\boldsymbol{\beta}\|} \\ \text{subject to} & \forall i,\ Y_i(\beta_0 + \boldsymbol{\beta}^T\mathbf{X}_i) > 0 \end{array}$$

Multiple solutions

- if $(\beta_0, \boldsymbol{\beta})$ is a solution of $(\mathcal{P}_0)$ then $(\lambda\beta_0, \lambda\boldsymbol{\beta})$ is also a solution for any $\lambda > 0$
- normalization needed

Optimization problem

Normalization

- let $(\beta_0, \boldsymbol{\beta})$ be a solution of $(\mathcal{P}_0)$ and denote $\lambda = \min_{1 \leq i \leq N} Y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i)$
- as $\lambda > 0$, $(\beta'_0, \boldsymbol{\beta}') = \frac{1}{\lambda}(\beta_0, \boldsymbol{\beta})$ is also a solution and

$$\forall i,\ \ Y_i(\beta'_0 + {\boldsymbol{\beta}'}^T \mathbf{X}_i) = \frac{1}{\lambda} Y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i)$$

  and thus $\min_{1 \leq i \leq N} Y_i(\beta'_0 + {\boldsymbol{\beta}'}^T \mathbf{X}_i) = 1$
- the maximal geometrical margin is then $\frac{1}{\|\boldsymbol{\beta}'\|}$
- in summary, if $(\mathcal{P}_0)$ has a solution, then there is $(\beta'_0, \boldsymbol{\beta}')$ such that
  - $\min_{1 \leq i \leq N} Y_i(\beta'_0 + {\boldsymbol{\beta}'}^T \mathbf{X}_i) = 1$
  - the margin is $\frac{1}{\|\boldsymbol{\beta}'\|}$

Equivalent problem

Normalized problem

- we consider

$$(\mathcal{P}_1) \quad \underset{\beta_0, \boldsymbol{\beta}}{\text{maximize}} \quad \frac{1}{\|\boldsymbol{\beta}\|} \qquad \text{subject to} \quad \forall i,\ Y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i) \geq 1$$

- let $(\beta_0, \boldsymbol{\beta})$ be a solution of $(\mathcal{P}_1)$ and for a $\lambda > 0$ denote $(\beta_0', \boldsymbol{\beta}') = \lambda(\beta_0, \boldsymbol{\beta})$
- then $\frac{1}{\|\boldsymbol{\beta}'\|} = \frac{1}{\lambda\|\boldsymbol{\beta}\|}$ and $\forall i,\ Y_i(\beta_0' + \boldsymbol{\beta}'^T \mathbf{X}_i) \geq \lambda > 0$
- thus $\min_{1 \leq i \leq N} \frac{Y_i(\beta_0' + \boldsymbol{\beta}'^T \mathbf{X}_i)}{\|\boldsymbol{\beta}'\|} \geq \frac{1}{\|\boldsymbol{\beta}\|}$
- therefore $(\mathcal{P}_0)$ has a solution with a maximal margin of at least $\frac{1}{\|\boldsymbol{\beta}\|}$
- with the converse result from last slide, we conclude that $(\mathcal{P}_1)$ and $(\mathcal{P}_0)$ have the same optimal value and have solutions together

Linearly separable SVM

Final problem

$$(\mathcal{P}) \quad \underset{\beta_0, \boldsymbol{\beta}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2$$
$$\text{subject to} \quad \forall i,\ Y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i) \geq 1$$

Convex problem: convex criterion and convex constraints

KKT conditions

- stationarity: $\boldsymbol{\beta}^* = \sum_{i=1}^N \mu_i Y_i \mathbf{X}_i$ and $\sum_{i=1}^N \mu_i Y_i = 0$
- complementary slackness: $\forall i,\ \mu_i \left(Y_i(\beta_0^* + {\boldsymbol{\beta}^*}^T \mathbf{X}_i) - 1 \right) = 0$

Support vectors

- the X_i such that $Y_i(\beta_0^* + {\boldsymbol{\beta}^*}^T \mathbf{X}_i) = 1$
- $\boldsymbol{\beta}^*$ depends only on them!

Non separable case

Soft margin

- if the data is not separable $\forall i,\ Y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i) \geq 1$ can not hold
- relax the constraints: $\forall i,\ Y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i) \geq 1 - \xi_i$ (with $\xi_i \geq 0$)
- penalize the relaxation: $\sum_{i=1}^N \xi_i$ should be minimal

Linear Support Vector Machine

$$
\begin{array}{lll}
(\mathcal{P}) & \underset{\beta_0, \boldsymbol{\beta}, \boldsymbol{\xi}}{\text{minimize}} & \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^N \xi_i \\
& \text{subject to} & \forall i,\ Y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i) \geq 1 - \xi_i \\
& & \forall i,\ \xi_i \geq 0
\end{array}
$$

