The Simplified SMO Algorithm

This document describes a simplified version of the Sequential Minimal Optimization (SMO) algorithm for training Support Vector Machines.

- A Support Vector Machine computes a linear classifier of the form

$$f(x) = w^T x + b \qquad (1)$$

- We will ultimately predict $y = 1$ if $f(x) >= 0$ and $y = -1$ if $f(x) < 0$.

$$f(x) = \sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, x \rangle + b \qquad (2)$$

Where we can substitute a kernel $K(x^{(i)}, x)$ in place of the inner product.

- We want to solve:

$$max_\alpha \quad W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \qquad (3)$$

$$subject\ to \qquad 0 \leq \alpha_i \leq C, \qquad i = 1, \dots, m \qquad (4)$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \qquad (5)$$

- The KKT conditions can be used to check for convergence to the optimal point.

$$\alpha_i = 0 \ \rightarrow \ y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 \qquad (6)$$

$$\alpha_i = C \ \rightarrow \ y^{(i)}\left(w^T x^{(i)} + b\right) \leq 1 \qquad (7)$$

$$0 < \alpha_i < C \ \rightarrow \ y^{(i)}\left(w^T x^{(i)} + b\right) = 1 \qquad (8) \quad (9)$$

Any α_i's that satisfy these properties for all i will be an optimal solution to the optimization problem.

- The SMO algorithm selects two α parameters, α_i and α_j and optimizes the objective value jointly for both these α's. Finally it adjusts the b parameter based on the new α's.

Step 1: Selecting α Parameters

- We simply iterate over all $\alpha_i, i = 1, \dots, m$. If α_i does not fulfill the KKT conditions to within some numerical tolerance, we select α_j at random from the remaining m-1 α_i's, then the algorithm terminates.

Step 2: Optimizing $\boldsymbol{\alpha_i}$ and $\boldsymbol{\alpha_j}$

- Having chosen the Lagrange multipliers α_i and α_j to optimize, we first compute constraints on the values of these parameters, then we solve the constraint maximization problem.
- First we want to find bounds L and H such that $L \le \alpha_j \le H$ must hold in order for α_j to satisfy the constraint that $0 \le \alpha_j \le C$.

$$if\ y^{(i)} \neq y^{(j)}, \quad L = \max\left(0, \alpha_j - \alpha_i\right), \ \ H = \min\left(C, C + \alpha_j - \alpha_i\right) \quad (10)$$

$$if\ y^{(i)} = y^{(j)}, \ \ L = \max\left(0, \alpha_i + \alpha_j - C\right), \ \ H = \min\left(C, \alpha_i + \alpha_j\right) \quad (11)$$

- Now we want to find α_j so as to maximize the objective function. If this value ends up lying outside the bounds L and H, we simply clip the value of α_j to lie within this range.

$$\alpha_j \coloneqq \alpha_j - \frac{y^{(j)}\left(E_i - E_j\right)}{\eta} \quad (12)$$

Where

$$E_k = f\left(x^{(k)}\right) - y^{(k)} \quad (13)$$

$$\eta = 2\langle x^{(i)}, x^{(j)}\rangle - \langle x^{(i)}, x^{(i)}\rangle - \langle x^{(j)}, x^{(j)}\rangle \quad (14)$$

- Next we clip α_j to lie within the range [L, H]

$$\alpha_j \coloneqq \begin{cases} H & if\ \alpha_j > H \\ \alpha_j & if\ L \le \alpha_j \le H \\ L & if\ \alpha_j < L \end{cases} \quad (15)$$

- Finally, having solved for α_j we want to find the value for α_i.

$$\alpha_i \coloneqq \alpha_i + y^{(i)} y^{(j)} \left(\alpha_j^{(old)} - \alpha_j\right) \quad (16)$$

Where $\alpha_j^{(old)}$ is the value of α_j before optimization by (12) and (15).

Step 3: Computing the b threshold

- After optimizing α_i and α_j, we select the threshold b such that the KKT conditions are satisfied for the ith and jth examples. If, after optimization, α_i is not at the bounds (i.e., $0 < \alpha_i < C$), then the following threshold b_1 is valid:

$$b_1 = b - E_i - y^{(i)}\left(\alpha_i - \alpha_i^{(old)}\right)\langle x^{(i)}, x^{(i)}\rangle - y^{(j)}\left(\alpha_j - \alpha_j^{(old)}\right)\langle x^{(i)}, x^{(j)}\rangle \quad (17)$$

