Support Vector Machine (SVM) and Sequential Minimal Optimization (SMO)

Huayu Zhang

July 10, 2016

1 Support Vector Machines

1.1 A linear classifier

Support Vector Machine is proposed as a robust linear binary classifiers. Given a set of data $\{(\boldsymbol{x}_i, y) \mid i \in \{1, \dots, N\}\} \in \mathbb{R}^p \times \{-1, 1\}$, we want to find a linear classifier $y = \text{sign}(\boldsymbol{w}^T\boldsymbol{x} - b)$ such that

$$\boldsymbol{w}^T\boldsymbol{x}_i - b > 0 \quad \forall i \in \{i : y_i = +1\} \tag{1}$$

$$\boldsymbol{w}^T\boldsymbol{x}_i - b < 0 \quad \forall i \in \{i : y_i = -1\} \tag{2}$$

We can reformulate Eq (1) by scaling the weights $\boldsymbol{w}$ and offset b

$$\begin{aligned} \boldsymbol{w}^T\boldsymbol{x}_i - b \geq 1 \quad \forall i \in \{i : y_i = +1\} \\ \boldsymbol{w}^T\boldsymbol{x}_i - b \leq -1 \quad \forall i \in \{i : y_i = -1\} \end{aligned}$$

and simplify the problem

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i - b) \geq 1 \quad \forall i \tag{3}$$

There are a plethora of candidates $\boldsymbol{w}, b$. SVM tries to find one that maximizes the margin of two classes (i.e. the distance between hyperplane $\boldsymbol{w}^T\boldsymbol{x}_i - b = 1$ and $\boldsymbol{w}^T\boldsymbol{x}_i - b = -1$). The margin $h = \frac{2}{\|\boldsymbol{w}\|_2}$. Therefore SVM solves a convex optimization problem

$$\begin{aligned} &\min \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ &s.t.\ y_i(\boldsymbol{w}^T\boldsymbol{x}_i - b) \geq 1, \quad \forall i \end{aligned} \tag{4}$$

1.2 A linearly inseparable classifier

Sometimes the data is not linearly separable. For these cases, Vapnik [1] suggests a penalty for misclassification

$$\begin{aligned} &\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N} \xi_i \\ &s.t.\ y_i(\boldsymbol{w}^T\boldsymbol{x}_i - b) \geq 1 - \xi_i, \quad \forall i \\ &\boldsymbol{\xi} \geq 0 \end{aligned} \tag{5}$$

1.3 Generalized SVM

Linear SVM can be generalized to nonlinear classifiers.

$$\begin{aligned}
\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \ & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i \\
s.t. \ & y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b) \geq 1 - \xi_i, \quad \forall i \\
& \boldsymbol{\xi} \geq 0
\end{aligned} \tag{6}$$

where $\phi : \mathbb{R}^p \mapsto \mathbb{R}^r$.

EXAMPLE 1.

$$\begin{aligned}
\boldsymbol{x} =& (x_1, x_2) \\
\phi(\boldsymbol{x}) =& (x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1)
\end{aligned}$$

makes a quadratic classifier.

2 Dual problem

2.1 Dual problem

The Lagrangian of (6) is

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i(1 - \xi_i - y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b)) - \sum_{i=1}^{N}\lambda_i\xi_i \tag{7}$$

where $\alpha_i \geq 0, \lambda_i \geq 0$ for any $i \in \{1, \dots, N\}$. The dual function

$$\begin{aligned}
g(\boldsymbol{\alpha}, \boldsymbol{\lambda}) &= \inf_{\boldsymbol{w},b,\boldsymbol{\xi}} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\lambda}) \\
&= \inf_{\boldsymbol{w},b,\boldsymbol{\xi}} \ \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \sum_{i=1}^{N}\alpha_i y_i \phi(\boldsymbol{x}_i)^T\boldsymbol{w} + \sum_{i=1}^{N}(C - \alpha_i - \lambda_i)\xi_i + \sum_{i=1}^{N}\alpha_i + \sum_{i=1}^{N}\alpha_i y_i b \\
&= \begin{cases} \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)\alpha_i\alpha_j & \text{if } \alpha_i + \lambda_i \leq C, \ \sum_{i=1}^{N}\alpha_i y_i = 0 \\ -\infty & \text{otherwise} \end{cases}
\end{aligned}$$

The dual problem is

$$\begin{aligned}
\min_{\boldsymbol{\alpha},\boldsymbol{\lambda}} \ & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)\alpha_i\alpha_j - \sum_{i=1}^{N}\alpha_i \\
s.t. \ & \alpha_i \geq 0 \\
& \lambda_i \geq 0 \\
& \lambda_i + \alpha_i \leq C \\
& \sum_{i=1}^{N}\alpha_i y_i = 0
\end{aligned} \tag{8}$$

The $\boldsymbol{\lambda}$ can be eliminated

$$\begin{aligned}
\min_{\boldsymbol{\alpha}} \ & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)\alpha_i\alpha_j - \sum_{i=1}^{N}\alpha_i \\
s.t. \ & 0 \leq \alpha_i \leq C \\
& \sum_{i=1}^{N}\alpha_i y_i = 0
\end{aligned} \tag{9}$$

but I will derivate the KKT conditions based on (8).

