Linear support vector machines

Nathanaël Carraz Rakotonirina

Mathématiques Informatique et Statistique Appliquées (MISA)
Université d'Antananarivo

Model

We want to find a binary classifier with a linear decision boundary :

$$f(x) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

The classifier assigns

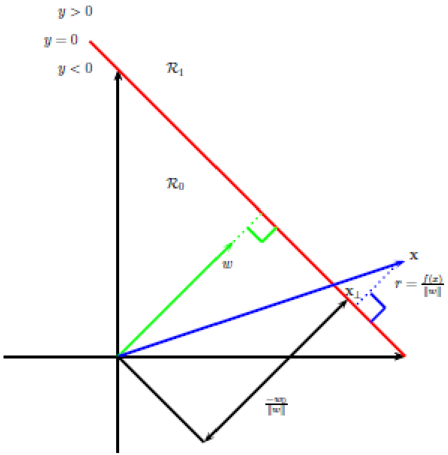
$$\hat{y} = \begin{cases} 1 & \text{if } f(\boldsymbol{x}) > 0 \\ -1 & \text{if } f(\boldsymbol{x}) < 0 \end{cases}$$

As in logistic regression, the decision boundary is a hyperplane (with normal vector $\boldsymbol{w}$ and offset from the origin b) separating the space into 2 half-spaces. We first assume the data is linearly separable.

Goal

We want to find the classifier with the maximum **margin** (the distance of the closest point to the decision boundary).

Distance to the decision boundary



$$\boldsymbol{x} = \boldsymbol{x}_\perp + d\frac{\boldsymbol{w}}{||\boldsymbol{w}||}$$

where d is the distance of $\boldsymbol{x}$ from the decision boundary and $\boldsymbol{x}_\perp$ is the orthogonal projection of $\boldsymbol{x}$ onto this boundary.

$$f(\boldsymbol{x}) = (\boldsymbol{w}^\top \boldsymbol{x}_\perp + b) + d||\boldsymbol{w}||$$

since $\boldsymbol{x}_\perp$ is on the hyperplane $\boldsymbol{w}^\top \boldsymbol{x}_\perp + b = 0$ thus $f(\boldsymbol{x}) = d||\boldsymbol{w}||$ and hence

$$d = \frac{f(\boldsymbol{x})}{||\boldsymbol{w}||} = \frac{\boldsymbol{w}^\top \boldsymbol{x} + b}{||\boldsymbol{w}||}$$

Large margin classifier

We want the classifier to :

- maximize the margin which is $\min_{i=1}^{N} f(\boldsymbol{x}_i)/||\boldsymbol{w}||$
- correctly classify each data point $f(\boldsymbol{x}_i)y_i > 0$

The objective is

$$\max_{\boldsymbol{w},b} \frac{1}{||\boldsymbol{w}||} \min_{i=1}^{N} \left[y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)\right]$$

We can rescale the parameters without changing the objective. The scale factor is defined such that $\min_{i=1}^{N} f(\boldsymbol{x}_i) = 1$. Maximizing $1/||\boldsymbol{w}||$ is equivalent to minimizing $||\boldsymbol{w}||^2$. The new objective is

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

$$\text{s.t } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, i = 1 \ldots N$$

Dual problem

Let $\alpha_i, i = 1...N$ ($\boldsymbol{\alpha} \in \mathbb{R}^N$) the Lagrange multipliers of the inequality constraints. The Lagrangian is

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} - \sum_{i=1}^{N} \alpha_i \left[y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1\right]$$

We want to find $(\hat{\boldsymbol{w}}, \hat{b}, \hat{\boldsymbol{\alpha}}) = \min_{\boldsymbol{w},b} \max_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha})$
By KKT stationarity conditions, $\frac{\partial \mathcal{L}(\boldsymbol{w},b,\boldsymbol{\alpha})}{\partial \boldsymbol{w}} = 0$ and $\frac{\partial \mathcal{L}(\boldsymbol{w},b,\boldsymbol{\alpha})}{\partial b} = 0$ gives :

$$\hat{\boldsymbol{w}} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

Substituting these back into the Lagrangian gives the dual problem

$$\max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j + \sum_{i=1}^{N}\alpha_i$$

$$\text{s.t } \alpha_i \geq 0, i = 1 \ldots N$$

$$\sum_{i=1}^{N}\alpha_i y_i = 0$$

which can be solved by standard quadratic programming solvers.

