Support Vector Machine

Wenfeng Luo

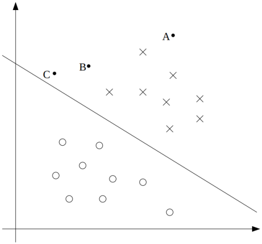
Sun Yat-sen University

12-22-2018

- Functional Margin
- Logistic Regression $P(y = 1|x; \theta) = \sigma(\theta^T x)$.
- If $\sigma(\theta^T x) > 0.5$, predict $y = 1$; otherwise, $y = 0$.
- The larger $\theta^T x$ is, the more confident our degree of "confidence" that the label is 1.

Margins: Intuition

- Geometric Margin
- Point A: far away from the decision boundary, we're very confident that A belongs to class $\times$.
- Point C: very close to the decision boundary, small changes to the separating hyper-plane will cause out C's prediction to be class $\circ$
- Point B: in-between case
- We want a decision boundary yielding **correct** and **confident** predictions.

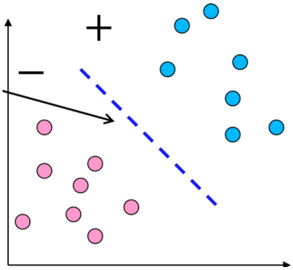


Notation

- features: $x \in \mathbb{R}^n$
- labels: $y \in \{-1, 1\}$
- training data: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(m)}, y^{(m)})\}$
- parameters: w, b
- classifier:

$$
\begin{aligned}
h_{w,b}(x) &= g(w^T x + b) \\
g(z) &= \begin{cases} 1 & z > 0 \\ -1 & z < 0 \end{cases}
\end{aligned} \tag{1}
$$

Geometric Margin

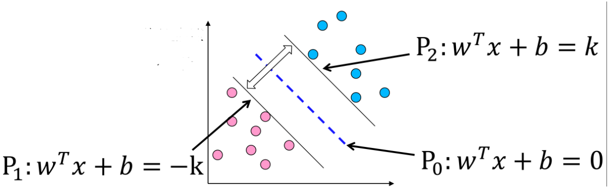


- Decision boundary

$$w^T x + b = 0 \tag{2}$$

- Classifier $f(x) = \text{sign}(w^T x + b)$

Maximal Margin



- Distance between P1 and P2

$$\text{margin} = \frac{k - (-k)}{\sqrt{w_1^2 + \ldots + w_n^2}} = \frac{2k}{||w||_2} \tag{3}$$

- Our goal is to maximize the margin.

Optimization Problem

- Optimization Problem 0

$$\begin{aligned} \text{argmax}_{w,b} \ & \frac{2k}{||w||_2} \\ \text{s.t.} \ & w^T x^{(i)} + b \geq k, \text{for } y^{(i)} = 1 \\ & w^T x^{(i)} + b \leq -k, \text{for } y^{(i)} = -1 \end{aligned} \tag{4}$$

- Optimization Problem 1

$$\begin{aligned} \text{argmax}_{w,b} \ & \frac{2k}{||w||_2} \\ \text{s.t.} \ & y^{(i)}(w^T x^{(i)} + b) \geq k \end{aligned} \tag{5}$$

Optimization Problem

- Optimization Problem 2

$$\begin{aligned} &\operatorname{argmax}_{w,b} \frac{2}{||\frac{w}{k}||_2} \\ &\text{s.t. } y^{(i)}\left(\left(\frac{w}{k}\right)^T x^{(i)} + \frac{b}{k}\right) \geq 1 \end{aligned} \tag{6}$$

- Optimization Problem 3($w' = \frac{w}{k}, b' = \frac{b}{k}$)

$$\begin{aligned} &\operatorname{argmax}_{w',b'} \frac{2}{||w'||_2} \\ &\text{s.t. } y^{(i)}\left(w'^T x^{(i)} + b'\right) \geq 1 \end{aligned} \tag{7}$$

Optimization Problem

- Optimization Problem 4

$$\begin{aligned} &\text{argmax}_{w,b} \ \frac{2}{||w||_2} \\ &\text{s.t.} \ y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 \end{aligned} \tag{8}$$

- Optimization Problem 5

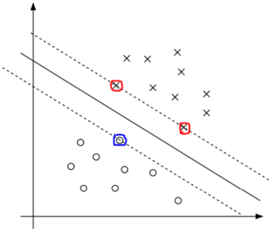
$$\begin{aligned} &\text{argmin}_{w,b} \ \frac{1}{2}||w||_2 \\ &\text{s.t.} \ y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 \end{aligned} \tag{9}$$

Optimization Problem

$$
\begin{aligned}
\operatorname{argmin}_{w,b}\ & \frac{1}{2} w^T w \\
\text{s.t. } & y^{(i)} \left(w^T x^{(i)} + b \right) \geq 1 \\
& i = 1, \ldots, m
\end{aligned} \tag{10}
$$

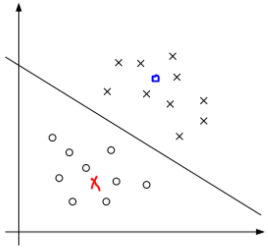
- Convex optimization problem
- Okay if the data is linearly separable

What are the Support Vectors?