Link to the hinge loss

Analysis

- let $(\beta_0^*, \boldsymbol{\beta}^*, \boldsymbol{\xi}^*)$ be a solution of $(\mathcal{P})$, then
  - either $\xi_i^* = 0$
  - or $\xi_i^* = 1 - Y_i(\beta_0^* + {\boldsymbol{\beta}^*}^T \mathbf{X}_i)$
- and thus $\xi_i^* = \max(0, 1 - Y_i(\beta_0^* + {\boldsymbol{\beta}^*}^T \mathbf{X}_i))$
- i.e. $\xi_i^* = l_{hinge}(g_{\beta_0^*, \boldsymbol{\beta}^*}(\mathbf{X}_i), Y_i)$
- therefore solving $(\mathcal{P})$ implies to minimize

$$\frac{1}{2}\|\boldsymbol{\beta}\|^2 + CN\widehat{R}_{l_{hinge}}(g_{\beta_0, \boldsymbol{\beta}}, \mathcal{D})$$

- one can show rigorously that $(\mathcal{P})$ is equivalent to RLM

In practice

Dual approach

- $(\mathcal{P})$ is equivalent to following easier to solve dual problem

$$(\mathcal{D}) \quad \begin{array}{ll} \underset{\boldsymbol{\alpha}}{\text{maximize}} & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j Y_i Y_j \mathbf{X}_i^T \mathbf{X}_j \\ \text{subject to} & \sum_{i=1}^N \alpha_i Y_i = 0 \\ & \forall i,\ 0 \leq \alpha_i \leq C \end{array}$$

- for any α_i such that $0 < \alpha_i < C$, $\beta_0 = Y_i - \sum_{j=1}^N \alpha_j Y_j \mathbf{X}_i^T \mathbf{X}_j$
- $\boldsymbol{\beta} = \sum_{j=1}^N \alpha_j Y_j \mathbf{X}_j$
- conditions on α_i
  - sparsity: $Y_i(\beta_0^* + \boldsymbol{\beta}^{*T} \mathbf{X}_i) > 1 \Rightarrow \alpha_i = 0$
  - support vector: $Y_i(\beta_0^* + \boldsymbol{\beta}^{*T} \mathbf{X}_i) < 1 \Rightarrow \alpha_i = C$
  - on the margin: $0 < \alpha_i < C \Rightarrow Y_i(\beta_0^* + \boldsymbol{\beta}^{*T} \mathbf{X}_i) = 1$

Algorithms

Standard (older) solutions

- with quadratic programming for the dual problem
- scales between $\Theta\left(N^2\right)$ and $\Theta\left(N^3\right)$, depending on the number of support vectors (and thus on C)

Modern approaches

For solutions within ϵ of the optimal one

- cutting plan methods $\Theta\left(\frac{NP}{\lambda\epsilon}\right)$
- stochastic sub-gradient methods $\Theta\left(\frac{P}{\lambda\epsilon}\right)$

Meta-parameter

- C or λ must be optimized (with a validation like approach)
- notice that the running time of the algorithm strongly depends on C/λ

Example

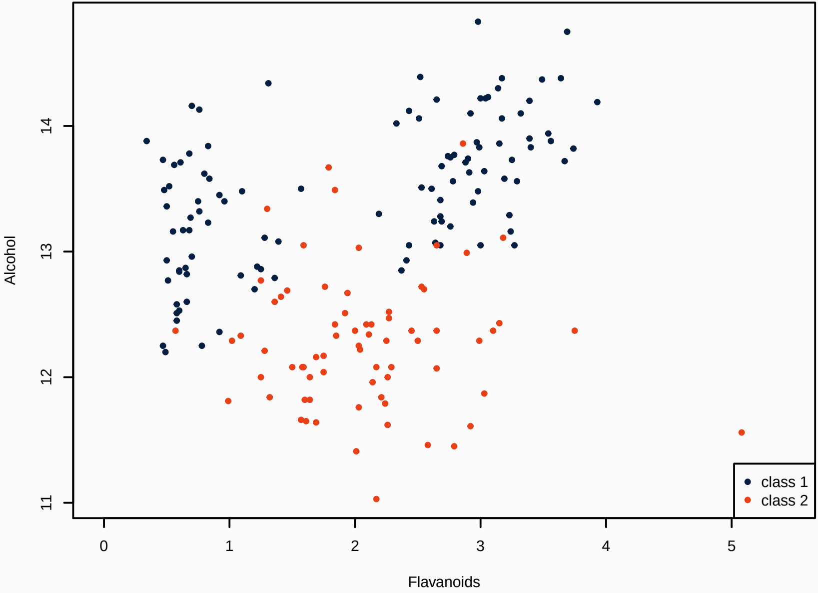
Wine data set

- chemical analysis of wines derived from three cultivars ($\mathcal{Y} = \{1, 2, 3\}$)
- merge cultivar 1 and 3 to get a binary classification problem
- 178 observations with 2 variables ($\mathcal{X} = \mathbb{R}^2$)

