Support Vector Machine e Random Forest

Contents

- Support Vector Machine
- Multiple classifiers
- Random Forest

Support Vector Machine (SVM)

The theory introduced by Vapnik (statistical learning theory) in 1965
Recently perfected by Vapnik himself and others in 1995

Vapnik suggests determining decision surfaces between classes to separate them (classification boundaries)

SVM was born as a binary classifier (2 classes), with different degrees of complexity:

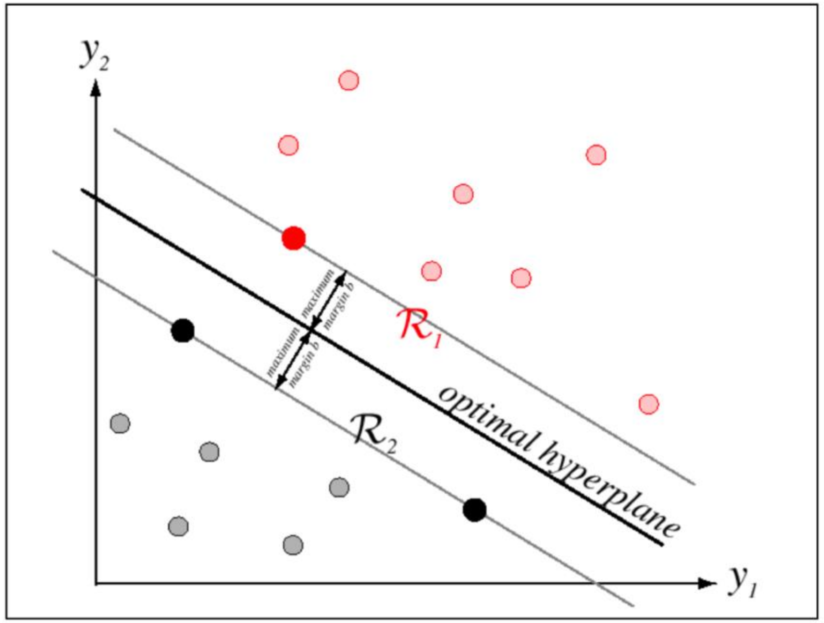
-Linear SVM (the separation surface is a hyperplane) and the patterns of the training set are linearly separable (there is hypothetically at least one hyperplane capable of separating them)

-Linear SVM and non-linearly separable patterns. Classification errors are allowed as there is no hyperplane that can perfectly separate the patterns

-Nonlinear SVM (complex separation surface) with no assumptions about pattern separability

Linear SVM separable pattern

Given two classes of linearly separable multidimensional patterns, of all possible separation hyperplanes, SVM determines the optimal one that can separate the classes by the greatest possible margin. The margin is the minimum distance of points of the two classes in the training set from the discovered hyperplane



If the patterns of the training set are classified by a wide margin, the model is better generalizable, i.e. it is expected that even patterns of the test set close to the boundary between classes are expected to be handled correctly by the model

Linear SVM separable pattern

Given two linearly separable pattern classes, and a training set containing n samples

$x_1, y_1 \ldots x_n, y_n$

$x_i \in \Re^d$ pattern d-dimensional (d=features)

$y_i \in +1, -1$ classes (abels)

