BITS F464: Machine Learning

25 Support Vector Machine (SVM)

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

March 22, 2021 ONLINE (Campus @ BITS-Pilani Jan-May 2021)

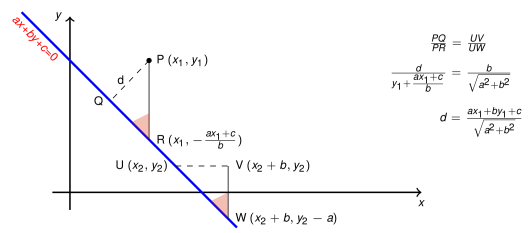
http://ktiwari.in/ml

Geometry

Determine the length of the perpendicular drawn on a line $ax + by + c = 0$ from a point (x_1, y_1)

$$d = \frac{|ax_1 + by_1 + c|}{\sqrt{a^2 + b^2}}$$

- Proof: (by geometry)



Geometry

- Essentially; distance of a point $X = (x_1, x_2, ..., x_n)$ from a hyperplane represented by $(b, w_1, w_2, ..., w_n)$ is given by

$$\frac{W^T X + b}{||W||}$$

where $||W||$ is norm [1]

Classification

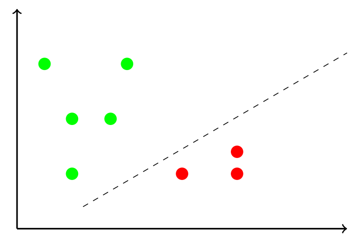
- Hyperplane have two sides (say +ve and -ve)
- which side a point $X = (x_1, x_2, ..., x_n)$ lies?

Substitute coordinates in the equation $W^T X + b$ and check the sign

[1] $W = (w_1, w_2, ..., w_n)$, and $||W|| = \sqrt{w_1^2 + w_2^2 + ... + w_n^2}$

Special Case of Classification

Consider a linearly separable dataset



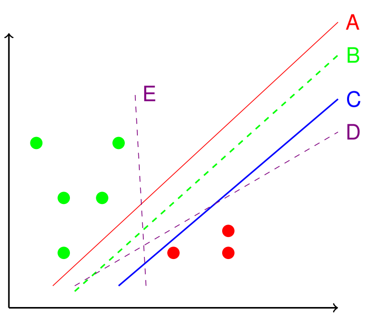
- Hyperplane is defined by W
- **Margin** is the distance of nearest data point from the separating hyperplane

$$\min_{x_i \in D} \frac{|W^T x_i + b|}{||W||}$$

- Most of the real-world problems are NOT linearly separable
- Sometime data could be transformed to a high-dimensional space where classes may be linearly separable
- Caution: this could lead to over-fitting

Which decision boundary is better

Many decision boundaries with perfect classification are possible



- Which of them are better?
- One that maximizes the margin

$$\underset{W,b}{\text{argmax}} \left[\min_{x_i \in D} \frac{|W^T x_i + b|}{||W||} \right]$$

- Fat **margin** is better

Regularize using classification accuracy, otherwise, it makes no sense.

Note: The separating hyperplane would be in middle

Look Closer

$$\underset{W,b}{\text{argmax}} \left[\min_{x_i \in D} \frac{|W^T x_i + b|}{||W||} \right] = \underset{W,b}{\text{argmax}} \frac{1}{||W||} \left[\min_{x_i \in D} |W^T x_i + b| \right] = \underset{W,b}{\text{argmax}} \frac{1}{||W||}$$

- Hypothesis is a hyperplane represented by W; scaled parameter $k.W$ also represents the same hyperplane
- For the points on hyperplane $W^T x_i + b = 0$
- For points NOT on hyperplane $W^T x_i + b = 0$ changes if $k.W$ is used instead of W. Ultimately different $\min_{x_i \in D} |W^T x_i + b|$
- You can get any value for $\min_{x_i \in D} |W^T x_i + b|$ by changing the value of k without changing the hyperplane

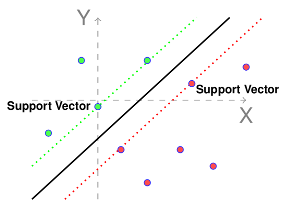
So without loss of generality, let us fix $\min_{x_i \in D} |W^T x_i + b| = 1$

