Introduction to Sequential minimal optimization

Yinbin Ma

University of Illinois at Chicago

May 4, 2019

Overview

Introduction

Support Vector Machine aims to find a decision boundary with maximum margin. Lets $X \in \mathbb{R}^{N \times K}, y \in \mathbb{R}^N, y_i \in \{-1, 1\}, w \in \mathbb{R}^K, b \in \mathbb{R}$. The objective function is:

$$\min_{w,b,\xi} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{N} \xi_i$$
$$s.t. \quad y_i(w^T x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0$$

Lagrangian

To Solve it and find the optimal result, we start by defining the generalized Lagrangian.

$$
\begin{aligned}
\mathcal{L}(w, b, \xi, \alpha, \beta) &= \frac{1}{2} w^T w + C \sum_{i}^{N} \xi_i \\
&\quad - \sum_{i}^{N} \alpha_i \Big(y_i(w^T x_i + b) - 1 + \xi_i \Big) - \sum_{i}^{N} \beta_i \xi_i \\
s.t. \quad & \alpha_i \geq 0 \quad \beta_i \geq 0
\end{aligned}
$$

Primal Problem

Lets say $\theta_p(w) = \max_{\alpha,\beta:\alpha_i \geq 0} \mathcal{L}(w, b, \xi, \alpha, \beta)$, if w is given but it violates any of the constraints, we will have $\theta_p(w) \to \infty$. Hence

$$\theta_p(w) = \begin{cases} \frac{1}{2} w^T w & \text{if w satisfies constraints} \\ \infty & \text{otherwise} \end{cases}$$

Then we $\min_w \theta_p(w) = \min_w \max_{\alpha,\beta:\alpha_i \geq 0} \mathcal{L}(w, b, \xi, \alpha, \beta)$, and we get a qualified w. However $\theta_p(w)$ is trivial.

Dual Problem

Supposing we have $\theta_d(\alpha, \beta) = \min_w \mathcal{L}(w, b, \xi, \alpha, \beta)$, and it is shown that

$$d^* = \max_{\alpha, \beta : \alpha_i \geq 0} \theta_d(\alpha, \beta) \leq \min_w \theta_p(w) = p^*$$

Under certain conditions, we could have $d^* = p^*$, so we could solve dual problem instead of primal problem. The conditions are called **Karush-Kuhn-Tucker (KKT) conditions**.

Karush-Kuhn-Tucker Conditions

Consider the following, which well call the primal optimization problem:

$$\begin{aligned} \min_{w} \quad & f(w) \\ s.t. \quad & g_i(w) \leq 0, i = 1, \ldots, k \\ & h_i(w) = 0, i = 1, \ldots, p \end{aligned}$$

Suppose f and each g_i are convex, and each h_i is affine which means linear. Suppose further that there exists some w so that $g_i(w) < 0$ for all i.

Karush-Kuhn-Tucker Conditions

The corresponding Lagrangian is

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{p} \beta_i h_i(w)$$

Under above assumptions, there must exists w^*, α^*, β^*, so that w^* is the solution to the primal problem, α^*, β^* are the solution to the dual problem, and moreover $p^* = d^* = \mathcal{L}(w^*, \alpha^*, \beta^*)$. Further more, we have:

$$\begin{aligned} \triangledown_{w_i} \mathcal{L}(w^*, \alpha^*, \beta^*) &= 0 && (1) \\ \triangledown_{\beta_i} \mathcal{L}(w^*, \alpha^*, \beta^*) &= 0 && (2) \\ \alpha_i^* g_i(w^*) &= 0 && (3) \\ g_i(w^*) &\leq 0 && (4) \\ \alpha^* &\geq 0 && (5) \end{aligned}$$

Optimizing

$$\mathcal{L}(w, b, \xi, \alpha, \beta) = \frac{1}{2}w^T w + C\sum_i^N \xi_i$$
$$- \sum_i^N \alpha_i \Big(y_i(w^T x_i + b) - 1 + \xi_i \Big) - \sum_i^N \beta_i \xi_i$$
$$s.t. \quad \alpha_i \geq 0 \quad \beta_i \geq 0$$

Lets take some derivatives:

$$\bigtriangledown_w \mathcal{L}(w, b, \xi, \alpha, \beta) = w - \sum_{i=1}^N \alpha_i y_i x_i = 0$$

$$\bigtriangledown_b \mathcal{L}(w, b, \xi, \alpha, \beta) = -\sum_{i=1}^N \alpha_i y_i = 0$$

$$\bigtriangledown_{\xi_i} \mathcal{L}(w, b, \xi, \alpha, \beta) = C - \alpha_i - \beta_i = 0$$