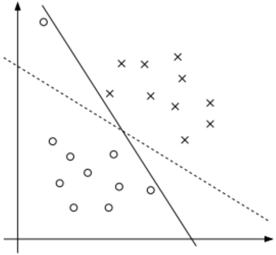


- Support Vectors are those data points that are closest to the decision boundary
- Number of support vectors could be much smaller than the size of the training set

- What if the data is not linearly separable?
- How to deal with outliers?



(a) Linearly non-separable



(b) Outliers

Regularization and the non-separable case

$$
\begin{aligned}
\operatorname{argmin}_{w,b} \ & \frac{1}{2} w^T w \\
\text{s.t. } & y^{(i)} \left(w^T x^{(i)} + b \right) \geq 1 \\
& i = 1, ..., m
\end{aligned}
\tag{11}
$$

- Allow the constrain of some data points not strictly satisfied
- And add in a penalization term(l_1 norm)
- Optimization problem 6

$$
\begin{aligned}
\operatorname{argmin}_{w,b,\xi} \ & \frac{1}{2} w^T w + C \sum_{i}^{m} \xi_i \\
\text{s.t. } & y^{(i)} \left(w^T x^{(i)} + b \right) \geq 1 - \xi_i, i = 1, ..., m \\
& \xi_i \geq 0, i = 1, ..., m
\end{aligned}
\tag{12}
$$

Optimization Problem

$$
\begin{aligned}
\operatorname{argmin}_{w,b,\xi} \ & \frac{1}{2} w^T w + C\mathbf{1}^T \xi \\
\text{s.t. } & y^{(i)} \left(w^T x^{(i)} + b \right) \geq 1 - \xi_i, i = 1, ..., m \\
& \xi_i \geq 0, i = 1, ..., m
\end{aligned} \tag{13}
$$

- C is a constant, just like the regularization hyper-parameter.
- How to find the optimal solution?
- **Lagrange multiplier** and **Coordinate descent**.

Lagrange duality

- Consider the problem

$$\begin{aligned}\min_{w} \quad & f(w) \\ \text{s.t.} \quad & h_i(w) = 0, i = 1, ..., m\end{aligned}$$

- Define **Lagrangian** to be

$$\mathcal{L}(w, \beta) = f(w) + \sum_{i}^{m} \beta_i h_i(w)$$

- β_i's are the Lagrange multipliers
- Compute the partial derivative and set them to 0

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \\ \frac{\partial \mathcal{L}}{\partial \beta_i} = 0, i = 1, ..., m \end{cases}$$

Lagrange duality

- More generally, there are both equality and inequality constrains
- Consider the following optimization problem

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

- Define the **generalized Lagrangian**

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

$$\alpha_i \geq 0, i = 1, ..., m$$

$$\max_{\alpha,\beta} \mathcal{L} = \begin{cases} f(w) & \text{constrain satisfied} \\ \infty & \text{otherwise} \end{cases}$$

Lagrange Duality

- **Primal** Problem

$$\min_{x} \max_{\alpha,\beta} \mathcal{L}$$

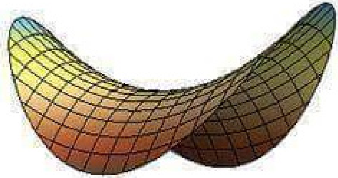
- **Dual** Problem

$$\max_{\alpha,\beta} \min_{x} \mathcal{L}$$

- This two problems are equal

$$\min_{x} \max_{\alpha,\beta} \mathcal{L} = \max_{\alpha,\beta} \min_{x} \mathcal{L} \tag{14}$$

Saddle-point Interpretation



$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = cz$$

Optimization Problem

- Optimization Problem 5

$$
\begin{aligned}
&\operatorname{argmin}_{w,b,\xi} \; \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i \\
&\text{s.t. } y^{(i)} \left(w^T x^{(i)} + b \right) \geq 1 - \xi_i, i = 1, \ldots, m \\
&\xi_i \geq 0, i = 1, \ldots, m
\end{aligned}
\tag{15}
$$