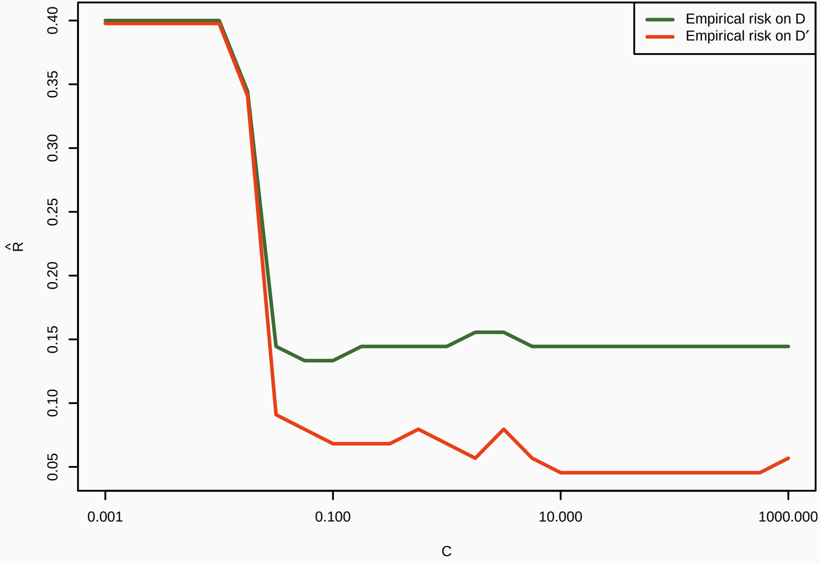
Support vector machine

- use half the data for the learning set $\mathcal{D}$
- use the other half $\mathcal{D}'$ for selecting C
- grid search: C of the form 10^k with $k \in \{-3, -2, \ldots, 3\}$
- notice that the range of C should depend on N

Example

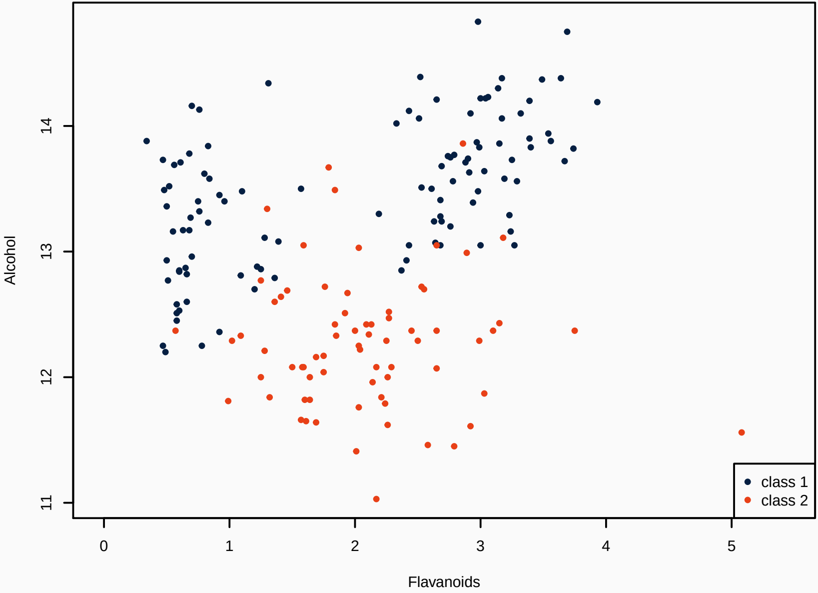


Results

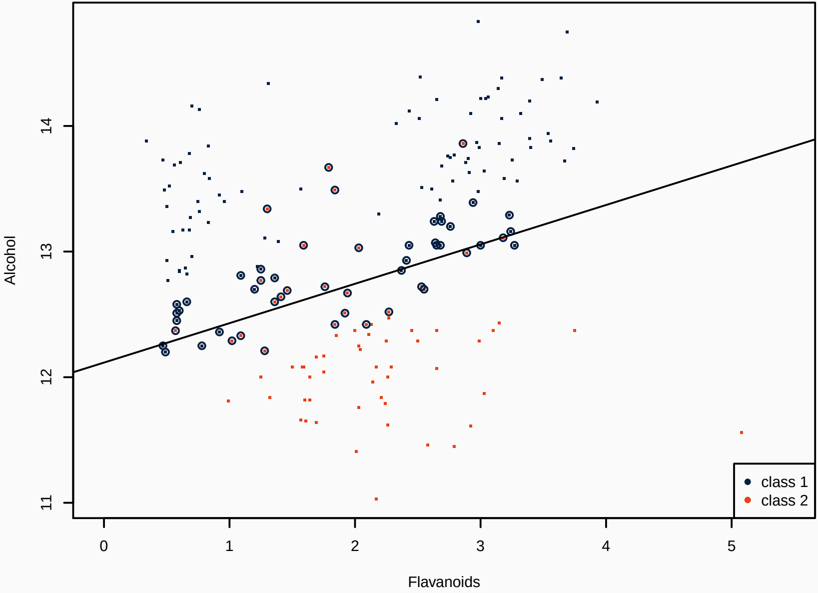


Empirical risk on D
Empirical risk on D′
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
$\hat{R}$
0.001
0.100
10.000
1000.000
C

