Machine Learning: Support Vector Machines: Linear Kernel, Dual Problem

Soft Margin SVMs (reprise): Linear and Non-Separable Cases

Recall the definition of the soft margin Support Vector Machine.

Let (X, Y), $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, $Y = \{y_1, \ldots, y_m\}$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$ be the training set.

Let $\mathbf{w} = (w_1, \ldots, w_n)$, and b be the linear coefficients and the intercept for a function

$L(\mathbf{w}, b) = \mathbf{x}^T\mathbf{w} + b$.

The soft margin Support Vector Machine optimization problem is described below:

Objective Function:

$$J(\mathbf{w}, b) = \min_{\mathbf{w}, b, \{\xi_i\}} \left(\frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^{n} \xi_i \right)$$

Subject to constraints:

$$Q_1 : y_i(\mathbf{w} \cdot \bar{x}_i + b) \geq 1 - \xi_i, \forall \bar{x}_i \in X$$

Training SVM Classifiers

Step 1. Get rid of the intercept. This can be accomplished by replacing all vectors $\bar{x} = (a_1, \ldots, a_d) \in X$ with the vectors $\bar{x'} = (a_1, \ldots, a_d, 1)$, and replacing the vector of weights $\mathbf{w} = (w_1, \ldots, w_d)$ with the vector $\mathbf{w}' = (w_1, \ldots, w_d, b)$. (This is a standard procedure that we have seen multiple times already).

Without loss of generality, we assume that all vectors $\mathbf{w}$ and $\bar{x}$ mentioned below have gone through this transformation.

Step 2. Pick the problem to optimize. There are two SVM problems that can be solved: *primal* and *dual*.

We address the solution of the **dual problem** here.

Step 3. Introduce Lagrangian Multipliers. One approach to optimizing a function $f(\mathbf{x})$ subject to some constraints $Q_1, \ldots, Q_k$ of the form $Q_i : g_i(\mathbf{x}) \geq 0$ is to consider optmizing a function

$$L(\alpha) = f(\mathbf{x}) + \sum_{i=1}^{k} \alpha_k g_i(\mathbf{x}).$$

Here, the values α_i are called Lagrangian multipliers and can be thought of as the *penalties assessed for the value* $\mathbf{x}$ *not satisfying the constraints* $g_i(\mathbf{x})$.

Essentially, $\alpha_1, \ldots, \alpha_k$ make it *difficult* for $L()$ to reach its optimal value at a point $\mathbf{x}$ where constraints $g_1(\mathbf{x}), \ldots, g_k(\mathbf{x})$ are violated.

We apply this approach to the problem of optimizing the soft margin SVM function as follows.

We replace the problem of optimizing $J(\mathbf{w}, b)$ subject to constraints $Q_1, \ldots, Q_m$ with the problem of optimizing the Lagrangian function:

$$L = \frac{||\mathbf{w}||^2}{2} + C\sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i\right) - \sum_{i=1}^{m} \beta_i \xi_i,$$

where $\alpha_1, \ldots, \alpha_m$ are Lagrangian multipliers applied to the constraints $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i \geq 0$, and $\beta_1, \ldots, \beta_m$ are Lagrangian multipliers applied to the constraints $\xi_i \geq 0$.

When L is considered as a function of $\mathbf{w}, b$, and ξ, L reaches its optimal value at the point where

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x_i} = 0,$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0,$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0.$$

From these equations, we obtain:

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x_i},$$

and

$$\beta_i = C - \alpha_i.$$

Noticing that $\alpha_i \neq 0$ implies that $\mathbf{x_i}$ is a support vector, the first equality can be interpreted as

> The vector $\mathbf{w}$ defining the separating plane (i.e., the normal vector to the plane) is determined as a linear combination of the **support vectors** for the plane.

Under the assumption that we reached the optimum values of $\mathbf{w}$, b and $\xi_1, \ldots, \xi_m$, we can eliminate these from the Lagrangian function L and construct a dual function:

$$L_{dual} = \frac{\mathbf{w}^T\mathbf{w}}{2} - \mathbf{w}^T \sum_{i=1}^{m} \alpha_i y_i \mathbf{x_i} + \sum_{i=1}^{n} \alpha_i y_i b - \sum_{i=1}^{m} \alpha_i + \sum_{i=1}^{m} (C - \alpha_i - \beta_i)\xi_i =$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\mathbf{w}^T\mathbf{w} = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \mathbf{x_i}\mathbf{x_j}.$$

i.e.,

$$L_{dual} = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}.$$

We need to **maximize** L_{dual} subject to the following constraints:

$0 \leq \alpha_i \leq C$ (because $C - \alpha_i = \beta_i \geq 0$) and

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

Step 4. Gradient Ascent/Stochastic Gradient Ascent. Our goal is to maximize

$$L_{dual}(\alpha) = J(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$$

subject to

$$0 \leq \alpha_i \leq C$$
$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

Given α_i, the part of $J(\mathbf{alpha})$ that depends on α_i can be written as follows:

$$J(\alpha_i) = \alpha_i - \frac{1}{2}\alpha_i^2 \mathbf{x_i}^T\mathbf{x_i} - \frac{1}{2}\alpha_i y_i \sum_{j=1, j\neq i} \alpha_j y_j \mathbf{x_i}^T\mathbf{x_j}$$

The gradient of $J(\alpha)$ is

$$\nabla J(\alpha) = \left(\frac{\partial J}{\partial \alpha_1}, \ldots, \frac{\partial J}{\partial \alpha_m}\right)$$

where

$$\frac{\partial J}{\partial \alpha_i} = 1 - y_i \left(\sum_{j=1} m \alpha_j y_j \mathbf{x_i}^T \mathbf{x_j} \right)$$

Gradient Ascent. The gradient ascent method proceeds as follows:

- $\alpha_0 = (0, 0, \ldots, 0)$ (or some other chosen set of initial values)
- $\alpha_{t+1} = \alpha_{\mathbf{t}} + \eta_t \nabla J(\alpha_{\mathbf{t}})$

Stochastic Gradient Ascent. Note that α_i coefficients represent to the impact of individual training set data points on the final shape of the function. These can be considered separately.

The update rule for the stochastic gradiaent ascent is

$$\alpha_i^{t+1} = \alpha_i^t + \eta_i \left(1 - y_k \sum_{j=1}^{m} \alpha_j^t y_j \mathbf{x_i}^T \mathbf{x_j} \right)$$

References

[1] Mohammed J. Zaki, Wagner Meira Jr., *Data Mining and Analysis: Fundamental Concepts and Algorithms*, Cambridge University Press, 2014.