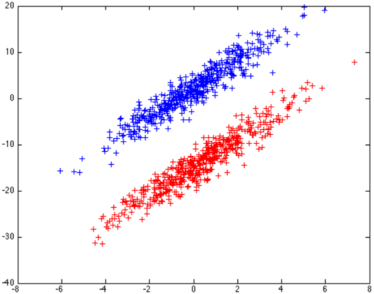
Classification

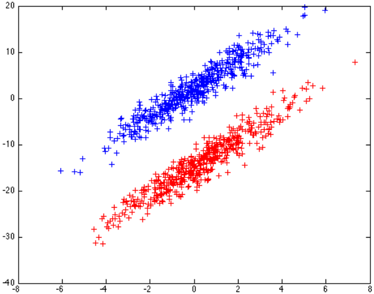
Copernicus master

What is binary classification ?



- From a learning ensemble :
  - Learn a model to classify data (i.e. assign label)
- Once the model is learned, a label can be assigned to any new point

What is binary classification ?



- From a learning ensemble :
  - Learn a model to classify data (i.e. assign label)
- Once the model is learned, a label can be assigned to any new point

Applications

Everywhere

- Medicine (decisions on illnesses, different behaviors against a virus, ...)
- Biology (classification animal / plant species)
- Languages (kind of languages)
- Web (dangerous images, social networks)
- Bank (Kind of customers)
- ...

$\Longrightarrow$ a large panel of approaches has been developed in all these disciplines

Applications

Everywhere

- Medicine (decisions on illnesses, different behaviors against a virus, ...)
- Biology (classification animal / plant species)
- Languages (kind of languages)
- Web (dangerous images, social networks)
- Bank (Kind of customers)
- ...

$\Longrightarrow$ a large panel of approaches has been developed in all these disciplines

Some notations

- We call the training set the N points with dimension d : $X = \{x_1, ..., x_N\}, x_i \in R^d$
- Each point is associated with a label $Y = \{y_1, ..., y_N\} \in N$
- Once the model learned, we aim at classifying (assign a label y_t) to a candidaet $x_t \in R^d$

Some notations

- We call the training set the N points with dimension d : $X = \{x_1, ..., x_N\}, x_i \in R^d$
- Each point is associated with a label $Y = \{y_1, ..., y_N\} \in N$
- Once the model learned, we aim at classifying (assign a label y_t) to a candidaet $x_t \in R^d$

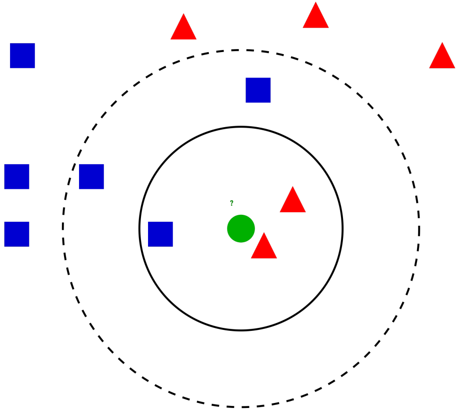
Some notations

- We call the training set the N points with dimension d : $X = \{x_1, ..., x_N\}, x_i \in R^d$
- Each point is associated with a label $Y = \{y_1, ..., y_N\} \in N$
- Once the model learned, we aim at classifying (assign a label y_t) to a candidaet $x_t \in R^d$

Outline

Algorithme des k-plus proches voisins

Goal : assign to a candidate x_t the majority class with respect to its K nearest neighboors in the training set