Results



Results

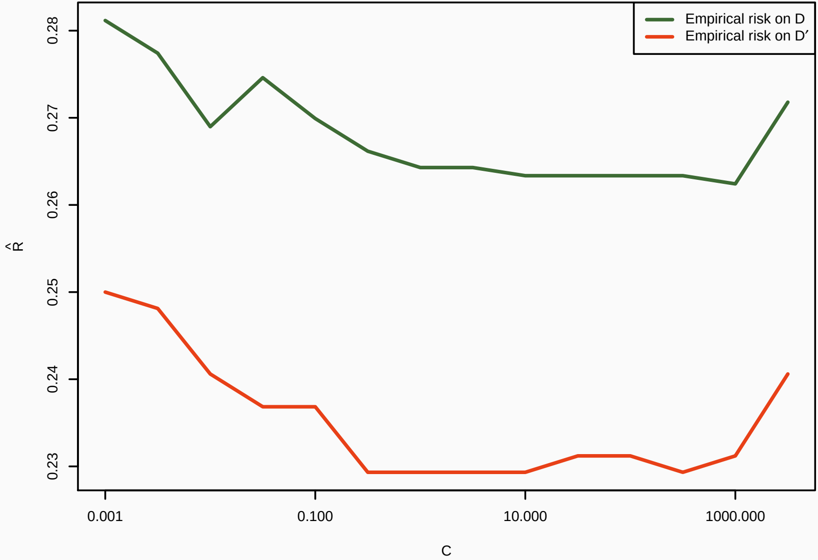


Example

Wine quality data

- quality of a wine graded from 1 to 10:
  - good wines for 6 or more
  - bad wines for 5 or less
- 11 numerical variables giving chemical and physical properties of the wine
- 1067 examples for learning
- 532 examples for selecting C

Results


Empirical risk on D
Empirical risk on D′
$\hat{R}$
0.28
0.27
0.26
0.25
0.24
0.23
0.001
0.100
10.000
1000.000
C

Results

Confusion matrix

- final model on the full data set

	BAD	GOOD
BAD	574	234
GOOD	170	621

- possible over estimated quality but consistent with the validation set performances

Outline

Linear Support Vector Machine

Kernelized SVM

Kernels

Linear separability

Are linearly separable data set frequent?

Results from T. Cover (1965) for points in general position:

- separability depends on the dimension of the input space
- the expectation of the maximum number of linearly separable points in dimension P is $2P$
- the expectation of the minimal number of independent variables needed to separate N points is $\frac{N+1}{2}$
- the transition from separable to non separable is more and more abrupt as the dimension increases

Linear separability

Are linearly separable data set frequent?

Results from T. Cover (1965) for points in general position:

- separability depends on the dimension of the input space
- the expectation of the maximum number of linearly separable points in dimension P is $2P$
- the expectation of the minimal number of independent variables needed to separate N points is $\frac{N+1}{2}$
- the transition from separable to non separable is more and more abrupt as the dimension increases

Consequence

Let's increase the number of variables!

Explicit mapping

Data transformation

- chose a mapping Φ from $\mathbf{X} = \mathbb{R}^P$ to $\mathbb{R}^Q$ with $Q \gg P$ with $\Phi(\mathbf{X}) = (\phi_1(\mathbf{X}), \ldots, \phi_Q(\mathbf{X}))^T$
- replace the data set $\mathcal{D} = ((\mathbf{X}_i, \mathbf{Y}_i))_{1 \leq i \leq N}$ by $((\Phi(\mathbf{X}_i), \mathbf{Y}_i))_{1 \leq i \leq N}$
- classical approach for the generalized linear model with
  - interaction terms $\phi(\mathbf{X}) = X_j \times X_k$
  - polynomial terms $\phi(\mathbf{X}) = X_j^k$
  - etc.