- Optimization Problem 6

$$
\begin{aligned}
&\operatorname{argmax}_{\alpha} \; \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \\
&\text{s.t. } 0 \leq \alpha_i \leq C, i = 1, \ldots, m \\
&\sum_{i=1}^{m} \alpha_i y^{(i)} = 0
\end{aligned}
\tag{16}
$$

Optimization Problem 5 $\to$ 6

- Constrain $y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1 - \xi_i$ and $\xi_i \geq 0$
- Constrain $-(y^{(i)}\left(w^T x^{(i)} + b\right) - 1 + \xi_i) \leq 0$ and $-\xi_i \leq 0$

$$
\begin{aligned}
L(w, b, \xi) = &\frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i \\
&- \sum_{i=1}^{m} \alpha_i \left(y^{(i)}(w^T x^{(i)} + b) - 1 + \xi_i \right) - \sum_{i=1}^{m} \beta_i \xi_i
\end{aligned}
$$

$$
\frac{\partial L}{\partial w} = w - \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}
$$

$$
\frac{\partial L}{\partial b} = -\sum_{i=1}^{m} \alpha_i y^{(i)}
$$

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

Optimization Problem 5 $\rightarrow$ 6

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \tag{17}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \tag{18}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \rightarrow \beta_i = C - \alpha_i (0 \leq \alpha_i \leq C) \tag{19}$$

Optimization Problem 5 $\rightarrow$ 6

- Use $w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}, \beta_i = C - \alpha_i$, substitude back to $\mathcal{L}$

$$
\begin{aligned}
L(w, b, \xi) =& \frac{1}{2} w^T w - \sum_{i=1}^{m} \alpha_i y^{(i)} w^T x^{(i)} - b \sum_{i=1}^{m} \alpha_i y^{(i)} + \\
& (C - \alpha_i - \beta_i) \sum_{i=1}^{m} \xi_i + \sum_{i=1}^{m} \alpha_i \\
=& \frac{1}{2} \left(\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \right)^T \left(\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \right) - \\
& \sum_{i=1}^{m} \alpha_i y^{(i)} \left(\sum_{j=1}^{m} \alpha_i y^{(j)} x^{(j)} \right)^T x^{(i)} + \sum_{i=1}^{m} \alpha_i \\
=& \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} \langle x^{(i)}, x^{(j)} \rangle
\end{aligned}
$$

Optimization Problem 6

$$\text{argmax}_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

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

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

- The Primal problem still belongs to semi-programming, no better than optimization 5
- Only inner product of input feature matters

KKT Conditions

- $\alpha_i = 0, y^{(i)}(w^T x^{(i)} + b) \geq 1$: $x^{(i)}$, outside the margin;
- $0 < \alpha < C, y^{(i)}(w^T x^{(i)} + b) = 1$: $x^{(i)}$ on the margin;
- $\alpha_i = C, y^{(i)}(w^T x^{(i)} + b) \leq 1$: $x^{(i)}$ inside margin.
- After optimizing over α's, recover the original model parameter w, b

$$
\begin{aligned}
w &= \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \\
b &= -\frac{1}{2}\left(\min_{y^{(i)}=1} w^T x^{(i)} + \max_{y^{(i)}=-1} w^T x^{(i)}\right)
\end{aligned}
\tag{20}
$$

- A new data point x

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

Kernel Trick

- Feature mapping ϕ, say

$$\phi(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$$

- For specific mapping ϕ, define the corresponding **Kernel** to be

$$K(x^{(i)}, x^{(j)}) = \langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle$$

- We want to use the feature $\phi(x)$, non-linearity
- Simply replace $\langle x^{(i)}, x^{(j)} \rangle$ with $K(x^{(i)}, x^{(j)})$
- Interestingly, $K(x^{(i)}, x^{(j)})$ could be efficiently computed without having to go through the actual feature mapping $\phi(x)$.

