Support Vector Machines

Kairit Sirts

02.05.2014

Keywords

- Functional and geometrical margins
- Maximal margin classifier
- Soft margin classifier
- Support vectors

Perceptron algorithm

$w_d \leftarrow 0$, for all $d = 1 \dots D$
$b \leftarrow 0$
for $iter = 1 \dots MaxIter$ **do**
    for all $(\mathbf{x}, y) \in \mathbf{D}$ **do**
        $a \leftarrow \sum_{d=1}^{D} w_d x_d + b$
        if $ya \leq 0$ **then**
            $w_d \leftarrow w_d + y x_d$, for all $d = 1 \dots D$
            $b \leftarrow b + y$
        end if
    end for
end for
return $w_1, \dots, w_D, b$

Perceptron properties

- Error-driven algorithm
- Learns a linear decision boundary
- Is guaranteed to find the solution with linearly separable data only
- Model and algorithm are together

Neural Networks

- Enable to learn non-linear decision boundaries
- Two-layer NN can be used to approximate any function (George Cybenko)
- Many hyperparameters (topology of the network, activation function)
- Non-convex optimization task (sensitive to initialization)

Support Vector Machines

Support vector machines have several nice features:

- Convex optimization task (only one optimum)
- Proven generalization bounds
- Resistant to overfitting
  - The number of features can be bigger than the number of training examples
- Enables to learn non-linear decision boundaries with linear model

Notation

$\mathbf{X} \in \mathbb{R}^{m \times n}$	matrix of inputs (design matrix)
$\mathbf{y} \in \{-1, +1\}^m$	vector of labels for each input
$\mathbf{w} \in \mathbb{R}^n$	vector of weights
$b \in \mathbb{R}$	bias term
$h_{\mathbf{w},b}(\mathbf{x}) = g(\mathbf{w}^T\mathbf{x} + b)$	hypothesis
$g(z) = 1$	if $z \geq 0$
$g(z) = -1$	otherwise

Functional margin

- Assume we have linearly separable data
- **Functional margin** of an example $(\mathbf{x}_i, y_i)$ with respect to a hyperplane $(\mathbf{w}, b)$ is defined as:

$$\hat{\gamma}_i = y_i(\mathbf{w}^T \mathbf{x}_i + b)$$

  $\hat{\gamma}_i$ is positive if y_i and $\mathbf{w}^T \mathbf{x}_i + b$ have the same sign
- Thus, $\hat{\gamma}_i > 0$ implies correct classification of $(\mathbf{x}_i, y_i)$
- The larger the functional margin the more confident the prediction

Functional margin

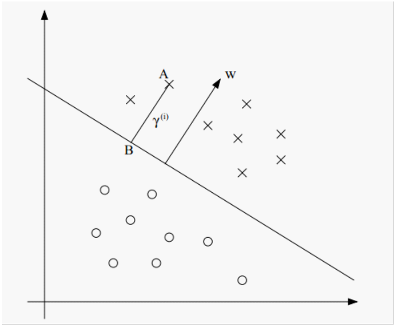
- The function margin of the hyperplane $(\mathbf{w}, b)$ with respect to some training set is the smallest functional margin of the individual training examples:

$$\hat{\gamma} = \min_{i=1,\ldots,m} \hat{\gamma}_i$$

- Functional margin is not a very good measure for confidence, because:
  - Both h and g depend only on the sign of $\mathbf{w}^T\mathbf{x} + b$
  - Rescaling $\mathbf{w}$ and b does not change their values
  - Thus the functional margin could be made arbitrarily large

Geometric margin

Geometric margin of an example $(\mathbf{x}_i, y_i)$ with respect to a hyperplane $(\mathbf{w}, b)$ is the Euclidean distance between the point $\mathbf{x}_i$ and the hyperplane:



- $\mathbf{w}$ is perpendicular to the hyperplane
- γ_i is the length of the segment AB
- $\mathbf{w}/\|\mathbf{w}\|$ is a unit vector
- A is some point $\mathbf{x}_i$
- $B = \mathbf{x}_i - \gamma_i \cdot \mathbf{w}/\|\mathbf{w}\|$

Geometric margin

- Point B lies on the separating hyperplane and for all points lying there:

$$\mathbf{w}^T\mathbf{x} + b = 0$$

- Therefore:

$$\mathbf{w}^T\left(\mathbf{x}_i - \gamma_i \frac{\mathbf{w}}{\|\mathbf{w}\|}\right) + b = 0$$

- Solving for γ_i yields:

$$\gamma_i = \left(\frac{\mathbf{w}}{\|\mathbf{w}\|}\right)^T \mathbf{x}_i + \frac{b}{\|\mathbf{w}\|}$$

Geometric margin

- For both positive and negative training examples the geometric margin is:

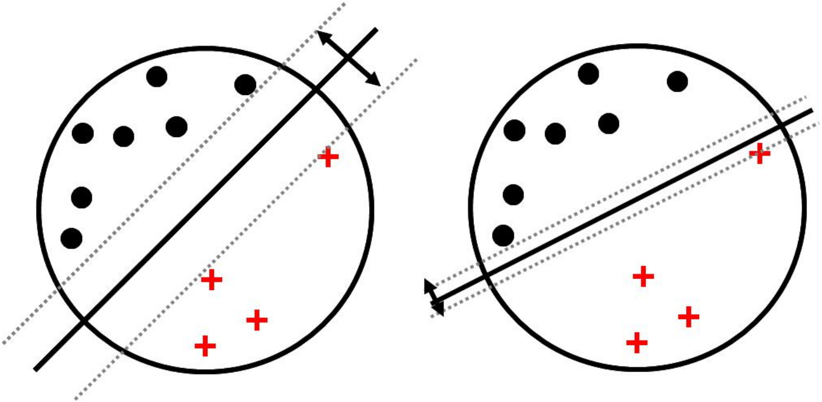
$$\gamma_i = y_i \left(\left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \right)^T \mathbf{x}_i + \frac{b}{\|\mathbf{w}\|} \right)$$

- If $\|\mathbf{w}\|$ is one then the geometric margin and functional margin are equal
- Geometric margin is invariant to the rescaling of the parameters
- Geometric margin of a hyperplane $(\mathbf{w}, b)$ with respect to a training set is the minimum geometric margin of the training examples:

$$\gamma = \min_{i=1,\dots,m} \gamma_i$$

Maximal margin

- **Margin of a training set** is the maximum geometric margin over all separating hyperplanes.
- The hyperplane realising the maximum is called **maximal margin hyperplane**.