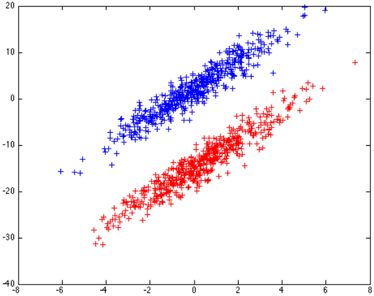


source : wikipedia

Outline

Perception : binary classification

Goal : find the equation of the hyperplan between the two classes

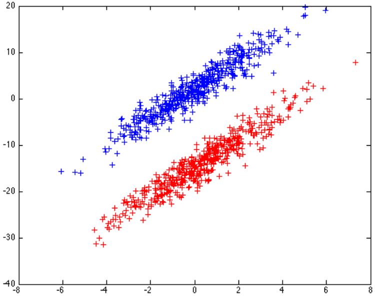


$\Longrightarrow$ We assume labels $Y \in \{-1, +1\}$

$\Longrightarrow$ The problem must be separable

Perception : binary classification

Goal : find the equation of the hyperplan between the two classes

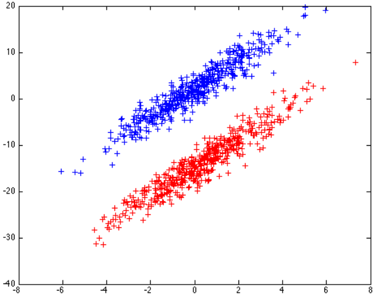


$\Longrightarrow$ We assume labels $Y \in \{-1, +1\}$

$\Longrightarrow$ The problem must be separable

Perception : binary classification

Goal : find the equation of the hyperplan between the two classes



$\Longrightarrow$ We assume labels $Y \in \{-1, +1\}$

$\Longrightarrow$ The problem must be separable

Perception : binary classification

- Equation of a hyperplan in dimension d :

$$h(x) = \sum_{i=1}^{d} w_i x_i + w_0 = \tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}} + w_0 = \boldsymbol{w}^T \boldsymbol{x}$$

with

- $\tilde{\boldsymbol{w}} = [w_1, ..., w_d]^T$
- $\tilde{\boldsymbol{x}} = [x_1, ..., x_d]^T$
- $\boldsymbol{w} = [w_0, w_1, ..., w_d]^T$
- $\boldsymbol{x} = [1, x_1, ..., x_d]^T$

CF dashboard

Perception : binary classification

- Equation of a hyperplan in dimension d :

$$h(x) = \sum_{i=1}^{d} w_i x_i + w_0 = \tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}} + w_0 = \boldsymbol{w}^T \boldsymbol{x}$$

  with

  - $\tilde{\boldsymbol{w}} = [w_1, ..., w_d]^T$
  - $\tilde{\boldsymbol{x}} = [x_1, ..., x_d]^T$
  - $\boldsymbol{w} = [w_0, w_1, ..., w_d]^T$
  - $\boldsymbol{x} = [1, x_1, ..., x_d]^T$

CF dashboard

Perception : binary classification

Principe

1. Initialisation : $\boldsymbol{w} = 0$
2. Loop over all the points
   1. Compute the value of (x_i) on the hyperplan : $P = \boldsymbol{w}^T \tilde{x}_i$ and multiply by y_i
   2. If $sign(Py_i) == 1$, we do nothing
   3. Otherwise, we modify the normal with $\boldsymbol{w} = \boldsymbol{w} + y_i \tilde{x}_i$

For any new data x_t (without label), a decision is taken with the following rule

- If $\boldsymbol{w}^T \tilde{x}_t > 0$, then $y_t = +1$
- Otherwise $y_t = -1$

Important note : only dot product are involved

Perception : binary classification

Principe

1. Initialisation : $\boldsymbol{w} = 0$
2. Loop over all the points
   1. Compute the value of (x_i) on the hyperplan : $P = \boldsymbol{w}^T \tilde{x}_i$ and multiply by y_i
   2. If $sign(Py_i) == 1$, we do nothing
   3. Otherwise, we modify the normal with $\boldsymbol{w} = \boldsymbol{w} + y_i \tilde{x}_i$

For any new data x_t (without label), a decision is taken with the following rule

- If $\boldsymbol{w}^T \tilde{x}_t > 0$, then $y_t = +1$
- Otherwise $y_t = -1$

Important note : only dot product are involved

Perception : binary classification

Principe

1. Initialisation : $\boldsymbol{w} = 0$
2. Loop over all the points
   1. Compute the value of (x_i) on the hyperplan : $P = \boldsymbol{w}^T \tilde{x}_i$ and multiply by y_i
   2. If $sign(Py_i) == 1$, we do nothing
   3. Otherwise, we modify the normal with $\boldsymbol{w} = \boldsymbol{w} + y_i \tilde{x}_i$

For any new data x_t (without label), a decision is taken with the following rule

