Support Vector Machine for Image Classification

Bruno Galerne
bruno.galerne@univ-orleans.fr

Université d'Orléans

Vendredi 27/03/2020 = **Confinement COVID-19 J11**
Statistiques pour le traitement d'images
Master 1 Statistique & Data Science, Ingénierie Mathématique

Course Plan

Supervised classification

Kernel SVM for linearly separable training set

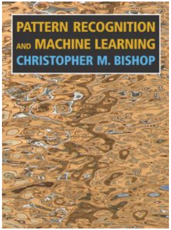
Kernel SVM for non-linearly separable training set

Multi-Class SVM

Practical session

Main references:

- C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, 2006
  Freely available:
  `https://www.microsoft.com/en-us/research/people/cmbishop/prml-book/`
- Various tutorials of [Scikit-learn]



Outline

Supervised classification

Kernel SVM for linearly separable training set

Kernel SVM for non-linearly separable training set

Multi-Class SVM

Practical session

Supervised classification

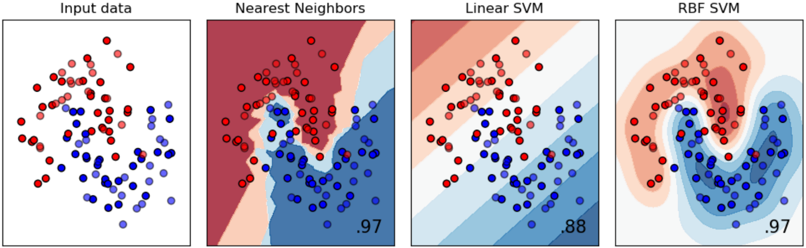
- The points $x \in \mathbb{R}^d$ are partitioned into K classes.
- We have a **training set** of N labeled points:

$$\mathcal{T} = \{(x_n, t_n)_{1 \leqslant n \leqslant N},\ x_n \in \mathbb{R}^d,\ t_n \in \{1, \dots, K\}\}.$$

- The goal is to use this training set to define a classification function

$$\Psi : \mathbb{R}^d \to \{1, \dots, K\}.$$

- The performance of the classifier is measured using a **test set** that is different from the training set.



Training points are solid, testing points are semi-transparent. [1]

[1] Image from: `https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html`

Outline

Feature map and kernel

- The initial data $x_n \in \mathbb{R}^d$ is often not well-described in its original form.
- We transform it using a feature map $\phi : \mathbb{R}^d \to \mathbb{R}^D$.
- This feature map is associated to a kernel function:

$$k(x, x') = \langle \phi(x), \phi(x') \rangle.$$

- The kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to [0, +\infty)$ is a symmetric and positive function.
- As will be shown, for SVM the mapping ϕ need not to be explicit since one only needs to compute the kernel.

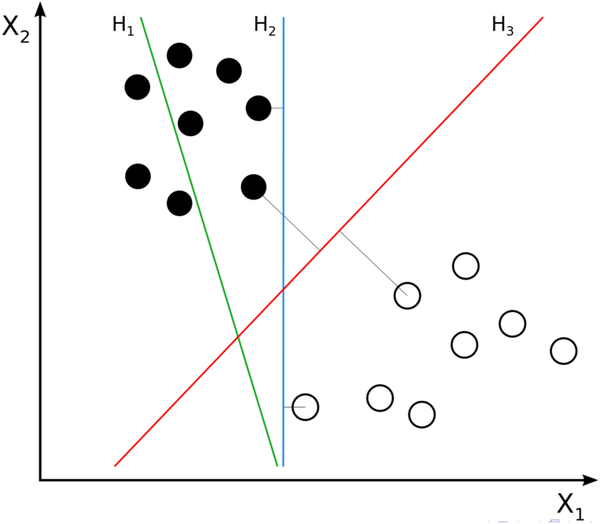
Examples:

- $\phi = \mathrm{id}$, $k(x, x') = \langle x, x' \rangle$.
- Gaussian RBF kernel (RBF = Radial Basis Function).

$$k(x, x') = e^{-\gamma \|x - x'\|}.$$

  Remark: The mapping ϕ such that $k(x_m, x_n) = \langle \phi(x_m), \phi(x_n) \rangle$ is often from $\mathbb{R}^d$ to an infinite dimensional Hilbert space $\mathcal{H}$ (called a reproducing kernel Hilbert space (RKHS)).
- Nice YouTube video: `https://youtu.be/9NrALgHFwTo`

Separating hyperplane



- The main idea of SVM is to find a separating hyperlane that has the largest margin from the dataset.
- Which hyperplane separate the data ? Which one is better?