Kernel Trick

- Compute $K(x, z) = (x^T z)^2$, $O(n)$
- Write $K(x, z)$ differently

$$
\begin{aligned}
K(x, z) &= \left(\sum_{i=1}^{m} x_i z_i \right)^T \left(\sum_{i=1}^{m} x_i z_i \right) \\
&= \sum_{i=1}^{m} \sum_{j=1}^{m} x_i x_j z_i z_j \\
&= \sum_{i,j=1}^{m} (x_i x_j)(z_i z_j)
\end{aligned}
$$

Kernel Trick

- Compute $K(x,z) = (x^T z)^2$, $O(n)$
- Write $K(x,z)$ differently $K(x,z) = \sum_{i,j=1}^{m} (x_i x_j)(z_i z_j)$
- The corresponding feature mapping is

$$\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{bmatrix}$$

- Compute $K(x,z) = \phi(x)^T \phi(z)$, $O(n^2)$

- A related kernel

$$K(x,z) = (x^T z + c)^2 = \sum_{i,j=1}^{n}(x_i x_j)(z_i z_j) + \sum_{i=1}^{n}(\sqrt{2c}x_i)(\sqrt{2c}z_i) + c^2$$

- The corresponding feature mapping is

$$\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \\ \sqrt{2c}x_1 \\ \sqrt{2c}x_2 \\ \sqrt{2c}x_3 \\ c \end{bmatrix}$$

Common Kernel Functions

- Linear kernel $K(x, z) = x^T z$
- Polynomial Kernel $K(x, z) = (x^T z + c)^d$
- Gaussian Kernel (RBF) $K(x, z) = \exp\{-\frac{||x-z||^2}{2\sigma^2}\}$, infinite dimension.
- Sigmoid, ...
- The point is if you could prove there exists a feature mapping $\phi(x)$ such that $K(x, z) = \phi(x)^T \phi(z)$, then it's a valid kernel.

Kernel Trick

We only need the output $K(x, z)$ and we don't have to go through the procedure

$$(x, z) \to (\phi(x), \phi(z)) \to \phi(x)^T \phi(z)$$

Implementation from Sklearn

- from sklearn import svm

```
svm.SVC()
```

```
Init signature: svm.SVC(C=1.0, kernel='rbf', degree=3, gamma='auto', coef0=0.0,
 shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None
, verbose=False, max_iter=-1, decision_function_shape='ovr', random_state=None)
Docstring:
C-Support Vector Classification.
```

The SMO algorithm

- Coordinate Descent

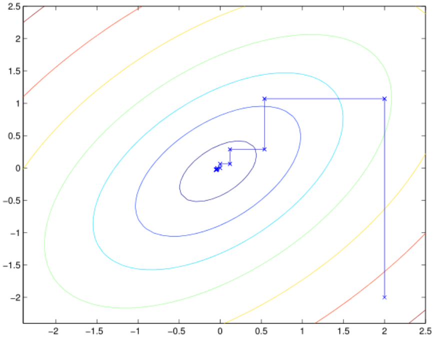
Loop until convergence: {

For $i = 1, \ldots, m$, {

$\alpha_i := \arg\max_{\hat{\alpha}_i} W(\alpha_1, \ldots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \ldots, \alpha_m)$.

}

}



The SMO algorithm

$$\begin{aligned} \text{argmax}_{\alpha} \ & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \\ \text{s.t. } & 0 \leq \alpha_i \leq C, i = 1, ..., m \\ & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \end{aligned} \tag{21}$$

- Maximize on α_1, fix the other $m - 1$ α's?

$$\alpha_1 = -y^{(1)} \sum_{i=2}^{m} \alpha_i y^{(i)} \tag{22}$$

- If $\alpha_2, ..., \alpha_m$ are fixed, then α_1 is also a constant, can't do any better.

The SMO algorithm

$$\text{argmax}_\alpha \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)}, x^{(j)}\rangle$$
$$\text{s.t. } 0 \le \alpha_i \le C, i = 1, ..., m \tag{23}$$
$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

- Choose two α's, say α_1, α_2

$$\alpha_1 y^{(1)} + \alpha_2 y^{(2)} = -\sum_{i=3}^{m} \alpha_i y^{(i)} \tag{24}$$

- Denote $\zeta = -\sum_{i=3}^{m} \alpha_i y^{(i)}$

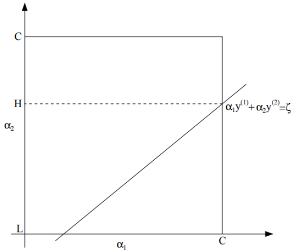
$$\alpha_1 y^{(1)} + \alpha_2 y^{(2)} = \zeta \tag{25}$$

The SMO algorithm

$$\begin{aligned} \text{argmax}_\alpha \; & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \\ \text{s.t. } & 0 \leq \alpha_i \leq C, i = 1, ..., m \\ & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \end{aligned} \tag{26}$$