- If $\boldsymbol{w}^T \tilde{x}_t > 0$, then $y_t = +1$
- Otherwise $y_t = -1$

Important note : only dot product are involved

Perception : binary classification

Principe

1. Initialisation : $\boldsymbol{w} = 0$
2. Loop over all the points
   1. Compute the value of (x_i) on the hyperplan : $P = \boldsymbol{w}^T \tilde{x}_i$ and multiply by y_i
   2. If $sign(Py_i) == 1$, we do nothing
   3. Otherwise, we modify the normal with $\boldsymbol{w} = \boldsymbol{w} + y_i \tilde{x}_i$

For any new data x_t (without label), a decision is taken with the following rule

- If $\boldsymbol{w}^T \tilde{x}_t > 0$, then $y_t = +1$
- Otherwise $y_t = -1$

Important note : only dot product are involved

Perception : binary classification

Principe

1. Initialisation : $\boldsymbol{w} = 0$
2. Loop over all the points
   1. Compute the value of (x_i) on the hyperplan : $P = \boldsymbol{w}^T \tilde{x}_i$ and multiply by y_i
   2. If $sign(Py_i) == 1$, we do nothing
   3. Otherwise, we modify the normal with $\boldsymbol{w} = \boldsymbol{w} + y_i \tilde{x}_i$

For any new data x_t (without label), a decision is taken with the following rule

- If $\boldsymbol{w}^T \tilde{x}_t > 0$, then $y_t = +1$
- Otherwise $y_t = -1$

Important note : only dot product are involved

Perception : binary classification

Principe

1. Initialisation : $\boldsymbol{w} = 0$
2. Loop over all the points
   1. Compute the value of (x_i) on the hyperplan : $P = \boldsymbol{w}^T \tilde{x}_i$ and multiply by y_i
   2. If $sign(Py_i) == 1$, we do nothing
   3. Otherwise, we modify the normal with $\boldsymbol{w} = \boldsymbol{w} + y_i \tilde{x}_i$

For any new data x_t (without label), a decision is taken with the following rule

- If $\boldsymbol{w}^T \tilde{x}_t > 0$, then $y_t = +1$
- Otherwise $y_t = -1$

Important note : only dot product are involved

Perception : binary classification

Principe

1. Initialisation : $\boldsymbol{w} = 0$
2. Loop over all the points
   1. Compute the value of (x_i) on the hyperplan : $P = \boldsymbol{w}^T \tilde{x}_i$ and multiply by y_i
   2. If $sign(Py_i) == 1$, we do nothing
   3. Otherwise, we modify the normal with $\boldsymbol{w} = \boldsymbol{w} + y_i \tilde{x}_i$

For any new data x_t (without label), a decision is taken with the following rule

- If $\boldsymbol{w}^T \tilde{x}_t > 0$, then $y_t = +1$
- Otherwise $y_t = -1$

Important note : only dot product are involved

Perception : exercise

Perform perceptron algorithm with

1. Initialisation : $\boldsymbol{w} = [1, 1, 1]^T$
2. $x_1 = [-1, 1]^T$, label $=$ 1
3. $x_2 = [0, 0]^T$, label $=$ -1
4. $x_3 = [1, 0]^T$, label $=$ -1
5. $x_4 = [0, 1]^T$, label $=$ 1
6. $x_5 = [-1, -1]^T$, label $=$ -1
7. $x_6 = [1, 2]^T$, label $=$ 1

Outline

SVM

Acronyms

- Support Vector Machines
- in French : "Séparateurs à Vaste Marge" (large margin separator)

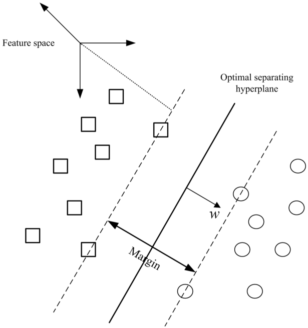
SVM

Acronyms

- Support Vector Machines
- in French : "Séparateurs à Vaste Marge" (large margin separator)

Principles

- Identical to perceptron : find a separator
- Main idea : maximize the margin



SVM

Notations

- Equation of a hyperplan h :

$$h(x) = \tilde{\boldsymbol{w}}^T x + w_0$$

- Distance from a point x to a hyperplan h :

$$d(x,h) = \frac{|\tilde{\boldsymbol{w}}^T x + w_0|}{||\tilde{\boldsymbol{w}}||}$$