There are several hyperplanes that can perform the desired separation

Hyperplane: properties

$D\ (\mathrm{x}) = \mathbf{w}^{\mathrm{T}} \cdot \mathbf{x} + b$

w: vector normal to the hyperplane

$b/\ ||\mathbf{w}||$: distance from the origin

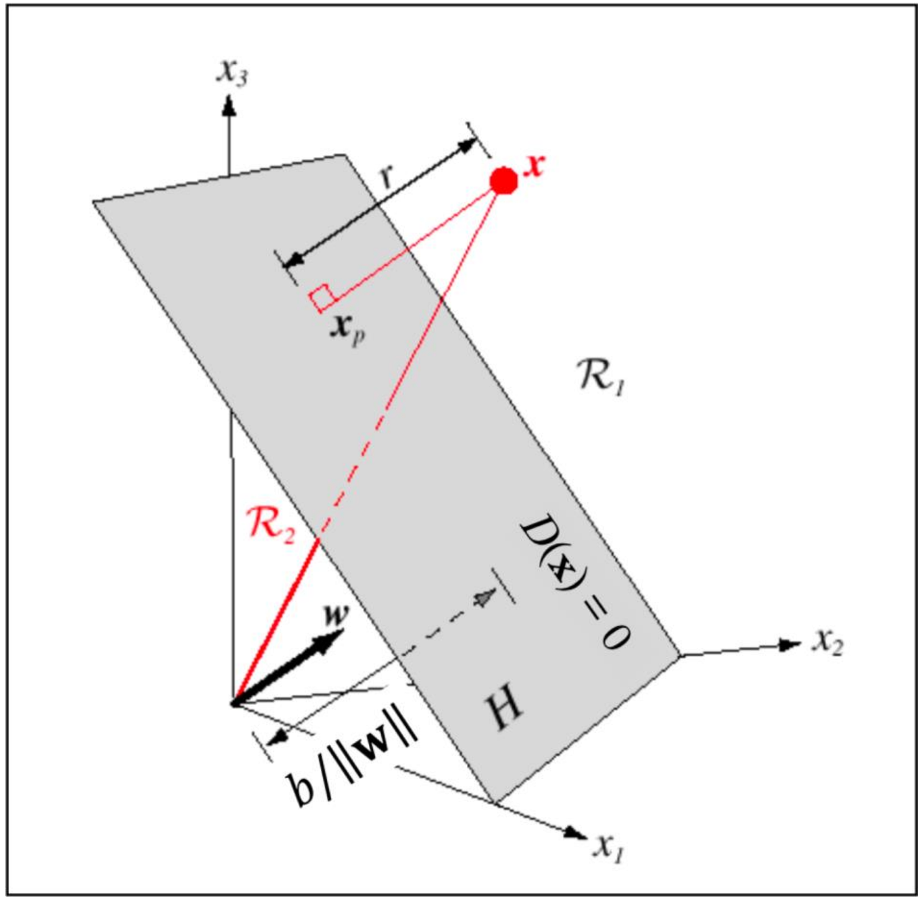
$D\ (\mathbf{x}) = 0$: vectors on the hyperplane

Considered the pattern **x**

$\mathbf{x} = \mathbf{x}_p + \mathbf{r}\ \mathbf{w}/\ ||\mathbf{w}||$

$D\ (\mathbf{x}_{\mathrm{p}}) = 0$

$D\ (\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \cdot \mathbf{x} + b = \mathrm{r}\ ||\mathbf{w}||$



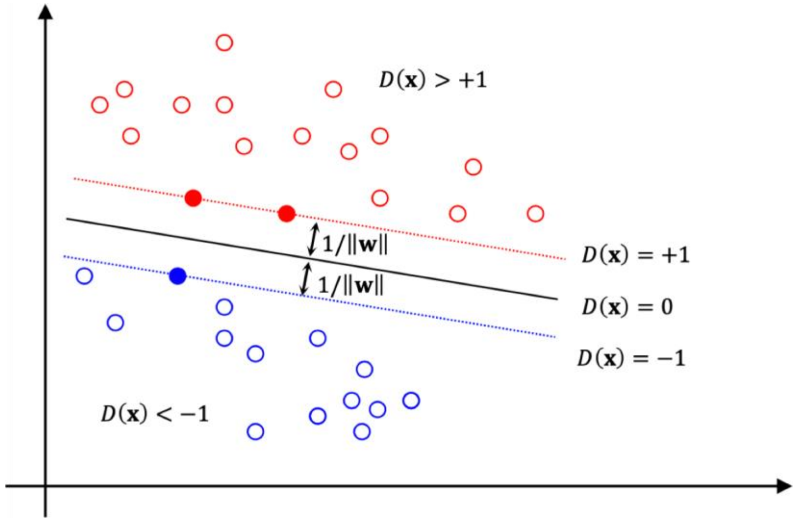
Distance of a vector x from the hyperplane r = $D\ (\mathrm{x})\ /\ ||\mathrm{w}||$

