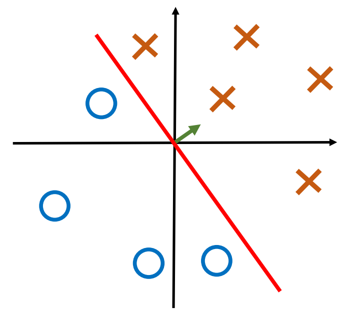
Support Vector Machines

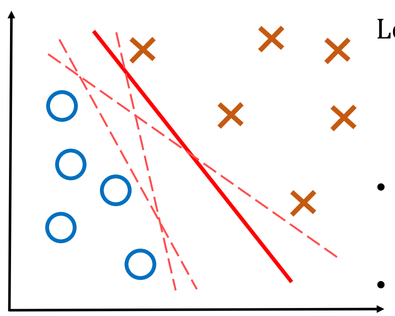
CISC 5800
Professor Daniel Leeds

Separating boundary, defined by **w**



- Separating **hyperplane** splits **class 0** and **class 1**
- Plane is defined by line **w** perpendicular to plan
- Is data point **x** in class 0 or class 1? $\mathbf{w}^T\mathbf{x}+b > 0$ class **1**
  $\mathbf{w}^T\mathbf{x}+b < 0$ class **0**

But, where do we place the boundary?

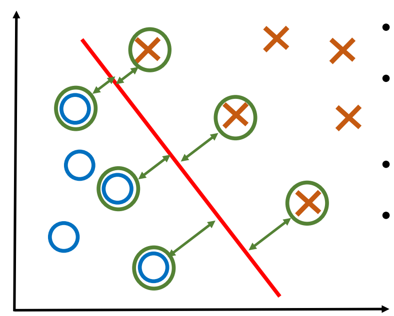


Logistic classifier:

$$LL(y|x;w): \sum_i (y^i - 1)\boldsymbol{w}^T\boldsymbol{x}^i - \log\left(1 + e^{-\boldsymbol{w}^T\boldsymbol{x}^i}\right)$$

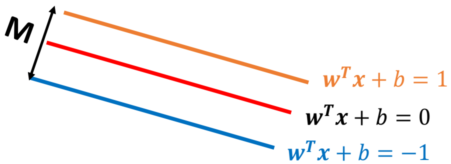
- Each data point $\boldsymbol{x}^i$ considered for boundary $\boldsymbol{w}$
- Outlier data pulls boundary towards it

Max margin classifiers



- Focus on boundary points
- Find largest margin between boundary points on both sides
- Works well in practice
- We can call the boundary points <u>**"support vectors"**</u>

Maximum margin definitions



Classify as +1
if $w^T x + b \geq 1$
Classify as -1
if $w^T x + b \leq -1$
Undefined
if $-1 < w^T x + b < 1$

- M is the margin width
- x^+ is a +1 point closest to boundary, x^- is a -1 point closest to boundary
- $x^+ = \lambda w + x^-$
- $|x^+ - x^-| = M$

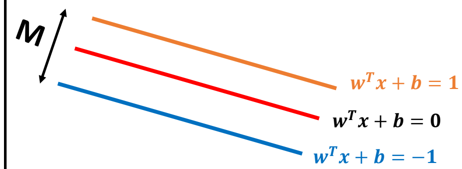
$$M = \frac{2}{\sqrt{w^T w}}$$

maximize M **minimize $w^T w$**

5

λ derivation

Optional extra math



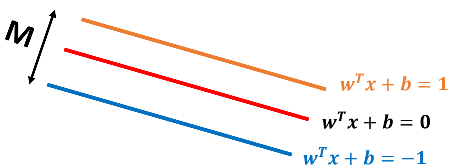
- $w^T x^- + b = -1$
- $w^T x^+ + b = +1$
- $x^+ = \lambda w + x^-$

- $w^T x^+ + b = +1$
- $w^T(\lambda w + x^-) + b = +1$
- $\lambda w^T w + w^T x^- + b = +1$
- $\lambda w^T w - 1 - b + b = +1$
- $\lambda = \frac{2}{w^T w}$