In the primal problem (6), the objective function are convex on $\mathbb{R}^r \times \mathbb{R} \times \mathbb{R}^N$ and the constraints are linear. If there exists a feasible point $(\boldsymbol{w}, b, \boldsymbol{\xi})$, then the Slater's condition is satisfied and the optimal value of the primal problem is exactly that of the dual problem.

2.2 Kernel tricks

Define the kernel $K(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^T \phi(\boldsymbol{y})$.

EXAMPLE 2 (Quadratic kernel)**.** Consider the nonlinear function in Example 1,

$$\begin{aligned} K(\boldsymbol{x}, \boldsymbol{y}) =& \phi(\boldsymbol{x})^T \phi(\boldsymbol{y}) = x_1^2 y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 + 2x_1y_1 + 2x_2y_2 + 1 \\ =& (x_1y_1 + x_2y_2)^2 + 2(x_1y_1 + x_2y_2) + 1 \\ =& (x_1y_1 + x_2y_2 + 1) = (\boldsymbol{x}^T\boldsymbol{y} + 1)^2 \end{aligned}$$

which simplifies the computation.

Common kernels

1. linear $K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T\boldsymbol{y}$
2. polynomial $K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T\boldsymbol{y} + 1)^d$
3. Radial basis function (RBF) $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma\|\boldsymbol{x} - \boldsymbol{y}\|_2^2)$

2.3 KKT conditions

KKT conditions of problem (6) are

1. Primal feasibility: $y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b) \geq 1 - \xi_i, \xi_i \geq 0$
2. Dual feasibility: $\alpha_i \geq 0, \lambda_i \geq 0, \alpha_i + \lambda_i \leq C, \sum_{i=1}^N \alpha_i y_i = 0$
3. Complementary slackness: $\alpha_i(1 - \xi_i - y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b)) = 0, \lambda_i\xi_i = 0$
4. Gradient of Lagragian:$\lambda_i + \alpha_i = C, \sum_{i=1}^N \alpha_i y_i = 0, \boldsymbol{w} = \sum_{i=1}^N \alpha_i y_i \phi(\boldsymbol{x}_i)$

PROPOSITION 1. Let $u_i = \boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b$

$$\begin{aligned} \alpha_i = 0 \Rightarrow y_iu_i \geq 1 &\quad y_iu_i > 1 \Rightarrow \alpha_i = 0 \\ 0 < \alpha_i < C \Rightarrow y_iu_i = 1 &\quad y_iu_i = 1 \Rightarrow 0 \leq \alpha_i \leq C \\ \alpha_i = C \Rightarrow y_iu_i \leq 1 &\quad y_iu_i < 1 \Rightarrow \alpha = C \end{aligned}$$

Proof. "$\Rightarrow$"

1. If $\alpha_i = 0$, $\lambda_i = C \Rightarrow \xi_i = 0$. $y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b) \geq 1 - \xi_i = 1$ i.e. $y_iu_i \geq 1$.
2. If $0 < \alpha_i < C$, $y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b) = 1 - \xi_i$ and $0 < \lambda_i < C \Rightarrow \xi_i = 0$, so $y_iu_i = 1$.
3. If $\alpha_i = C$, $y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b) = 1 - \xi_i \leq 1$ i.e. $y_iu_i \leq 1$.

"$\Leftarrow$"

1. $y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b) > 1 \Rightarrow 1 - \xi_i - y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b) < 0 \Rightarrow \alpha_i = 0$
2. $y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b) = 1 \Rightarrow \xi_i = 0 \Rightarrow \alpha_i$ is unconstrained i.e. $0 \leq \alpha_i \leq C$.
3. $y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b) < 1 \Rightarrow \xi_i > 0 \Rightarrow \lambda_i = 0 \Rightarrow \alpha_i = C$.

$\square$

3 Sequential minimal optimization (SMO) [2]

The main challenge of solving the problem 9 (Why bother with the dual problem when fitting SVM?) is that we need $O(N^2)$ memory to store the dual function (standard formulation of a quadratic programming problem), especially when N goes extremely large.

