

Supper Vector Machines

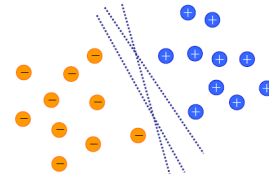
Chen Yu
Indiana University

Adapted from the slides by Martin Law

Linear Classifier

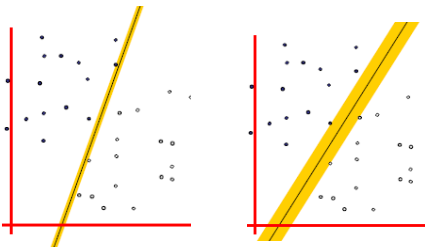
- How to classify the data? Which is the best?

$$f(\mathbf{x}, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$$



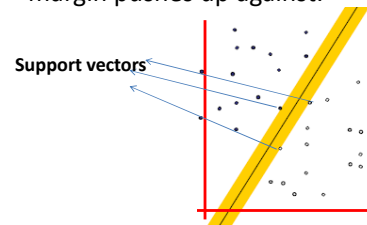
Maximum Margin

- Margin: the width that the boundary could be increased by before hitting a data point.



Support Vectors

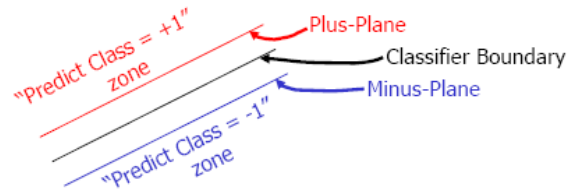
- Maximum margin linear classifier
- Support vectors are those datapoints that the margin pushes up against.



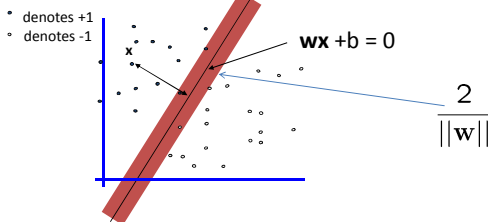
Why Maximum Margin

- It feels safe.
- Empirically it works well.
- If the location of the boundary is not perfect due to noise, this gives us the least chance of misclassification.
- Not sensitive to removal of any non support vector datapoints.
- Some theory called VC dimension.

Specifying a line and margin



Estimate the Margin



- What is the distance expression for a point \mathbf{x} to a line $\mathbf{w}\mathbf{x}+b=0$?

$$d(\mathbf{x}) = \frac{|\mathbf{x} \cdot \mathbf{w} + b|}{\sqrt{\|\mathbf{w}\|_2^2}} = \frac{|\mathbf{x} \cdot \mathbf{w} + b|}{\sqrt{\sum_{i=1}^d w_i^2}}$$

- Let $\{x_1, \dots, x_n\}$ be our data set and let $y_i \in \{1, -1\}$ be the class label of x_i .
- The decision boundary should classify all points correctly $\Rightarrow y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad \forall i$
- The decision boundary can be found by solving the following constrained optimization problem

$$\begin{aligned} &\text{Minimize } \frac{1}{2} \|\mathbf{w}\|^2 \\ &\text{subject to } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \forall i \end{aligned}$$

- This is a constrained optimization problem.

The Primal problem

Minimize $\frac{1}{2}\|\mathbf{w}\|^2$
 subject to $1 - y_i(\mathbf{w}^T \mathbf{x}_i + b) \leq 0$ for $i = 1, \dots, n$

- The Lagrange function is

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^T \mathbf{w} + \sum_{i=1}^n \alpha_i (1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))$$

– Note that $\|\mathbf{w}\|^2 = \mathbf{w}^T \mathbf{w}$

- Setting the gradient of \mathcal{L} w.r.t. \mathbf{w} and b to zero, we have

$$\mathbf{w} + \sum_{i=1}^n \alpha_i (-y_i) \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$$

$$\sum_{i=1}^n \alpha_i y_i = 0$$

- If we substitute $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$ to \mathcal{L} , we have

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^T \mathbf{w} + \sum_{i=1}^n \alpha_i (1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))$$

$$\begin{aligned} \mathcal{L} &= \frac{1}{2} \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i^T \sum_{j=1}^n \alpha_j y_j \mathbf{x}_j + \sum_{i=1}^n \alpha_i \left(1 - y_i \left(\sum_{j=1}^n \alpha_j y_j \mathbf{x}_j^T \mathbf{x}_i + b \right) \right) \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \alpha_i y_i \sum_{j=1}^n \alpha_j y_j \mathbf{x}_j^T \mathbf{x}_i - b \sum_{i=1}^n \alpha_i y_i \\ &= -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^n \alpha_i \end{aligned}$$

- The constraint $\sum_{i=1}^n \alpha_i y_i = 0$
- This is a function of α_i only