6

M derivation

Optional extra math



- $w^T x^- + b = -1$
- $w^T x^+ + b = +1$
- $x^+ = \lambda w + x^-$
- $|x^+ - x^-| = M$

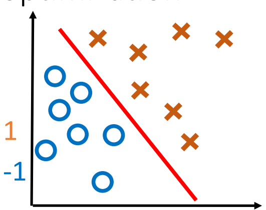
- $M = |\lambda w + x^- - x^-| = |\lambda w| = \lambda |w|$
- $M = \lambda \sqrt{w^T w}$
- $M = \frac{2}{w^T w} \sqrt{w^T w} = \frac{2}{\sqrt{w^T w}}$

maximize M minimize $w^T w$

7

Support vector machine (SVM) optimization

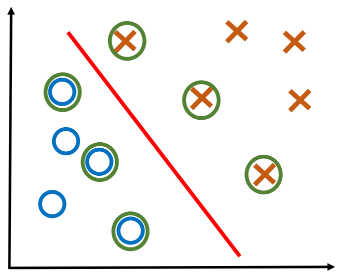
$\text{argmin}_w\ w^T w$
subject to
$w^T x + b \geq 1$ for **x** in class 1
$w^T x + b \leq -1$ for **x** in class -1



$\text{argmin}_w\ w^T w + \left(\sum_{i \in +1} \lambda_i \left(1 - \left(w^T x^i + b\right)\right) + \right.$

9

Alternate SVM formulation



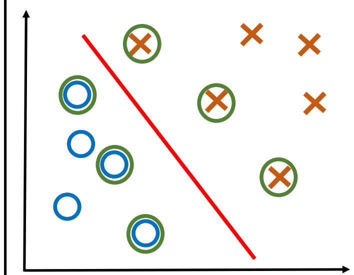
$$\boldsymbol{w} = \sum_i \alpha^i \boldsymbol{x}^i y^i$$

Support vectors $\boldsymbol{x_i}$ have $\alpha_i > 0$

y_i are the data labels +1 or -1

$\boldsymbol{\alpha^i \geq 0 \ \forall i}$ $\quad \boldsymbol{\sum_i \alpha^i y^i = 0}$

Example



$\boldsymbol{w} = \sum_i \alpha^i \boldsymbol{x}^i y^i$ $\quad \boldsymbol{\alpha^i \geq 0 \ \forall i}$, $\boldsymbol{\sum_i \alpha^i y^i = 0}$

$\boldsymbol{x}^1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, y^1 = +1, \alpha^1 = 0.5$

$\boldsymbol{x}^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, y^2 = +1, \alpha^2 = 0.7$

$\boldsymbol{x}^3 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, y^3 = -1, \alpha^3 = 1$

$\boldsymbol{x}^4 = \begin{bmatrix} -0.5 \\ -3 \end{bmatrix}, y^4 = -1, \alpha^4 = 0.2$

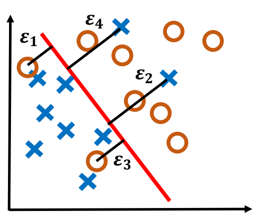
$\boldsymbol{w} =$

$0.5 \times \begin{bmatrix} -1 \\ 1 \end{bmatrix} + 0.7 \times \begin{bmatrix} 0 \\ 0 \end{bmatrix} - 1 \times \begin{bmatrix} -1 \\ -1 \end{bmatrix} - 0.2 \times \begin{bmatrix} -0.5 \\ -3 \end{bmatrix}$

$= \begin{bmatrix} -0.5 + 1 + 0.1 \\ 0.5 + 1 + 0.6 \end{bmatrix} = \begin{bmatrix} \mathbf{0.6} \\ \mathbf{2.1} \end{bmatrix}$

Support vector machine (SVM) optimization
with slack variables

What if data not completely ~~linearly~~ separable?



argmin$_{w,b}$ $\boldsymbol{w}^T\boldsymbol{w} + C\sum_i \varepsilon^i$

subject to

$\boldsymbol{w}^T\boldsymbol{x} + b \geq 1 - \varepsilon^i$ **for x in class 1**

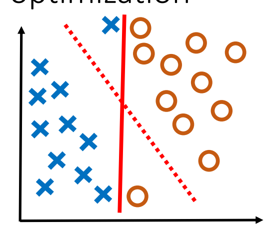
$\boldsymbol{w}^T\boldsymbol{x} + b \leq -1 + \varepsilon^i$ **for x in class -1**