The SMO updates two Lagrangian multipliers, denoted by α_1, α_2 each time. Fix $\alpha_i, i = 3, \ldots, N$,

$$y_1\alpha_1 + y_2\alpha_2 = C_0 \tag{10}$$

Let $s = y_1 y_2$, we have

$$\begin{aligned} \alpha_1 + \alpha_2 = C_1 \quad &\text{if } s = 1 \\ \alpha_1 - \alpha_2 = C_2 \quad &\text{if } s = -1 \end{aligned}$$

The domain of new α_2 is $[L, H]$ where

$$\begin{aligned} L = \max(0, \alpha_2 - \alpha_1), \quad H = \min(C, C + \alpha_2 - \alpha_1) \quad &\text{if } s = -1 \\ L = \max(0, \alpha_2 + \alpha_1 - C), \quad H = \min(C, \alpha_1 + \alpha_2) \quad &\text{if } s = 1 \end{aligned}$$

Let α_1^*, α_2^* be the values of α_1, α_2 in the last iteration,

$$\alpha_1 + s\alpha_2 = \alpha_1^* + s\alpha_2^* = y_1 C_0 =: t \tag{11}$$

First make some denotations,

1. $K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$
2. $v_i = \sum_{j=3}^N y_j K_{ij}\alpha_j = u_i + b^* - y_1\alpha_1^* K_{1i} - y_2\alpha_2^* K_{2i},\ i = 1, 2$

The dual function with respect to α_1, α_2

$$\begin{aligned} g(\alpha_1, \alpha_2) =& \frac{1}{2}(\alpha_1^2 K_{11} + \alpha_2^2 K_{22} + 2s\alpha_1\alpha_2 K_{12}) + \alpha_1 y_1 v_1 + \alpha_2 y_2 v_2 - \alpha_1 - \alpha_2 + Const \\ g(\alpha_2) =& \frac{1}{2}K_{11}(t - s\alpha_2)^2 + \frac{1}{2}K_{22}\alpha_2^2 + sK_{12}(t - s\alpha_2)\alpha_2 + y_1(t - s\alpha_2)v_1 + s\alpha_2 - t + y_2\alpha_2 v_2 - \alpha_2 + Const \end{aligned}$$

Take the first derivative of $g(\alpha_2)$ and let it equal to 0,

$$\frac{dg}{d\alpha_2} = -sK_{11}t + K_{11}\alpha_2 + K_{22}\alpha_2 - K_{12}\alpha_2 + sK_{12}t - K_{12}\alpha_2 - y_2(v_1 - v_2) + s - 1 = 0$$

we have

$$\begin{aligned} (K_{11} + K_{22} - 2K_{12})\alpha_2 =& s(K_{11} - K_{12})t + y_2(v_1 - v_2) + 1 - s \\ =& s(K_{11} - K_{12})(\alpha_1^* + s\alpha_2^*) + y_2(u_1 - u_2 - y_1\alpha_1^*(K_{11} - K_{12}) - y_2\alpha_2^*(K_{21} - K_{22})) + 1 - s \\ =& (K_{11} + K_{22} - 2K_{12})\alpha_2^* + y_2(u_1 - u_2 + y_2 - y_1) \overset{E_i = u_i - y_i}{=} (K_{11} + K_{22} - 2K_{12})\alpha_2^* + y_2(E_1 - E_2) \end{aligned}$$

so

$$\alpha_2^{\text{new}} = \alpha_2^* + \frac{y_2(E_1 - E_2)}{\eta} \tag{12}$$

where $\eta = K_{11} + K_{22} - 2K_{12}$ Consider the domain of α_2, if $\eta \geq 0$, the new value is

$$\alpha_2^{\text{new,clipped}} = \begin{cases} H & \text{if } \alpha_2^{\text{new}} \geq H \\ \alpha_2^{\text{new}} & \text{if } L < \alpha_2^{\text{new}} < H \\ L & \text{if } \alpha_2^{\text{new}} \leq L \end{cases} \tag{13}$$

$$\alpha_1^{\text{new}} = \alpha_1 + s(\alpha_2 - \alpha_2^{\text{new,clipped}}) \tag{14}$$

If $\eta < 0$, $\alpha_2^{\text{new,clipped}} = \underset{\alpha_2 \in \{L,H\}}{\arg\min}\ g(\alpha_2)$.

After updating $\boldsymbol{\alpha}$, b is updated

$$\begin{aligned} b_1 =& E_1 + y_1(\alpha_1^{\text{new}} - \alpha_1)K_{11} + y_2(\alpha_2^{\text{new, clipped}} - \alpha_2)K_{12} + b \\ b_2 =& E_2 + y_1(\alpha_1^{\text{new}} - \alpha_1)K_{12} + y_2(\alpha_2^{\text{new, clipped}} - \alpha_2)K_{22} + b \\ b =& (b_1 + b_2)/2 \end{aligned}$$

If the SVM is linear, we can also update the weights

$$\boldsymbol{w}^{\text{new}} = \boldsymbol{w} + y_1(\alpha_1^{\text{new}} - \alpha_1)\boldsymbol{x}_1 + y_2(\alpha_2^{\text{new,clipped}} - \alpha_2)\boldsymbol{x}_2 \tag{15}$$

The detailed algorithm is given in [2].

References

[1] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297.

[2] John Platt et al. "Sequential minimal optimization: A fast algorithm for training support vector machines". In: (1998).