Support Vector Machine

- applies directly with $g_{\beta_0,\boldsymbol{\beta}}(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^T \Phi(\mathbf{x})$
- can be used with $Q = N$ by leveraging the regularization (as in ridge regression)

SVM optimization revisited

Primal version

$$(\mathcal{P}) \quad \underset{\beta_0,\boldsymbol{\beta},\boldsymbol{\xi}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to} \quad \forall i,\ Y_i(\beta_0 + \boldsymbol{\beta}^T\Phi(\mathbf{X}_i)) \geq 1 - \xi_i$$

$$\forall i,\ \xi_i \geq 0$$

The parameter number grows with Q

Dual version

$$(\mathcal{D}) \quad \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j Y_i Y_j \Phi(\mathbf{X}_i)^T\Phi(\mathbf{X}_j)$$

$$\text{subject to} \quad \sum_{i=1}^{N}\alpha_i Y_i = 0$$

$$\forall i,\ 0 \leq \alpha_i \leq C$$

The $\boldsymbol{\alpha}$ parameter dimension does not depend on Q

SVM optimization revisited

Primal version

$$
(\mathcal{P}) \quad \underset{\beta_0, \boldsymbol{\beta}, \boldsymbol{\xi}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C \sum_{i=1}^{N} \xi_i
$$

$$
\text{subject to} \quad \forall i,\ Y_i(\beta_0 + \boldsymbol{\beta}^T \Phi(\mathbf{X}_i)) \geq 1 - \xi_i
$$

$$
\forall i,\ \xi_i \geq 0
$$

The parameter number grows with Q

Dual version

$$
(\mathcal{D}) \quad \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j Y_i Y_j \Phi(\mathbf{X}_i)^T \Phi(\mathbf{X}_j)
$$

$$
\text{subject to} \quad \sum_{i=1}^{N} \alpha_i Y_i = 0
$$

$$
\forall i,\ 0 \leq \alpha_i \leq C
$$

The $\boldsymbol{\alpha}$ parameter dimension does not depend on Q

Dual problem

Representation of the solution

- $\beta_0 = Y_i - \sum_{j=1}^{N} \alpha_j \Phi(\mathbf{X}_i)^T \Phi(\mathbf{X}_j)$ for an α_i such that $0 < \alpha_i < C$
- $\boldsymbol{\beta} = \sum_{j=1}^{N} \alpha_j Y_j \Phi(\mathbf{X}_j)$ and therefore

$$g_{\beta_0,\boldsymbol{\beta}}(\mathbf{x}) = \beta_0 + \sum_{j=1}^{N} \alpha_j Y_j \Phi(\mathbf{X}_j)^T \Phi(\mathbf{x}) = g_{\beta_0,\boldsymbol{\alpha}}(\mathbf{x})$$

Calculating the SVM

- $\boldsymbol{\beta}$ is not needed
- the parameters β_0 and $\boldsymbol{\alpha}$ are of fixed dimension independent of Q
- everything can be computed using only $\Phi(\mathbf{x})^T \Phi(\mathbf{x}')$ for $\mathbf{x} \in \mathcal{X}$ and $\mathbf{x}' \in \mathcal{X}$

Dual problem

Representation of the solution

- $\beta_0 = Y_i - \sum_{j=1}^{N} \alpha_j \Phi(\mathbf{X}_i)^T \Phi(\mathbf{X}_j)$ for an α_i such that $0 < \alpha_i < C$
- $\boldsymbol{\beta} = \sum_{j=1}^{N} \alpha_j Y_j \Phi(\mathbf{X}_j)$ and therefore

$$g_{\beta_0,\boldsymbol{\beta}}(\mathbf{x}) = \beta_0 + \sum_{j=1}^{N} \alpha_j Y_j \Phi(\mathbf{X}_j)^T \Phi(\mathbf{x}) = g_{\beta_0,\boldsymbol{\alpha}}(\mathbf{x})$$

Calculating the SVM

- $\boldsymbol{\beta}$ is not needed
- the parameters β_0 and $\boldsymbol{\alpha}$ are of fixed dimension independent of Q
- everything can be computed using only $\Phi(\mathbf{x})^T \Phi(\mathbf{x}')$ for $\mathbf{x} \in \mathcal{X}$ and $\mathbf{x}' \in \mathcal{X}$

Regularization point of view

Regularization term

- original term $\|\boldsymbol{\beta}\|^2$
- using the formula for $\boldsymbol{\beta}$ we have

$$\|\boldsymbol{\beta}\|^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j Y_i Y_j \Phi(\mathbf{X}_i)^T \Phi(\mathbf{X}_j)$$

Hinge loss

$$\widehat{R}_{l_{hinge}}(g_{\beta_0,\boldsymbol{\alpha}}, \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} \max\left(0, 1 - Y_i\left(\beta_0 + \sum_{j=1}^{N} \alpha_j Y_j \Phi(\mathbf{X}_j)^T \Phi(\mathbf{X}_i)\right)\right)$$

Regularization point of view

Regularization term

- original term $\|\boldsymbol{\beta}\|^2$
- using the formula for $\boldsymbol{\beta}$ we have

$$\|\boldsymbol{\beta}\|^2 = \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j Y_i Y_j \Phi(\mathbf{X}_i)^T\Phi(\mathbf{X}_j)$$

Hinge loss

$$\widehat{R}_{l_{hinge}}(g_{\beta_0,\boldsymbol{\alpha}},\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\max\left(0, 1 - Y_i\left(\beta_0 + \sum_{j=1}^{N}\alpha_j Y_j \Phi(\mathbf{X}_j)^T\Phi(\mathbf{X}_i)\right)\right)$$

Implicit approach

Specifying Φ

- with the dual problem or the regularization approach we do not need Φ but rather $\Phi(\mathbf{x})^T\Phi(\mathbf{x}')$ for any $(\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2$
- the linear case corresponds to $\Phi(\mathbf{x})^T\Phi(\mathbf{x}') = \mathbf{x}^T\mathbf{x}'$
- could we specify directly $k(\mathbf{x}, \mathbf{x}')$?