$\varepsilon^i \geq 0 \ \ \forall i$

Each error ε^i is penalized based on distance from separator

Support vector machine (SVM) optimization
with slack variables

Example: Linearly separable but with narrow margins



argmin$_{w,b}$ $\boldsymbol{w}^T\boldsymbol{w} + C\sum_i \varepsilon^i$

subject to

$\boldsymbol{w}^T\boldsymbol{x} + b \geq 1 - \varepsilon^i$ **for x in class 1**

$\boldsymbol{w}^T\boldsymbol{x} + b \leq -1 + \varepsilon^i$ **for x in class -1**

$\varepsilon_i \geq 0 \ \ \forall i$

Hyper-parameters for learning

$$\text{argmin}_{w,b}\ \boldsymbol{w}^T\boldsymbol{w} + C\sum_i \varepsilon_i$$

Optimization constraints: ***C*** influences tolerance for label errors versus narrow margins

$$w_j \leftarrow w_j + \varepsilon \boldsymbol{x}_j^i\left(y^i - g(w^T\boldsymbol{x}^i)\right) - \frac{w_j}{\lambda}$$

Gradient ascent:

- ***ε*** influences effect of individual data points in learning
- ***T*** number of training examples, ***L*** number of loops through data – balance learning and over-fitting

Regularization: ***λ*** influences the strength of your prior belief

Parameter counts

Each data point $\boldsymbol{x}^i$ has *N* features (presuming classify with $\boldsymbol{w}^T\boldsymbol{x}^i+b$)

Separator: ***w*** and *b*

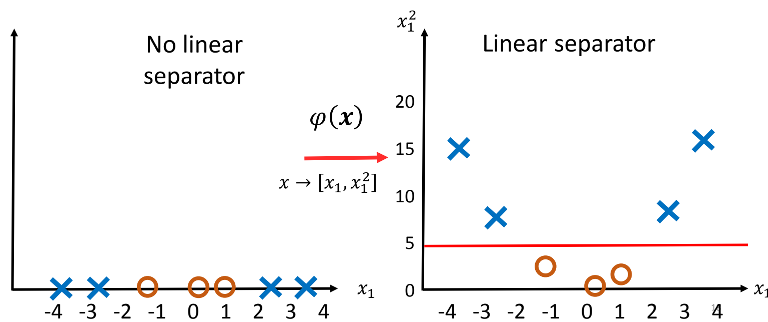
- *N* elements of **w**, 1 value for *b*: *N+1* parameters **OR**
- *t* support vectors -> *t* non-zero α^i, 1 value for *b*: *t+1* parameters

Binary -> *M*-class classification

- Learn boundary for class *m* vs all other classes
  - Only need M-1 separators for M classes – M^th class is for data outside of classes 1, 2, 3, …, M-1
- Find boundary that gives highest margin for data points $\mathbf{x}^i$

Classifying with additional dimensions

Note: More dimensions makes it easier to separate T training points: training error minimized, may risk over-fit



Quadratic mapping function (math)

$w^T x^k + b = \sum_i \alpha^i y^i (x^i)^T x^k + b$

x_1, x_2, x_3, x_4 -> $x_1, x_2, x_3, x_4, x_1^2, x_2^2, \ldots, x_1x_2, x_1x_3, \ldots, x_2x_4, x_3x_4$

N features -> $N + N + \frac{N\times(N-1)}{2} \approx N^2$ features

N^2 values to learn for w in higher-dimensional space

Or, observe: $(\boldsymbol{v}^T\boldsymbol{x}+1)^2 = \boldsymbol{v}_1^2 x_1^2 + \cdots + \boldsymbol{v}_N^2 x_N^2 + \boldsymbol{v}_1\boldsymbol{v}_2 x_1 x_2 + \cdots + \boldsymbol{v}_{N-1}\boldsymbol{v}_N x_{N-1} x_N + \boldsymbol{v}_1 x_1 + \cdots + \boldsymbol{v}_N x_N$

v with N elements operating in quadratic space

19

Quadratic mapping function *Simplified*

x= $[x_1, x_2]$ -> $[\sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2, 1]$