- Functional margin of a point x_i associated with a label y_i :

$$\gamma_i = y_i(\tilde{\boldsymbol{w}}^T x + w_0)$$

- Geometrical margin of a point x_i associated with a label y_i :

$$\mu = y_i \frac{(\tilde{\boldsymbol{w}}^T x + w_0)}{||\tilde{\boldsymbol{w}}||}$$

SVM

Notations

- Equation of a hyperplan h :

$$h(x) = \tilde{\boldsymbol{w}}^T x + w_0$$

- Distance from a point x to a hyperplan h :

$$d(x,h) = \frac{|\tilde{\boldsymbol{w}}^T x + w_0|}{\|\tilde{\boldsymbol{w}}\|}$$

- Functional margin of a point x_i associated with a label y_i :

$$\gamma_i = y_i(\tilde{\boldsymbol{w}}^T x + w_0)$$

- Geometrical margin of a point x_i associated with a label y_i :

$$\mu = y_i \frac{(\tilde{\boldsymbol{w}}^T x + w_0)}{\|\tilde{\boldsymbol{w}}\|}$$

SVM

Notations

- Equation of a hyperplan h :

$$h(x) = \tilde{\boldsymbol{w}}^T x + w_0$$

- Distance from a point x to a hyperplan h :

$$d(x, h) = \frac{|\tilde{\boldsymbol{w}}^T x + w_0|}{\|\tilde{\boldsymbol{w}}\|}$$

- Functional margin of a point x_i associated with a label y_i :

$$\gamma_i = y_i(\tilde{\boldsymbol{w}}^T x + w_0)$$

- Geometrical margin of a point x_i associated with a label y_i :

$$\mu = y_i \frac{(\tilde{\boldsymbol{w}}^T x + w_0)}{\|\tilde{\boldsymbol{w}}\|}$$

SVM

Notations

- Equation of a hyperplan h :

$$h(x) = \tilde{\boldsymbol{w}}^T x + w_0$$

- Distance from a point x to a hyperplan h :

$$d(x, h) = \frac{|\tilde{\boldsymbol{w}}^T x + w_0|}{\|\tilde{\boldsymbol{w}}\|}$$

- Functional margin of a point x_i associated with a label y_i :

$$\gamma_i = y_i(\tilde{\boldsymbol{w}}^T x + w_0)$$

- Geometrical margin of a point x_i associated with a label y_i :

$$\mu = y_i \frac{(\tilde{\boldsymbol{w}}^T x + w_0)}{\|\tilde{\boldsymbol{w}}\|}$$

COPERNICUS MASTER IN DIGITAL EARTH

SVM

Notations

- Canonical hyperplan associated with a dataset $X = \{x_1, ..., x_N\}$:

$$\min_{x_i} |\tilde{\boldsymbol{w}}^T x_i + w_0| = 2$$

- Optimal canonical hyperplan associated with a dataset $X = \{x_1, ..., x_N\}$:
  - it maximizes the distances between itself and all data

$$\forall i \in \{1, ..., N\}, y_i h(x_i) > 1$$

- Margin of a hyperplan associated with the dataset :

$$M = \frac{2}{\|\tilde{\boldsymbol{w}}\|}$$

SVM

Notations

- Canonical hyperplan associated with a dataset $X = \{x_1, ..., x_N\}$:

$$\min_{x_i} |\tilde{\boldsymbol{w}}^T x_i + w_0| = 2$$

- Optimal canonical hyperplan associated with a dataset $X = \{x_1, ..., x_N\}$:
  - it maximizes the distances between itself and all data

$$\forall i \in \{1, ..., N\}, y_i h(x_i) > 1$$

- Margin of a hyperplan associated with the dataset :

$$M = \frac{2}{\|\tilde{w}\|}$$

SVM

Notations

- Canonical hyperplan associated with a dataset $X = \{x_1, ..., x_N\}$:

$$\min_{x_i} |\tilde{\boldsymbol{w}}^T x_i + w_0| = 2$$

- Optimal canonical hyperplan associated with a dataset $X = \{x_1, ..., x_N\}$:
  - it maximizes the distances between itself and all data

$$\forall i \in \{1, ..., N\}, y_i h(x_i) > 1$$

- Margin of a hyperplan associated with the dataset :