Definition (Kernel)

A kernel K on a set $\mathcal{X}$ is a function from $\mathcal{X} \times \mathcal{X}$ to $\mathbb{R}$ such that

- k is **symmetric**: $\forall(\mathbf{x}, \mathbf{x}') \in \mathcal{X} \times \mathcal{X}$, $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$
- k is **positive definite**: for all $n \geq 1$, all $(\mathbf{x}_i)_{1\leq i\leq n}$, n elements of $\mathcal{X}$, and all $(\gamma_i)_{1\leq i\leq n}$, n elements of $\mathbb{R}$,

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_i\gamma_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0$$

Kernel based SVM

Dual with kernel

- optimization problem

$$(\mathcal{D}) \quad \begin{array}{ll} \text{maximize}_{\boldsymbol{\alpha}} & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j Y_i Y_j k(\mathbf{X}_i, \mathbf{X}_j) \\ \text{subject to} & \sum_{i=1}^{N} \alpha_i Y_i = 0 \\ & \forall i,\ 0 \leq \alpha_i \leq C \end{array}$$

- $\beta_0 = Y_i - \sum_{j=1}^{N} \alpha_j k(\mathbf{X}_i, \mathbf{X}_j)$
- model

$$g_{\beta_0, \boldsymbol{\alpha}}(\mathbf{x}) = \beta_0 + \sum_{j=1}^{N} \alpha_j Y_j k(\mathbf{X}_j, \mathbf{x})$$

- similar construction for the regularized point of view

Issues

Important questions

- Does this construction make sense?
- Why specifying a kernel k rather than a transformation Φ?

Outline

Linear Support Vector Machine

Kernelized SVM

Kernels

Kernels are efficient

Polynomial kernel

- for $\mathcal{X} = \mathbb{R}^P$, $k_d(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}' + 1)^d$ is a kernel
- computation time $\Theta(P + d)$
- one can show that there is Φ_d from $\mathcal{X}$ to $\mathbb{R}^Q$ such that $k_d(\mathbf{x}, \mathbf{x}') = \Phi_d(\mathbf{x})^T\Phi_d(\mathbf{x}')$
- but $Q = \binom{P+d}{P} = \frac{(P+d)!}{P!d!}$ and computing Φ costs a lot, e.g. $\Theta\left(P^d\right)$ when d is small and P is large

Illustration

- for $d = 2$ and $P = 2$, $\Phi_2(\mathbf{x}) = \left(1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2\right)^T$
- $\Phi_2(\mathbf{x})$ computation involves 6 operations and $\Phi_2(\mathbf{x})^T\Phi_2(\mathbf{x}')$ uses 11 operations
- $k_2(\mathbf{x}, \mathbf{x}')$ needs 5 operations

Kernels are flexible

Numerical examples

For $\mathcal{X} = \mathbb{R}^P$

- Linear kernel: $k_{lin}(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$
- Polynomial kernel: $k_d(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}' + 1)^d$
- Gaussian kernel: $k_{gauss,\sigma}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$
- Laplace kernel: $k_{lap,\gamma}(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma\|\mathbf{x} - \mathbf{x}'\|\right)$

Dissimilarity spaces

When $\mathcal{X}$ has a dissimilarity d, one can use the Gaussian kernel as follows: $k_{gauss,\sigma}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{d(\mathbf{x},\mathbf{x}')}{2\sigma^2}\right)$

Kernels are flexible

String kernels

For $\mathcal{X}$ the set of words on an alphabet Σ (i.e., $\mathcal{X} = \Sigma^*$)

- generic matching kernel
  $k(\mathbf{x}, \mathbf{x}') = \sum_{s \in \sigma^*} \text{num}(\mathbf{x}, s)\text{num}(\mathbf{x}', s)w(s)$ where $\text{num}(\mathbf{x}, s)$ denotes the number of times a string s occurs in $\mathbf{x}$ and $w(s)$ is a weighting function on Σ^*
- examples:
  - common characters: $w(s) = 0$ is $|s| > 1$ (where $|s|$ is the length of the string s)
  - common sub-strings or words with similar tricks
- numerous extensions: position dependent weighting, approximate matches, etc.

Kernels are flexible

Many other input types

- graph nodes
- whole graphs
- texts
- images
- etc.