Support Vector Machine (SVM)

SVM is a liner decision machine; uses $\text{sign}(w^T x^{(i)} + b)$ for decision

- We want $w^T x^{(i)} + b \geq \gamma$ for $+$ve (and $< -\gamma$ for $-$ve)
- Distance of a point (x, y) from hyper plane $w^T x + b = 0$ is $\frac{|w^T x + b|}{||w||}$
- Distance can be maximized, by either **maximizing** b or by **minimizing** $||w||$

- We needs $w^T x + b \geq \gamma ||w||$ let $\gamma ||w|| = 1$
  - $w^T x + b \geq 1$ if x is $+1$
  - $w^T x + b \leq -1$ if x is -1
- It leads to $\boxed{y^{(i)}(w^T x^{(i)} + b) \geq 1}$
- Points with $y^{(i)}(w^T x^{(i)} + b) = 1$ are called **support vector**



Support Vector Machine (SVM)

- Minimization of w is same as minimization of $\Phi(w) = \frac{1}{2} w.w$
- Other constraints are $y^{(i)}(w^T x^{(i)} + b) \geq 1$
- However, for support vectors $y^{(i)}(w^T x^{(i)} + b) = 1$
- Define a Lagrangian Multiplier to optimize $L(w, b) = \frac{1}{2} w.w - \sum \alpha_i (y^{(i)}(w^T x^{(i)} + b) - 1)$
- Derivative $\frac{\partial L}{\partial b} = -\sum \alpha_i y^{(i)}$ that should be equated to zero

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

- Derivative $\frac{\partial L}{\partial w} = w - \sum \alpha_i y^{(i)} x^{(i)}$ equated to zero gives

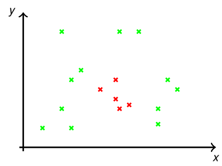
$$w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$$

Lagrangian with optimized values

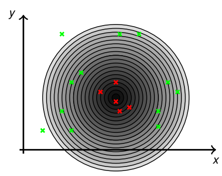
$$L(w, b) = \tfrac{1}{2} w.w - \sum \alpha_i (y^{(i)}(wx^{(i)} + b) - 1)$$
$$L(w, b) = \tfrac{1}{2} w.w - \sum \alpha_i y^{(i)} wx^{(i)} - \sum \alpha_i y^{(i)} b + \sum \alpha_i$$
$$L(w, b) = \tfrac{1}{2} w.w - \sum \alpha_i y^{(i)} wx^{(i)} + \sum \alpha_i$$
$$L(w, b) = \tfrac{1}{2} \sum \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)} x^{(j)} - \sum \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)} x^{(j)} + \sum \alpha_i$$
$$L(w, b) = \sum \alpha_i - \tfrac{1}{2} \sum \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)} x^{(j)}$$

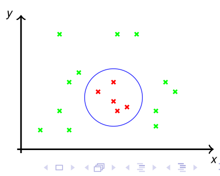
- One have to maximize $L(w, b)$ subject to $\alpha_i \geq 0$ and $\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$
- If α_i is large the corresponding training sample is support vector. Otherwise, when $\alpha_i = 0$ it is not a support vector
- Optimized α_i provides the value of $w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$ to be used in linear decision boundary

Transformation: An Example



- Transform all 2D points $(x^{(i)}, y^{(i)})$ in 3D as $(x^{(i)}, y^{(i)}, z)$ where $z = (x^{(i)} - x_c)^2 + (y^{(i)} - y_c)^2$





Support Vector Machine (SVM)

Advantages

- It works well with clear margin of separation
- Do not stuck to local minima
- Effective in high dimensional spaces
- It is effective in cases where number of dimensions is greater than the number of samples
- Memory efficient as it uses a subset of training points in decision (called support vectors)

Disadvantages

- It doesn't perform well, when data set is large as the training time is higher
- Doesn't perform well when target classes are overlapping
- Probability estimates are not directly provide
- Higher classification confidence for points lying far from the decision boundary

Thank You!

Thank you very much for your attention!

Queries ?

(Reference[2])

[2] [1] Text Book: *Machine Learning*, by *Tom M Mitchell* [2] Mod-01 Lec-29 Support Vector Machine https://www.youtube.com/watch?v=SRVswRH5Q7E