The Dual Problem

- The new objective function is in terms of α_i only
- It is known as the dual problem: if the primal problem has an optimal solution, the dual problem also has an optimal solution, and the corresponding optimal values are the **same**.
- The objective function of the dual problem needs to be maximized!

$$\max. W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

- With the constraints

$$\text{subject to } \alpha_i \geq 0, \quad \sum_{i=1}^n \alpha_i y_i = 0$$

$$\max. W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } \alpha_i \geq 0, \quad \sum_{i=1}^n \alpha_i y_i = 0$$

- This is a quadratic programming (QP) problem
 - A global maximum of α_i can always be found
- \mathbf{w} can be recovered by $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$
- KKT condition:

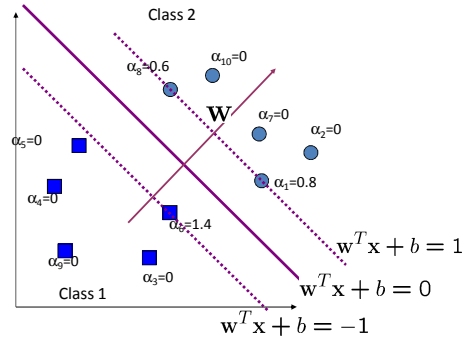
$$\alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1] = 0 \text{ for all } i$$

\mathbf{x}_i with $\alpha_i > 0$ are support vectors

Solution

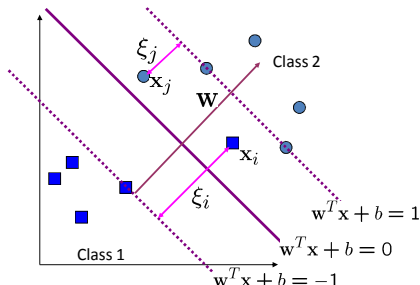
- Many of the α_i are zero
 - \mathbf{w} is a linear combination of a small number of data points
- \mathbf{x}_i with non-zero α_i are called support vectors (SVs)
 - The decision boundary is determined only by the SVs
 - Let t_j ($j=1, \dots, s$) be the indices of the s support vectors. We can write $\mathbf{w} = \sum_{j=1}^s \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$
- For testing with a new data \mathbf{z}
 - Compute $\mathbf{w}^T \mathbf{z} + b = \sum_{j=1}^s \alpha_{t_j} y_{t_j} (\mathbf{x}_{t_j}^T \mathbf{z}) + b$ and classify \mathbf{z} as class 1 if the sum is positive, and class 2 otherwise
 - Note: \mathbf{w} need not be formed explicitly

Example



Non-Separable Case

- We allow “error” ξ_i in classification; it is based on the output of the discriminant function $\mathbf{w}^T \mathbf{x} + b$



A new optimization problem

- If we minimize $\sum_i \xi_i$, ξ_i can be computed by

$$\begin{cases} \mathbf{w}^T \mathbf{x}_i + b \geq 1 - \xi_i & y_i = 1 \\ \mathbf{w}^T \mathbf{x}_i + b \leq -1 + \xi_i & y_i = -1 \\ \xi_i \geq 0 & \forall i \end{cases}$$
 - ξ_i are “slack variables” in optimization
 - Note that $\xi_i=0$ if there is no error for \mathbf{x}_i
- We want to minimize $\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i$
 - C : tradeoff parameter between error and margin
- The optimization problem becomes

$$\begin{aligned} &\text{Minimize } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \\ &\text{subject to } y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned}$$

- The dual of this new constrained optimization problem is

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$
 subject to $C \geq \alpha_i \geq 0, \sum_{i=1}^n \alpha_i y_i = 0$
- Once again, a QP solver can be used to find α_i
- \mathbf{w} is recovered as $\mathbf{w} = \sum_{j=1}^s \alpha_j y_j \mathbf{x}_{t_j}$
- This is very similar to the optimization problem in the linear separable case, except that there is an upper bound C on α_i .

Non-Linear Classification

- So far, we have only considered large-margin classifiers with a linear decision boundary
- How to generalize it to become nonlinear?
- Key idea: transform \mathbf{x}_i to a higher dimensional space.
 - Input space: the space the point \mathbf{x}_i are located
 - Higher-dimension space: the space of $\phi(\mathbf{x}_i)$ after transformation
- Why transform?
 - Linear operation in the higher-dimensional space is equivalent to non-linear operation in a lower-dimensional space
 - Classification can become easier with a proper transformation.

Transformation

- Computation in the high-dimensional space can be costly.
- The solution: kernel trick.