Similarly, the following threshold b_2 is valid if $0 < \alpha_j < C$

$$b_2 = b - E_j - y^{(i)}\left(\alpha_i - \alpha_i^{(old)}\right)\langle x^{(i)}, x^{(j)}\rangle - y^{(j)}\left(\alpha_j - \alpha_j^{(old)}\right)\langle x^{(j)}, x^{(j)}\rangle \quad (18)$$

If both $0 < \alpha_i < C$ and $0 < \alpha_j < C$ then both these thresholds and valid, and they will be equal. If both new α's are at the bounds, then all the thresholds between b_1 and b_2 satisfy the KKT conditions. We let $b \coloneqq (b_1 + b_2)/2$.

$$b \coloneqq \begin{cases} b_1 & if\ 0 < \alpha_i < C \\ b_2 & if\ 0 < \alpha_j < C \\ \dfrac{b_1 + b_2}{2} & otherwise \end{cases} \quad (19)$$

Pseudo-Code for Simplified SMO

Input:

C: regularization parameter

Tol: numerical tolerance

Max_passes: max # of times to iterate over α's without changing

$(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$: training data

Output:

$\alpha \in R^m$: Lagrange multipliers for solution

$b \in R$: threshold for solution

- Initialize $\alpha_i = 0, \forall i, \;\; b = 0.$
- Initialize $passes = 0.$
- While $(passes < \max\_passes)$
  - o $num\_changed\_alphas = 0.$
  - o For $i = 1, \dots, m,$
    - ▪ Calculate $E_i = f(x^{(i)}) - y^{(i)}$

$$= (\sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, x^{(i)} \rangle + b) - y^{(i)}$$

    - If $((y^{(i)} E_i < -tol \;\&\&\; \alpha_i < C) || (y^{(i)} E_i > tol \;\&\&\; \alpha_i > 0))$
      - o Select $j \neq i$ randomly.
      - o Calculate $E_i = f(x^{(i)}) - y^{(i)}$

$$= (\sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, x^{(i)} \rangle + b) - y^{(i)}$$

      - o Save old α's: $\alpha_i^{(old)} = \alpha_i, \alpha_j^{(old)} = \alpha_j$
      - o Compute L and H:

$$if\ y^{(i)} \neq y^{(j)},\ \ L = \max(0, \alpha_j - \alpha_i), H = \min(C, C + \alpha_j - \alpha_i)$$

$$if\ y^{(i)} \neq y^{(j)},\ \ L = \max(0, \alpha_i + \alpha_j - C), H = \min(C, \alpha_i + \alpha_j)$$

      - o If (L == H)

        Continue to next i.

- Compute η:

$$\eta = 2\langle x^{(i)}, x^{(j)}\rangle - \langle x^{(i)}, x^{(i)}\rangle - \langle x^{(j)}, x^{(j)}\rangle$$

- If ($\eta \geq 0$)

  Continue to next i.

- Compute and clip new value for α_j:

$$\alpha_j \coloneqq \alpha_j - \frac{y^{(j)}(E_i - E_j)}{\eta}$$

$$\alpha_j \coloneqq \begin{cases} H & if\ \alpha_j > H \\ \alpha_j & if\ L \leq \alpha_j \leq H \\ L & if\ \alpha_j < L \end{cases}$$

- If ($\left|\alpha_j - \alpha_j^{(old)}\right| < 10^{-5}$)

  Continue to next i.

- Determine value for α_i:

$$\alpha_i \coloneqq \alpha_i + y^{(i)}y^{(j)}(\alpha_j^{(old)} - \alpha_j)$$

- Compute b_1 and b_2:

$$b_1 = b - E_i - y^{(i)}\left(\alpha_i - \alpha_i^{(old)}\right)\langle x^{(i)}, x^{(i)}\rangle - y^{(j)}(\alpha_j - \alpha_j^{(old)})\langle x^{(i)}, x^{(j)}\rangle$$

$$b_2 = b - E_j - y^{(i)}\left(\alpha_i - \alpha_i^{(old)}\right)\langle x^{(i)}, x^{(j)}\rangle - y^{(j)}(\alpha_j - \alpha_j^{(old)})\langle x^{(j)}, x^{(j)}\rangle$$

- Compute b:

$$b \coloneqq \begin{cases} b_1 & if\ 0 < \alpha_i < C \\ b_2 & if\ 0 < \alpha_j < C \\ \frac{b_1 + b_2}{2} & otherwise \end{cases}$$

- $num\_changed\_alphas \coloneqq num\_changed\_alphas + 1$

* End if

End for

If ($num\_changed\_alphas == 0$)

$$passes \coloneqq passes + 1$$

Else

$$passes := 0$$

End while