Support vectors

The other KKT conditions must be satisfied:

- $\alpha_i \geq 0$
- $y_i f(\boldsymbol{x}_i) - 1 \geq 0$
- $\alpha_i (y_i f(\boldsymbol{x}_i) - 1) = 0$

We may have one of the following situations :

- for samples x_i such that $y_i f(\boldsymbol{x}_i) - 1 > 0$ or $y_i f(\boldsymbol{x}_i) > 1$, we must have $\alpha_i = 0$ (inactive constraint)
- for samples x_i such that $y_i f(\boldsymbol{x}_i) - 1 = 0$ or $y_i f(\boldsymbol{x}_i) = 1$, we must have $\alpha_i > 0$ (active constraint)

The points of the active constraint lie on the decision boundary. These are called **support vectors**. The value of $\hat{\boldsymbol{w}}$ depends only on these points.

Solve for b

For any support vector, we have $y_i f(\boldsymbol{x}_i) = 1$. By multiplying both sides by y_i and using $y_i^2 = 1$, we have

$$\hat{b} = y_i - \hat{\boldsymbol{w}}^\top \boldsymbol{x}_i$$

In practice we get better results by averaging over all the support vectors

$$\hat{b} = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} (y_i - \hat{\boldsymbol{w}}^\top \boldsymbol{x}_i)$$

where $\mathcal{S}$ is the set of the indices of the support vectors.

SVM in a nutshell

1. Solve the dual (using the training set) to get the optimal dual parameters $\hat{\alpha}_i, i = 1 \ldots N$
2. Compute $\hat{\boldsymbol{w}} = \sum_{i=1}^{N} \hat{\alpha}_i y_i \boldsymbol{x}_i$
3. Compute $\hat{b} = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} (y_i - \hat{\boldsymbol{w}}^\top \boldsymbol{x}_i)$
4. Compute the classification function for an example $\boldsymbol{x}$

$$f(\boldsymbol{x}) = \hat{\boldsymbol{w}} + \boldsymbol{x} + \hat{b} = \sum_{i=1}^{N} \hat{\alpha}_i y_i \boldsymbol{x}_i^\top \boldsymbol{x} + \hat{b} = \sum_{i \in \mathcal{S}} \hat{\alpha}_i y_i \boldsymbol{x}_i^\top \boldsymbol{x} + \hat{b}$$

5. Predict the label of $\boldsymbol{x}$ using

$$\hat{y} = \begin{cases} 1 & \text{if } f(\boldsymbol{x}) > 0 \\ -1 & \text{if } f(\boldsymbol{x}) < 0 \end{cases}$$

Soft margin classifier for non separable case

If the data is not linearly separable, there will be no feasible solution correctly classifying all training data points.
We introduce **slack variables** $\xi_i \geq 0$ and replace the hard constraints $y_i f(\boldsymbol{x}_i) \geq 1$ with the **soft margin constraints** $y_i f(\boldsymbol{x}_i) \geq 1 - \xi_i$. The new objective is

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \frac{1}{2}||\boldsymbol{w}||^2 + C \sum_{i=1}^{N} \xi_i$$

$$\text{s.t } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i, i = 1 \ldots N$$

$$\xi_i \geq 0, i = 1 \ldots N$$

where $C \geq 0$ is a hyperparameter controlling the trade-off between slack errors and the margin maximization.

The corresponding Lagrangian is

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i \left[y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1 + \xi_i \right] + \sum_{i=1}^{N} \mu_i \xi_i$$

with the Langrangian multipliers $\alpha_i \geq 0$ and $\mu_i \geq 0$. Optimizing $\boldsymbol{w}$, b and $\boldsymbol{\xi}$ gives the dual problem

$$\max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j + \sum_{i=1}^{N} \alpha_i$$

$$\text{s.t } 0 \leq \alpha_i \leq C, i = 1...N$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

which is the same as the linearly separable case except for the constraint on α_i. Once we get the optimal $\boldsymbol{\alpha}$. We proceed as we did before.

Multi-class classification

There are two common approaches to extend binary SVM multi-class:

one-vs-all

- For each class k, train a binary classifier (where the data from class k is treated as positive, and the data from all the other classes is treated as negative.)
- To classify, select $\arg\max_k \{f_1, ..., f_K\}$

one-vs-one (all pairs)

- Train $K(K-1)/2$ binary classifiers (discriminate all pairs f_k, k')
- To classify, select the class which has the highest number of votes.

There are ambiguities as well as other issues associated with both methods.

What's next ?

Explore

- Kernel machines
- SVM for regression
- SVM outputs into probabilities
- Other variants of SVM