- $\alpha_1 = y^{(1)}(\zeta - \alpha_2 y^{(2)})$
- Optimization sub-problem(fix $\alpha_3, ..., \alpha_m$, replace α_1 with α_2)

$$\begin{aligned} \text{argmax}_\alpha \; & A\alpha_2^2 + B\alpha_2 \\ \text{s.t. } & 0 \leq \alpha_1, \alpha_2 \leq C \end{aligned} \tag{27}$$



- Both α_1 and α_2 have to satisfy the box constrain.
- Consider only the constrain on $L \leq \alpha_2 \leq H$
- If $y^{(1)} y^{(2)} = 1$

$$\begin{cases} L = \max(0, \alpha_1^{old} + \alpha_2^{old} - C) \\ H = \min(C, \alpha_1^{old} + \alpha_2^{old}) \end{cases} \tag{28}$$

- Else

$$\begin{cases} L = \max(0, \alpha_2^{old} - \alpha_1^{old}) \\ H = \min(C, C + \alpha_2^{old} - \alpha_1^{old}) \end{cases} \tag{29}$$

$y^{(1)}y^{(2)} = 1$

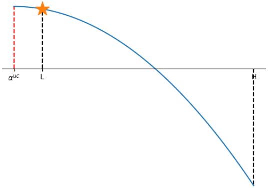
- Both α_1 and α_2 are in the line

$$\alpha_1 y^{(1)} + \alpha_2 y^{(2)} = \zeta = \alpha_1^{old} y^{(1)} + \alpha_2^{old} y^{(2)}$$

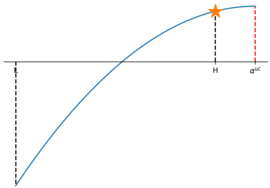
- Simply just $\alpha_1 + \alpha_2 = \alpha_1^{old} + \alpha_2^{old}$

$$\begin{cases} L = \max(0, \alpha_1^{old} + \alpha_2^{old} - C) \\ H = \min(C, \alpha_1^{old} + \alpha_2^{old}) \end{cases} \tag{30}$$

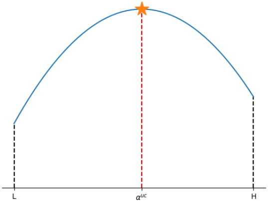
- $A \sim 2\langle x_1, x_2\rangle - \langle x_1, x_1\rangle - \langle x_2, x_2\rangle = -||x_1 - x_2||^2 \leq 0$, so



(1) $\alpha_2^{uc} < L$



(2) $\alpha_2^{uc} > H$



(3) $L \leq \alpha_2^{uc} \leq H$

$$\begin{aligned} \text{argmax}_{\alpha_2} \; & A\alpha_2^2 + B\alpha_2 \\ \text{s.t. } & L \leq \alpha_2 \leq H \end{aligned} \tag{31}$$

- Denote $\alpha_2^{uc} = -\frac{B}{2A}$, where the superscript stands for "unclipping".

$$\alpha_2^* = \begin{cases} L, & \alpha_2^{uc} < L \\ H, & \alpha_2^{uc} > H \\ \alpha_2^{uc}, & \textit{otherwise} \end{cases} \tag{32}$$

- Or put it simple

$$\alpha_2^* = \min\left(\max\left(L, \alpha_2^{uc}\right), H\right)$$
$$\alpha_1^* = y^{(1)}(\zeta - y^{(2)}\alpha_2^*)$$

Algorithm 1 Simple Implementation of SMO

Input: $\{(x^{(1)}, y^{(1)}), ...(x^{(m)}, y^{(m)})\}$
Output: w, b

Initialize all α's to 0
tol=10, iter=0
while iter $<$ tol **do**
    iter $=$ iter $+$ 1
    for $i \leftarrow 1$ to m **do**
        if KKT condition not satisfied for α_i **then**
            Randomly choose another α_j
            Compute the bound L and H for α_j
            if $L = H$ **then**
                continue
            end if
            iter $= 0$
            Compute the unclipping value α_j^{uc} for α_j
            $\zeta = \alpha_i^{old} y^{(i)} + \alpha_j^{old} y^{(j)}$
            $\alpha_j^{new} = \min(\max(L, \alpha_j^{uc}), H)$
            $\alpha_i^{new} = y^{(i)}(\zeta - \alpha_j^{new} y^{(j)})$
        end if
    end for
end while
Use equation (20) to Compute w, b from α's

One Possible Implementation

- svm.py: Kernel functions, SVM class.
- svm_solver.py: Optimization Module, WSS1Solver and WSS3Solver.

Conclusion

- Geometric marginal
- A series of equivalent optimization problems
- Kernel Functions
- Lagrange Duality and Coordinate Descent

References

- CS229 class notes

Thanks