$\mathbf{x}^i$=[5,-2] ->

$\mathbf{x}^k$=[3,-1] ->

$\varphi(\boldsymbol{x}^i)^T \varphi(\boldsymbol{x}^k) =$

Or, observe: $\left({\boldsymbol{x}^i}^T \boldsymbol{x}^k + 1\right)^2 =$

21

Mapping function(s)

- Map from low-dimensional space $\boldsymbol{x} = (x_1, x_2)$ to higher dimensional space $\varphi(\boldsymbol{x}) = (\sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2, 1)$
- N data points guaranteed to be separable in space of N-1 dimensions or more

$$\boldsymbol{w} = \sum_i \alpha_i \varphi(\boldsymbol{x}^i) y^i$$

Classifying $\boldsymbol{x}^k$:

$$\sum_i \alpha_i y^i \varphi(\boldsymbol{x}^i)^T \varphi(\boldsymbol{x}^k) + b$$

22

Kernels

Classifying $\boldsymbol{x}^k$:

$$\sum_i \alpha_i y^i \varphi(\boldsymbol{x}^i)^T \varphi(\boldsymbol{x}^k) + b$$

Kernel trick:

- Estimate high-dimensional dot product with function
- $K(\boldsymbol{x}^i, \boldsymbol{x}^k) = \varphi(\boldsymbol{x}^i)^T \varphi(\boldsymbol{x}^k)$

Now classifying $\boldsymbol{x}^k$

$$\sum_i \alpha_i y^i K(\boldsymbol{x}^i, \boldsymbol{x}^k) + b$$

23

Radial Basis Kernel

Try projection to infinite dimensions

$$\varphi(\boldsymbol{x}) = [x_1 , \cdots , x_n , x_1^2 , \cdots , x_n^2 , \cdots , x_1^\infty \cdots , x_n^\infty]$$

Taylor expansion: $e^x = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^\infty}{\infty!}$

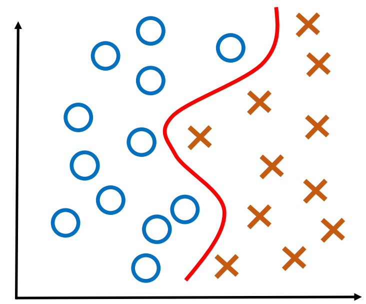
$$K(\boldsymbol{x}^i, \boldsymbol{x}^k) = \exp\left(-\frac{(\boldsymbol{x}^i - \boldsymbol{x}^k)^2}{2\sigma^2}\right)$$

Note: $(\boldsymbol{x}^i - \boldsymbol{x}^k)^2 = (\boldsymbol{x}^i - \boldsymbol{x}^k)^T(\boldsymbol{x}^i - \boldsymbol{x}^k)$

Draw separating plane to curve around all support vectors

24

Example RBF-kernel separator

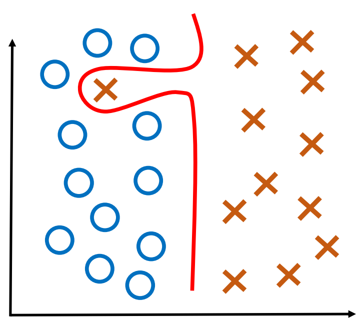


Large margin

Non-linear separation

25

Potential dangers of RBF-kernel separator



Small margin - **overfitting**

Non-linear separation

26

The power of SVM (+kernels)

Boundary defined by a few support vectors

- Caused by: maximizing margin
- Causes: less overfitting
- Similar to: regularization

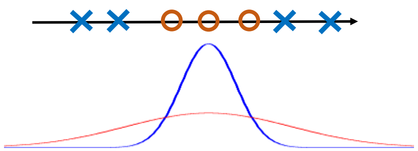
Kernels keep number of learned parameters in check

27

Benefits of generative methods

- $P(\boldsymbol{D}|\boldsymbol{\theta})$ and $P(\boldsymbol{\theta}|\boldsymbol{D})$ can generate non-linear boundary

- E.g.: Gaussians with multiple variances



28