Kernel theory

Definition (Reproducing Kernel Hilbert Space (RKHS))

Let $\mathcal{X}$ be an arbitrary space and let H be a Hilbert space of functions from $\mathcal{X}$ to $\mathbb{R}$ (with the inner product $\langle ., . \rangle_H$). H is a Reproducing Kernel Hilbert Space if there is a function K from $\mathcal{X} \times \mathcal{X}$ to $\mathbb{R}$ such that:

1. for all $\mathbf{x} \in \mathcal{X}$, $K(\mathbf{x}, .) \in H$
2. for all $f \in H$ and for all $\mathbf{x} \in \mathcal{X}$, $f(x) = \langle f, K(\mathbf{x}, .) \rangle_H$

K is the reproducing kernel of H.

Theorem (Moore–Aronszajn)

Let k be a symmetric positive definite kernel on $\mathcal{X}$. Then there is a unique RKHS of functions from $\mathcal{X}$ to $\mathbb{R}$ for which k is the reproducing kernel.

From theory to practice

Kernel trick

Assume given a symmetric positive definite kernel k on $\mathcal{X}$.

- then there is a **feature space** H and a **feature map** Φ from $\mathcal{X}$ to H defined by $\Phi(\mathbf{x}) = k(\mathbf{x}, .)$
- H is a vector space with an inner product: any machine learning algorithm that uses only the Euclidean structure of $\mathbb{R}^P$ can be implemented in H
- for instance the linear SVM:
    - replace $\boldsymbol{\beta}^T\mathbf{x}$ by $\langle\boldsymbol{\beta}, \Phi(\mathbf{x})\rangle_H$
    - and $\|\boldsymbol{\beta}\|^2$ by $\|\boldsymbol{\beta}\|_H^2$

Representer theorem

To ease the use of the kernel trick, we need a theorem.

Theorem (Representer theorem)

Let k be a symmetric positive definite kernel on $\mathcal{X}$ with H the associated RKHS. Assume that C is a strictly increasing function from $\mathbb{R}^+$ to $\mathbb{R}$ and R is a function from $\mathbb{R}^N$ to $\mathbb{R}$. Any solution to

$$\arg\min_{f \in H} R(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)) + C(\|f\|_H)$$

for some fixed elements of $\mathcal{X}$, $\mathbf{x}_1, \ldots, \mathbf{x}_N$, can be written $f = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, .)$

Regularized version

- optimization problem with $\boldsymbol{\beta} \in H$

$$\min_{\beta_0,\boldsymbol{\beta}} \frac{1}{N} \sum_{i=1}^{N} l_{hinge}\left(\beta_0 + \langle \boldsymbol{\beta}, \Phi(\mathbf{X}_i)\rangle_H, Y_i\right) + \lambda \|\boldsymbol{\beta}\|_H^2$$

  with $\Phi(\mathbf{X}_i) = k(\mathbf{X}_i, .)$
- according to the representer theorem, $\boldsymbol{\beta}$ has the form $\sum_{j=1}^{N} \alpha_j k(\mathbf{X}_j, .)$ and therefore
  - $\langle \boldsymbol{\beta}, \Phi(\mathbf{X}_i)\rangle_H = \sum_{j=1}^{N} \alpha_j \langle k(\mathbf{X}_j, .), k(\mathbf{X}_i, .)\rangle_H = \sum_{j=1}^{N} \alpha_j k(\mathbf{X}_i, \mathbf{X}_j)$
  - $\|\boldsymbol{\beta}\|_H^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j k(\mathbf{X}_i, \mathbf{X}_j)$
- new optimization problem

$$\min_{\beta_0,\boldsymbol{\alpha}} \frac{1}{N} \sum_{i=1}^{N} l_{hinge}\left(\beta_0 + \sum_{j=1}^{N} \alpha_j k(\mathbf{X}_i, \mathbf{X}_j), Y_i\right) + \lambda \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j k(\mathbf{X}_i, \mathbf{X}_j)$$

Kernel ridge regression

Ridge regression

- regularized linear regression
- optimization problem in an arbitrary Hilbert space H

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{N} \sum_{i=1}^{N} (\beta_0 + \langle \boldsymbol{\beta}, \Phi(\mathbf{X}_i) \rangle_H - Y_i)^2 + \lambda \|\boldsymbol{\beta}\|_H$$

Kernel ridge regression

- representer theorem again!
- transformed optimization problem

$$\min_{\beta_0, \boldsymbol{\alpha}} \frac{1}{N} \sum_{i=1}^{N} \left(\beta_0 + \sum_{j=1}^{N} \alpha_j k(\mathbf{X}_i, \mathbf{X}_j) - Y_i \right)^2 + \lambda \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j k(\mathbf{X}_i, \mathbf{X}_j)$$

Kernels provide consistent learning

Dimension

- many kernels increase a lot the dimension of the data representation
- some kernels lead to a RHKS of infinite dimension (e.g., the Gaussian kernel)
- according to Cover's theorem, this should turn any data set into a linearly separable one
- but interesting RHKS have infinite VC-dimension!