$$M = \frac{2}{\|\tilde{\boldsymbol{w}}\|}$$

SVM

Main equation

- We aim at maximizing the margin, i.e. find the hyperplan of maximal margin :

$$\frac{1}{2}\min_{\tilde{\boldsymbol{w}}} \|\tilde{\boldsymbol{w}}\|$$

Under the constraints

$$y_i\big(< \tilde{\boldsymbol{w}}, x_i > + w_0\big) \geqslant 1, \quad \forall i \in \{1, ..., N\}$$

cf démo

The solution depends only on some points, named support vectors

Important note : only dot products are involved

SVM

Main equation

- We aim at maximizing the margin, i.e. find the hyperplan of maximal margin :

$$\frac{1}{2}\min_{\tilde{\boldsymbol{w}}} \|\tilde{\boldsymbol{w}}\|$$

Under the constraints

$$y_i\big(< \tilde{\boldsymbol{w}}, x_i > +w_0\big) \geqslant 1, \quad \forall i \in \{1, ..., N\}$$

cf démo

The solution depends only on some points, named support vectors

Important note : only dot products are involved

SVM

Main equation

- We aim at maximizing the margin, i.e. find the hyperplan of maximal margin :

$$\frac{1}{2}\min_{\tilde{\boldsymbol{w}}} \|\tilde{\boldsymbol{w}}\|$$

Under the constraints

$$y_i\big(< \tilde{\boldsymbol{w}}, x_i > + w_0 \big) \geqslant 1, \quad \forall i \in \{1, ..., N\}$$

cf démo

The solution depends only on some points, named support vectors

Important note : only dot products are involved

SVM

Main equation

- We aim at maximizing the margin, i.e. find the hyperplan of maximal margin :

$$\frac{1}{2} \min_{\tilde{\boldsymbol{w}}} \|\tilde{\boldsymbol{w}}\|$$

Under the constraints

$$y_i \big(< \tilde{\boldsymbol{w}}, x_i > + w_0 \big) \geqslant 1, \quad \forall i \in \{1, ..., N\}$$

cf démo

The solution depends only on some points, named support vectors

Important note : only dot products are involved

SVM

Main equation

- We aim at maximizing the margin, i.e. find the hyperplan of maximal margin :

$$\frac{1}{2} \min_{\tilde{\boldsymbol{w}}} \|\tilde{\boldsymbol{w}}\|$$

Under the constraints

$$y_i \big(< \tilde{\boldsymbol{w}}, x_i > + w_0 \big) \geqslant 1, \quad \forall i \in \{1, ..., N\}$$

cf démo

The solution depends only on some points, named support vectors

Important note : only dot products are involved

SVM

SVM with soft margin

- We aim at maximizing the margin but we relax the constraint for some points :

$$\min_{\tilde{\boldsymbol{w}}, w_0} \|w_t\| + C \sum_i \xi_i$$

Under the constraints

$$y_i\big(< \tilde{\boldsymbol{w}}, x_i > + w_0 \big) \geqslant 1 - \xi_i, \quad \forall i \in \{1, ..., N\}$$

et

$$\xi_i > 0, \quad \forall i \in \{1, ..., N\}$$

- where $C > 0$ is constant

SVM

SVM with soft margin

- We aim at maximizing the margin but we relax the constraint for some points :

$$\min_{\tilde{\boldsymbol{w}}, w_0} \|w_t\| + C \sum_i \xi_i$$

Under the constraints

$$y_i \big(< \tilde{\boldsymbol{w}}, x_i > + w_0 \big) \geqslant 1 - \xi_i, \quad \forall i \in \{1, ..., N\}$$

et

$$\xi_i > 0, \quad \forall i \in \{1, ..., N\}$$

- where $C > 0$ is constant

SVM

SVM with soft margin

- We aim at maximizing the margin but we relax the constraint for some points :

$$\min_{\tilde{\boldsymbol{w}}, w_0} \|w_t\| + C \sum_i \xi_i$$

Under the constraints

$$y_i\big(< \tilde{\boldsymbol{w}}, x_i > + w_0\big) \geqslant 1 - \xi_i, \quad \forall i \in \{1, ..., N\}$$

et

$$\xi_i > 0, \quad \forall i \in \{1, ..., N\}$$

- where $C > 0$ is constant