Optimizing

So we know that

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \quad C - \alpha_i - \beta_i = 0$$

Take these equations and plug them back into the Lagrangian, and simplify. Then we obtain the following dual optimization problem:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{i=1}^{N} \alpha_i \\ s.t. \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C \end{aligned}$$

Optimizing

$$
\begin{aligned}
\min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{i=1}^{N}\alpha_i \\
s.t. \quad & \sum_{i=1}^{N}\alpha_i y_i = 0 \\
& 0 \le \alpha_i \le C
\end{aligned}
$$

After we obtain a satisfied α, we could construct a classifier:

$$
f(x) = sign(\sum_{i=1}^{N}\alpha_i y_i \langle x_i, x \rangle + b)
$$

Optimizing

Due to **KKT conditions** and further lemmas, we know that:

$$\alpha_i\big(y_i(w^T x_i + b) - 1 + \xi_i\big) = 0 \quad \beta_i \xi_i = 0$$
$$y_i(w^T x_i + b - 1 + \xi_i) \geq 0 \quad C - \beta_i - \alpha_i = 0$$

If a x_i, y_i pair satisfied constraints, it will follow:

$$\begin{aligned} \alpha_i = 0 \Rightarrow \qquad & \beta_i = C, \xi_i = 0 \Rightarrow y_i(w^T x_i + b) \geq 1 \\ 0 < \alpha_i < C \Rightarrow & 0 < \beta_i < C, \xi_i = 0 \Rightarrow y_i(w^T x_i + b) = 1 \\ \alpha_i = C \Rightarrow \qquad & \beta_i = 0, \xi_i \geq 0 \Rightarrow y_i(w^T x_i + b) \leq 1 \end{aligned}$$

In next slide, we will utilize this lemma to test if final result is converged.

What's SMO?

Sequential minimal optimization (SMO) is an algorithm for solving the quadratic programming (QP) problem that arises during the training of SVM. It was invented by John Platt in 1998 at Microsoft Research. In SVM, the QP problem is expressed as follow:

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{i=1}^{N}\alpha_i \tag{6}$$

$$s.t. \quad \sum_{i=1}^{N}\alpha_i y_i = 0 \tag{7}$$

$$0 \le \alpha_i \le C \tag{8}$$

What's SMO?

In constraint (7), lets say we select α_1, α_2 from α, then fix the rest of variables, which means treating them as a constant, then we could have:

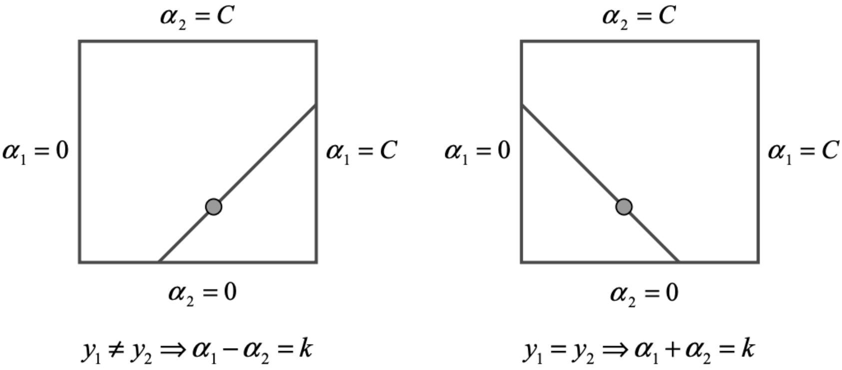
$$\alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{N} \alpha_i y_i = \zeta$$

So the QP problem seperate several subproblems, we assume $K = XX^T \in \mathbb{R}^{N \times N}$, and one of subproblems is shown as follow:

$$\begin{aligned}
\min_{\alpha_1, \alpha_2} \; W(\alpha_1, \alpha_2) = & \frac{1}{2} K_{11} {\alpha_1}^2 + \frac{1}{2} K_{22} {\alpha_2}^2 + y_1 y_2 K_{12} \alpha_1 \alpha_2 \\
& - (\alpha_1 + \alpha_2) + y_1 \alpha_1 \sum_{i=3}^{N} y_i \alpha_i K_{i1} + y_2 \alpha_2 \sum_{i=3}^{N} y_i \alpha_i K_{i2} \\
s.t. \; \alpha_1 y_1 + \alpha_2 y_2 = \zeta \qquad & 0 \leq \alpha_1 \leq C, \; i = 1, 2
\end{aligned}$$