Effect of regularization

- in the convex case $g^* = \arg\min_{\{g \in H | \|g\|_H \leq \mu\}} \widehat{A}(g, \mathcal{D})$
- even if H has an infinite VC-dimension, $\{g \in H | \|g\|_H \leq \mu\}$ has a controlled capacity
- consistency can be proved in some cases (e.g. SVM)

General framework

- chose a kernel k
- chose a loss function l
- chose a penalty function $\mathbf{C}$ (strictly increasing function)
- solve

$$\min_{\beta_0,\boldsymbol{\alpha}} \frac{1}{N}\sum_{i=1}^{N} l\left(\beta_0 + \sum_{j=1}^{N}\alpha_j k(\mathbf{X}_i, \mathbf{X}_j), Y_i\right) + \lambda \mathbf{C}\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j k(\mathbf{X}_i, \mathbf{X}_j)\right)$$

- use a model derived from (or equal to) $\mathbf{X} \mapsto \beta_0 + \sum_{j=1}^{N} \alpha_j k(\mathbf{X}_i, \mathbf{X})$
- very general consistent results are available
- very good performances in practice

In practice

Practical aspects

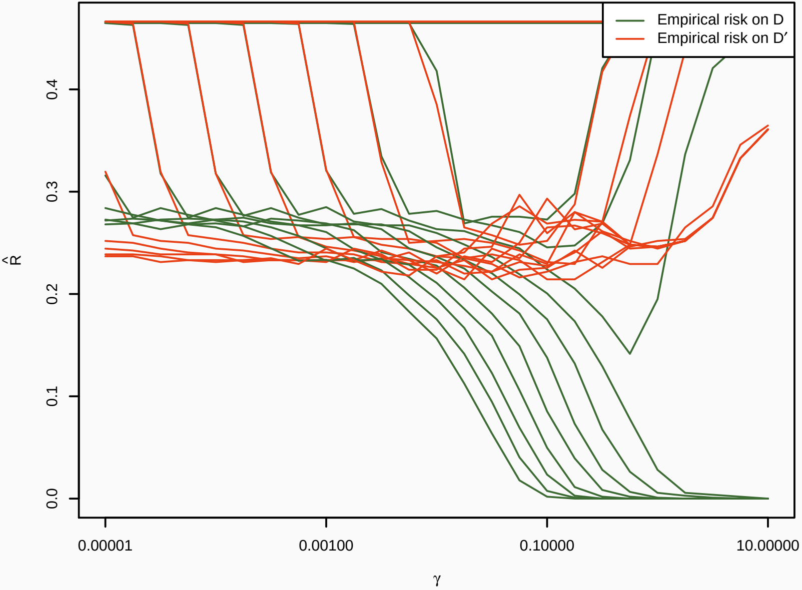
- loss functions?
  - quadratic in regression
  - hinge loss in classification
  - numerous other possibilities
- kernel?
  - must be chosen by a validation like approach
  - meta parameter(s) must be tuned
  - the Gaussian kernel is the de facto standard choice
- the regularization trade-off must be tuned by a validation like approach

Example

Wine quality data

- Same setting as before
  - quality of a wine graded from 1 to 10:
    - good wines for 6 or more
    - bad wines for 5 or less
  - 11 numerical variables giving chemical and physical properties of the wine
  - 1067 examples for learning
  - 532 examples for selecting C
- SVM with the hinge loss and a Gaussian kernel
- in R with e1071, the Gaussian kernel is $k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$

Results



Empirical risk on D
Empirical risk on D′
0.0
0.1
0.2
0.3
0.4
$\hat{R}$
0.00001
0.00100
0.10000
10.00000
γ

Results

Confusion matrix

- final model on the full data set

	BAD	GOOD
BAD	574	234
GOOD	170	621

Linear kernel

	BAD	GOOD
BAD	584	184
GOOD	160	671

Gaussian kernel

- possible over estimated quality but consistent with the validation set performances

Remarks

- strong overfitting when C is large and γ also
- somewhat similar effects of C and γ: small values favor regular models, large values favor overfitting

Outline

Summary

Kernel methods

- a general framework for convex regularized loss minimization
- pros and cons
  - \+ very flexible framework (regression, classification, but also semi-supervised learning, novelty detection, etc)
  - \+ can be applied to exotic data with adapted kernels
  - \+ state of the art performances
  - \+ strong theoretical results
  - \- relatively slow especially for nonlinear kernels
  - \- many meta-parameters
  - \- somewhat complex implementations

Licence

Version

Last modification: 2018-02-05
By: Fabrice Rossi (Fabrice.Rossi@apiacoa.org)
Git hash: 6cf2ee5ae0a4a099004a76f362158e0bd5917749