SVM theory

SVM theory relies on Lagrange duality of constrained convex problems. Below are slides from: Stephen Boyd and Lieven Vandenberghe, *Convex Optimization* `http://web.stanford.edu/~boyd/cvxbook/`

Lagrangian

standard form problem (not necessarily convex)

$$
\begin{array}{ll}
\text{minimize} & f_0(x) \\
\text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\
 & h_i(x) = 0, \quad i = 1, \ldots, p
\end{array}
$$

variable $x \in \mathbf{R}^n$, domain $\mathcal{D}$, optimal value $p^\star$

Lagrangian: $L : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$, with $\mathbf{dom}\, L = \mathcal{D} \times \mathbf{R}^m \times \mathbf{R}^p$,

$$
L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)
$$

- weighted sum of objective and constraint functions
- λ_i is Lagrange multiplier associated with $f_i(x) \leq 0$
- ν_i is Lagrange multiplier associated with $h_i(x) = 0$

Lagrange dual function

Lagrange dual function: $g : \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$,

$$
\begin{aligned}
g(\lambda, \nu) &= \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \\
&= \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right)
\end{aligned}
$$

g is concave, can be $-\infty$ for some λ, ν

lower bound property: if $\lambda \succeq 0$, then $g(\lambda, \nu) \leq p^\star$

proof: if $\tilde{x}$ is feasible and $\lambda \succeq 0$, then

$$
f_0(\tilde{x}) \geq L(\tilde{x}, \lambda, \nu) \geq \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = g(\lambda, \nu)
$$

minimizing over all feasible $\tilde{x}$ gives $p^\star \geq g(\lambda, \nu)$

The dual problem

Lagrange dual problem

$$\begin{array}{ll}\text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda \succeq 0\end{array}$$

- finds best lower bound on $p^\star$, obtained from Lagrange dual function
- a convex optimization problem; optimal value denoted $d^\star$
- λ, ν are dual feasible if $\lambda \succeq 0$, $(\lambda, \nu) \in \mathbf{dom}\, g$
- often simplified by making implicit constraint $(\lambda, \nu) \in \mathbf{dom}\, g$ explicit

example: standard form LP and its dual (page 5–5)

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \succeq 0\end{array} \qquad\qquad \begin{array}{ll}\text{maximize} & -b^T \nu \\ \text{subject to} & A^T \nu + c \succeq 0\end{array}$$

Weak and strong duality

weak duality: $d^\star \le p^\star$

- always holds (for convex and nonconvex problems)
- can be used to find nontrivial lower bounds for difficult problems

  for example, solving the SDP

$$\begin{array}{ll} \text{maximize} & -\mathbf{1}^T\nu \\ \text{subject to} & W + \mathbf{diag}(\nu) \succeq 0 \end{array}$$

  gives a lower bound for the two-way partitioning problem on page 5–7

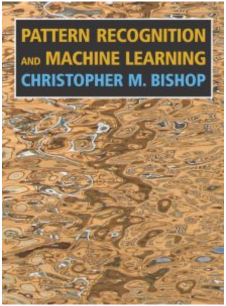
strong duality: $d^\star = p^\star$

- does not hold in general
- (usually) holds for convex problems
- conditions that guarantee strong duality in convex problems are called **constraint qualifications**

SVM theory

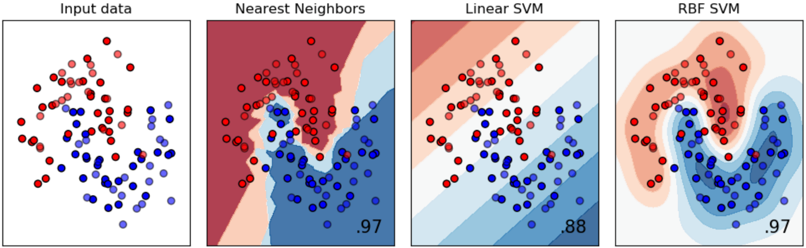
Please read **pages 326 to 331** of

- C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, 2006
  Freely available:
  `https://www.microsoft.com/en-us/research/people/cmbishop/prml-book/`



Outline

Kernel SVM for non-linearly separable training set

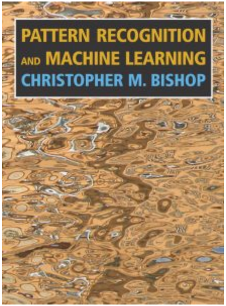


- Datasets are generally not linearly separable.
- A single outlier can make the hypothesis false.
- To overcome this problem, the optimization problem must be relaxed.

SVM theory

Please read Section "7.1.1 Overlapping class distributions" **pages 332 to 336** of

- C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, 2006
  Freely available:
  `https://www.microsoft.com/en-us/research/people/cmbishop/prml-book/`



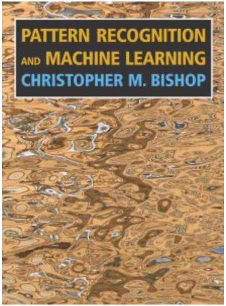
Outline

Multi-Class SVM

- SVM is designed to separate two classes.
- Next we see how to use it with $K > 2$ classes, although it is an open issue.

Please read Section "7.1.3 Multiclass SVMs" **pages 338 to 339** of

- C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, 2006
  Freely available:
  `https://www.microsoft.com/en-us/research/people/cmbishop/prml-book/`



Outline

The practical session is here:

```
https://github.com/bgalerne/M1MAS_Stat_Images/blob/
master/TP_SVM_images.ipynb
```

- C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, 2006
- Stephen BOYD and Lieven VANDENBERGHE, *Convex Optimization*, Cambridge University Press, 2004
- *Pedregosa et al.*, Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011