Margin

Hyperplanes(w,b) separating the patterns of the training set with minimum distance 1/ $||\mathrm{w}||$ on each side they satisfy the equations:

$\mathrm{w}\cdot\mathrm{x}_{(}+b\geq+1$ *se* $y_1=+1$

$\mathrm{w}\cdot\mathrm{x}_{(}+b\leq-1$ *se* $y_1=-1$



The distance of the points lying on the hyperplane$D\ (\mathrm{x}) = +1$ from the separation hyperplane$D\ (\mathrm{x}) = 0$ è $1/\ ||\mathrm{w}||$; the same is true for the points of the hyperplane $D\ (\mathrm{x}) = -1$

Il margin is $\tau = 2/\ ||\mathrm{w}||$

The optimal hyperplane according to SVM is the one that maximizes the margin τ (or alternatively minimizes its inverse) and satisfies pattern separation constraints

Minimizes: $||\mathrm{w}||^2\ /\ 2$ → *Objective function of SVM*

Constraints: $y_\mathrm{i}\ [\mathrm{w}\cdot\mathrm{x}_\mathrm{i}+b]-1\geq 0$ for $1=1\ldots n$

Troubleshooting the optimization issue of the object function

Minimizes: $|| \mathbf{w} ||^2 / 2$ → *Objective function of SVM*

Constraints: $y_i \, [\mathbf{w} \cdot x_i + b] - 1 \geq 0$ *per* $i=1 \ldots n$

$\mathbf{w}^* \quad b^*$

Può essere risolto passando a una formulazione Lagrangiana e successivamente a una formulazione duale
La formulazione Lagrangiana prevede di introdurre un moltiplicatore ($\alpha_i \geq 0$) per ogni vincolo nella forma $equazione \geq 0$ e di sottrarre il vincolo moltiplicato per α_i dalla funzione obiettivo:

$$|| \mathbf{w} ||^2 / 2 - \sum_{i=1}^{n} \alpha_i \left(y_i [\mathbf{w} \cdot \mathbf{x}_i + b] - 1\right) = Q(w, b, \alpha)$$

→ *Nuova funzione obiettivo di SVM*

da minimizzare rispetto a $\mathbf{w}$ e b e massimizzare rispetto a $\alpha_i \geq 0$.

$$|| \mathbf{w} ||^2 / 2 - \sum_{i=1}^{n} \alpha_i \left(y_i[\mathbf{w} \cdot \mathbf{x_i} + b] - 1\right) = \mathrm{Q}\,(\mathrm{w}, \mathrm{b}, \alpha)$$ → *New objective fcuntion of SVM*

to be minimised with respect to $\mathbf{w}$ e b and maximize compared to $\alpha(\geq 0$

The solution is to derive the optimal values$\alpha 1$, $\alpha 2$ ... αn

The Karush-Kuhn-Tucker conditions(KKT) ensure that$\alpha(^* = 0$ for all carriers that are not support vector

$$\mathbf{w}^* = \sum_{i=1...n} \alpha_i{}^* y_i\, \mathbf{x}_i$$

e $$b^* = y_s - \sum_{i=1...n} \alpha_i{}^* y_i(\mathbf{x}_i \cdot \mathbf{x}_s)$$ where (x_{S}, y_{S}) is one of the support vector

The optimal hyperplane is therefore:

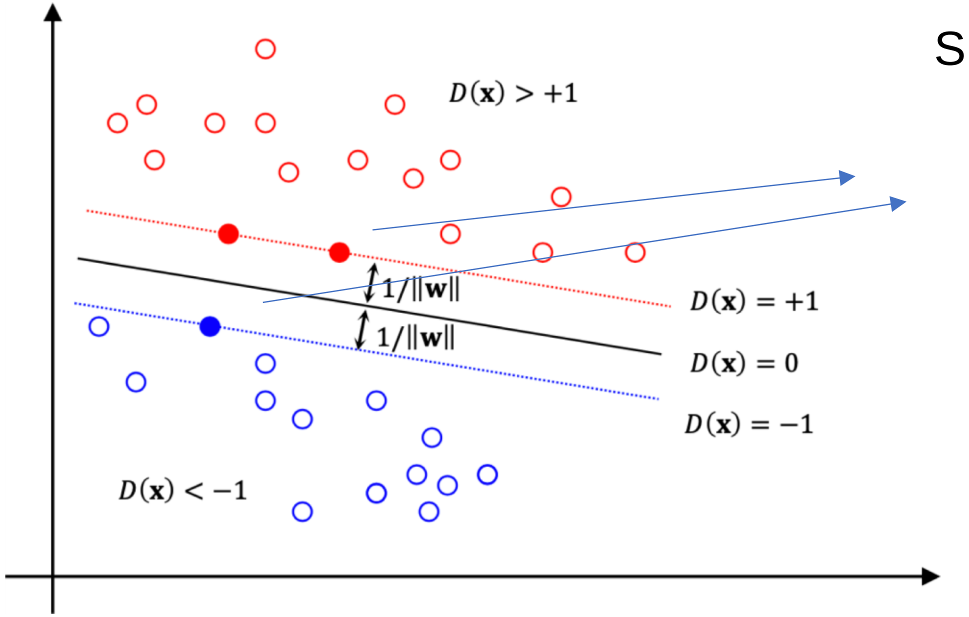
$$D(\mathbf{x}) = \mathbf{w}^* \cdot \mathbf{x} + b^* = \sum_{i=1...n} \alpha_i{}^* y_i(\mathbf{x} \cdot \mathbf{x}_i) + b^*$$

Please note:
The sign of Function$D$ (x) classifies a generic pattern(x)
Summations are reducible to support vectors only

Support Vectors

The patterns in the training set that lie on the edge are the support vectors.

These patterns, which are the most complex cases (precisely those close to the opposite class), completely define the solution of the problem.
The best separation hyperplane can be expressed as a function of only such patterns, regardless of the dimensionality of the space$d$ and the number$n$ of Training Set Patterns



Support vectors

SVM Advantages/Disadvantages

Definition of the solution to the problem of optimizing the objective function based on a small number of support vectors

SVM scales very well with respect to the dimensionality d of the feature space

The computational complexity in the training is quadratic with respect to the number of n patterns in the training set

SVMs with non-separable patterns

In some cases (most real cases) not all patterns can be separated by a hyperplane
It is necessary to relax the separation constraints so that some patterns (as few as possible) can cross the class boundary

Constraints for Separable Patterns: $y_{\mathrm{i}}\ [+\cdot\mathrm{x}_{\mathrm{i}} + b] - 1 \geq 0\ per\ \mathrm{i}{=}1\ldots n$

N positive slack variables are introduced$\xi\mathrm{i}\,, \mathrm{i} = 1 \ldots n$ and change the separation constraints of separable patterns

$$y_{\mathrm{i}}\,[+\cdot\mathrm{x}_{.} + b] - 1 + \xi\mathrm{i} \geq 0 \qquad per\ \mathrm{i} = 1 \ldots n$$

For every patternx_{i} of the training set, the variable$\xi\mathrm{i}$ represents the deviation from the margin

*Multiple Classifiers**

- An approach where several classifiers are used together (in parallel, cascade, or hierarchically) to perform pattern classification

- The predictions of the individual classifiers are merged at a certain level of the classification process

It has been shown (theoretically but mostly in practice) that the use of combinations of classifiers can improve performance even by a lot instead of investing time in the maniacal optimization of a single classifier

*multi-classifier, combination of classifiers, classifier fusion, ensemble learning

Multiple Classifiers

The combination is only effective when the individual classifiers are (at least partially) independent of each other, i.e. they do not make the same types of errors

Independence is normally achieved:

- Using different features (unrelated or uncorrelated) for the same classifier
- Using different feature extraction techniques combined for the same classifier
- Using different classification algorithms on the same features
- Training the same classification algorithm on different portions of the training set (bagging-random forest)
- Training the same classification algorithm Insisting on misclassified patterns (boosting-adaboost)
- The combination can be done at the decision level or at the classification confidence level