Subproblem Optimizing

Due to the constraint, α_1, α_2 must lie within the box $[0, C] \times [0, C]$ as shown below:



Subproblem Optimizing

The following bounds apply to $\alpha_2 \in [L, H]$ is:

$$L = \begin{cases} \max(0, \alpha_2 + \alpha_1 - C) & y_1 = y_2 \\ \max(0, \alpha_2 - \alpha_1) & y_1 \neq y_2 \end{cases}$$

$$H = \begin{cases} \min(C, \alpha_2 + \alpha_1) & y_1 = y_2 \\ \min(C, \alpha_2 - \alpha_1 + C) & y_1 \neq y_2 \end{cases}$$

Subproblem Optimizing

Lets say:

$$E_i = \sum_{j=1}^{N} \alpha_i y_i \langle x_i, x_j \rangle + b - y_i$$

$$\eta = K_{11} + K_{22} - 2K_{12}$$

$$\alpha_2^{unclip} = \alpha_2 + \frac{y_2(E_1 - E_2)}{\eta}$$

The result need to be clipped:

$$\alpha_2^{new} = \begin{cases} H & \text{if } \alpha_2^{unclip} > H \\ L & \text{if } \alpha_2^{unclip} < L \\ \alpha_2^{unclip} & \text{otherwise} \end{cases}$$

After we obtain the α_2, we could get $\alpha_1^{new} = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{new})$.

Update other variables

For next iteration, we need to update b and E_i, For $j \in \{1, 2\}$.

$$\because \sum_{i=1}^{N} \alpha_i y_i K_{ij} + b_j = y_j$$

$$\therefore b_j^{new} = y_j - \sum_{i=3}^{N} \alpha_i y_i K_{ij} - \alpha_1^{new} K_{1j} - \alpha_2^{new} K_{2j}$$

$$b^{new} = \begin{cases} b_1^{new}, & \text{if } \alpha_1 \in (0, C) \\ b_2^{new}, & \text{if } \alpha_2 \in (0, C) \\ (b_1^{new} + b_2^{new})/2, & \text{otherwise} \end{cases}$$

$$E_j^{new} = \sum_{i \in S} y_i \alpha_i K_{ij} + b^{new} - y_i$$

S is the set of support vectors, which means $y_i(w^T x_i + b) \leq 1$.

Select α_1 and α_2

In most of full SMO algorithm implements, they are dedicated to heuristics to maximize the objective function as much as possible.
However, in practical we follow a simplified version. First we iterate α_i and check if it violates KKT conditions, then we select it as α_2 and we randomly choose α_1 from the remaining α_i and attempt to jointly optimize α_1, α_2.
Repeat this step on all α_i, and check if all α_i are obey the KKT conditions. If they are, we could yield the α, and construct the classifier:

$$f(x) = sign(\sum_{i \in S} \alpha_i y_i \langle x_i, x \rangle + b)$$

$$S = \{\alpha_i | \alpha_i \in \alpha, \alpha_i \neq 0\}$$

Gradient Optimization

In general, the primal problem could be written as:

$$\min_{w,b} \quad \mathcal{L}(w,b) = \sum_{i=1}^{N} \mathbb{1}\left[1 - y_i\big(w \cdot x_i + b\big)\right] + \lambda ||w||^2$$

Just take the derivatives:

$$\bigtriangledown_w \mathcal{L}(w,b) = -\sum_{i=1}^{N} \mathbb{1}\left[1 - y_i\big(w \cdot x_i + b\big)\right] y_i x_i + 2\lambda w$$

$$\bigtriangledown_b \mathcal{L}(w,b) = -\sum_{i=1}^{N} \mathbb{1}\left[1 - y_i\big(w \cdot x_i + b\big)\right] y_i$$

Therefore, we first initialize w, b, and set the learning rate η. Using the gradient optimization, the $\mathcal{L}(w,b)$ will converge at a minimal point.

Reference

Stanford CS229 class notes and section notes.

- Autumn 2009 The Simplified SMO Algorithm.
- Lecture Notes 3, Andrew Ng.

Paper

- Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines.
- C.-C. Chang and C.-J. Lin. LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011.

The End