Kernel Trick

- The SVM optimization problem

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$
 subject to $C \geq \alpha_i \geq 0, \sum_{i=1}^n \alpha_i y_i = 0$
- The data points only appear as **inner product**
- As long as we can calculate the inner product in the transformed space, we do not really need the mapping of data points.
- Define the kernel function K by

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

- Suppose $\phi(\cdot)$ is given as follows

$$\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

- An inner product in the feature space is

$$\langle \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right), \phi\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) \rangle = (1 + x_1y_1 + x_2y_2)^2$$

- So, if we define the kernel function as follows, there is no need to carry out $\phi(\cdot)$ explicitly

$$K(\mathbf{x}, \mathbf{y}) = (1 + x_1y_1 + x_2y_2)^2$$

- This use of kernel function to avoid carrying out $\phi(\cdot)$ explicitly is known as the **kernel trick**.

New Problem

- Change all inner products to kernel functions
- For training,

Original

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

subject to $C \geq \alpha_i \geq 0, \sum_{i=1}^n \alpha_i y_i = 0$

With kernel function

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

subject to $C \geq \alpha_i \geq 0, \sum_{i=1}^n \alpha_i y_i = 0$

Testing

- For testing, the new data \mathbf{z} is classified as class 1 if $f \geq 0$, and as class 2 if $f < 0$

Original

$$\mathbf{w} = \sum_{j=1}^s \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$$

$$f = \mathbf{w}^T \mathbf{z} + b = \sum_{j=1}^s \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}^T \mathbf{z} + b$$

With kernel function

$$\mathbf{w} = \sum_{j=1}^s \alpha_{t_j} y_{t_j} \phi(\mathbf{x}_{t_j})$$

$$f = \langle \mathbf{w}, \phi(\mathbf{z}) \rangle + b = \sum_{j=1}^s \alpha_{t_j} y_{t_j} K(\mathbf{x}_{t_j}, \mathbf{z}) + b$$

- Since the training of SVM only requires the value of $K(\mathbf{x}_i, \mathbf{x}_j)$, there is no restriction of the form of \mathbf{x}_i and \mathbf{x}_j .
- $K(\mathbf{x}_i, \mathbf{x}_j)$ is just a similarity measure comparing \mathbf{x}_i and \mathbf{x}_j .
- For a test object \mathbf{z} , the discriminant function essentially is a weighted sum of the similarity between \mathbf{z} and a pre-selected set of the support vectors.

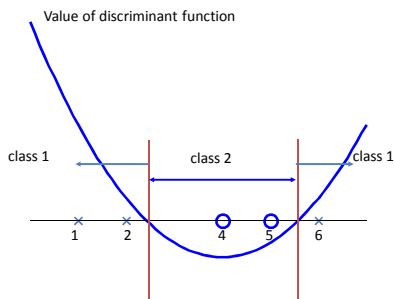
$$f(\mathbf{z}) = \sum_{\mathbf{x}_i \in \mathcal{S}} \alpha_i y_i K(\mathbf{z}, \mathbf{x}_i) + b$$

\mathcal{S} : the set of support vectors

Example

- Suppose we have 5 1D data points
 - $x_1=1, x_2=2, x_3=4, x_4=5, x_5=6$, with 1, 2, 6 as class 1 and 4, 5 as class 2 $\Rightarrow y_1=1, y_2=1, y_3=-1, y_4=-1, y_5=1$
- We use the polynomial kernel of degree 2
 - $K(x,y) = (xy+1)^2$
 - C is set to 100 $\max. \sum_{i=1}^5 \alpha_i - \frac{1}{2} \sum_{i=1}^5 \sum_{j=1}^5 \alpha_i \alpha_j y_i y_j (x_i x_j + 1)^2$
- We first find $\alpha_i (i=1, \dots, 5)$ by
 subject to $100 \geq \alpha_i \geq 0, \sum_{i=1}^5 \alpha_i y_i = 0$

- By using a QP solver, we get
 - $\alpha_1=0, \alpha_2=2.5, \alpha_3=0, \alpha_4=7.333, \alpha_5=4.833$
 - Note that the constraints are indeed satisfied
 - The support vectors are $\{x_2=2, x_4=5, x_5=6\}$
- The discriminant function is $\alpha_5 y_5 K(z, x_5)$
 $f(z) = 2.5(1)(2z+1)^2 + 7.333(-1)(5z+1)^2 + 4.833(1)(6z+1)^2 + b$
 $= 0.6667z^2 - 5.333z + b$
- b is recovered by solving $f(2)=1$ or by $f(5)=-1$ or by $f(6)=1$, as x_2 and x_5 lie on the line $\phi(w)^T \phi(x) + b = 1$ and x_4 lies on the line $\phi(w)^T \phi(x) + b = -1$
- All three give $b=9$
 $f(z) = 0.6667z^2 - 5.333z + 9$



Summary

- SVM is useful.
- Two key concepts of SVM: maximization of the margin and the kernel trick
- A lesson learnt in SVM: a linear algorithm in a HD space is equivalent to a non-linear algorithm in a LD space
- Standard linear algorithms can be generalized to its non-linear version by going to a HD space. E.g. Kernel principal component analysis.
- Many SVM implementations are available on the web for you to try on your data set.