Maximum margin classifier: hard margin SVM

- **Idea:** Find the optimal separating hyperplane by maximizing the geometric margin of the training set $\gamma(\mathbf{w}, b)$.
- For ensuring that the margin separates the data points, we also need the constraints imposed on functional margins
- We require functional and geometric margin to be equal, then the geometric margin for each point is at least γ
- This leads to the following optimization problem:

$$\max_{\mathbf{w},b} \gamma$$

$$\text{s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq \gamma, \text{ for all } i$$

$$\|\mathbf{w}\| = 1$$

Hard margin SVM

- The last constraint is non-convex
- We can remove it by changing the optimization problem:

$$\max_{\mathbf{w},b} \frac{\hat{\gamma}}{\|\mathbf{w}\|}$$
$$\text{s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq \hat{\gamma}, \text{ for all } i$$

- Recall that $\gamma = \hat{\gamma}/\|\mathbf{w}\|$
- The objective is still non-convex

Hard margin SVM

- Recall that geometric margin is invariant to scaling
- We now introduce the scaling constraint that the functional margin of the hyperplane $(\mathbf{w}, b)$ with respect to the training set must be 1:

$$\hat{\gamma} = 1$$

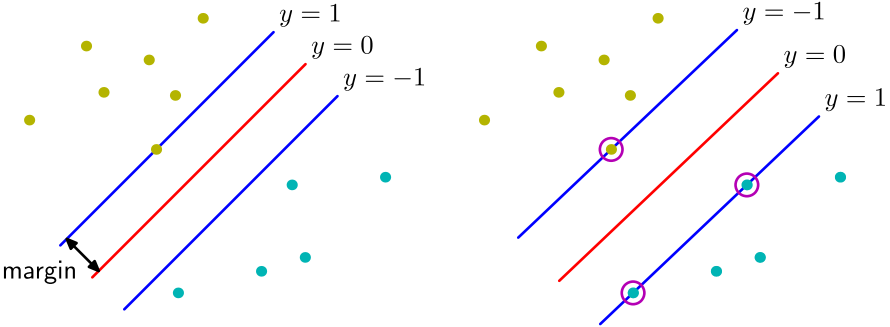
- We plug that in to the optimization problem and turn maximization into minimization:

$$\min_{\mathbf{w},b} \|\mathbf{w}\|$$
$$\text{s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \text{ for all } i$$

- Thus we require all data points to be correctly classified and to have the functional margin at least 1.

Support vectors

The points lying exactly on the maximal margin are **support vectors**



Soft margin SVM

- What if the data is not linearly separable?
- This means that some of the datapoints fail the margin (the functional margin is negative)
- The **slack variables** measure how much each of the points fails to meet the target of having a positive margin:

$$\xi_i = \max\left(0, \hat{\gamma} - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right)$$

Soft margin SVM

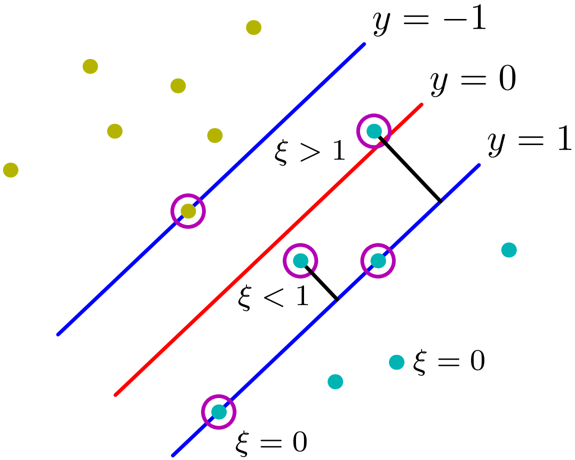
- In addition to maximizing the margin we now also want to minimize the sum of the slack variables:

$$\min_{\mathbf{w},b,\xi} \|\mathbf{w}\| + C \sum_i \xi_i$$

- The constrains now have to take the slack into account:

$$\begin{aligned} y_i(\mathbf{w}^T \mathbf{x}_i + b) &\geq 1 - \xi_i, \quad && \text{for all } i \\ \xi_i &\geq 0, && \text{for all } i \end{aligned}$$

Soft margin SVM



Objective function for both hard and soft margin

- For hard margin:

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1, \text{ for all } i$$

- For soft margin:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i$$

$$\begin{aligned} y_i(\mathbf{w}^T\mathbf{x}_i + b) &\geq 1 - \xi_i, &&\text{for all } i \\ \xi_i &\geq 0, &&\text{for all } i \end{aligned}$$

- These are convex quadratic optimization problems with linear constraints and can be solved by quadratic programming.

Multiclass SVM

- SVM is fundamentally a two-class classifier
- For building multiclass SVM-s there are several methods:
  - K one-versus-all SVM-s
  - all-versus-all approach - $K(K-1)/2$ classifiers
  - Several more complicated problems
  - Still an open problem