Machine learning - HT 2016
7. Classification: Support Vector Machines

Varun Kanade

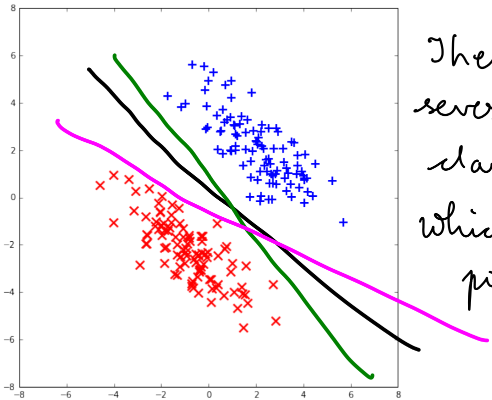
University of Oxford
February 12, 2016

Outline

Today we'll discuss classification using support vector machines.

- No clear probabilistic interpretation
- Maximum Margin Formulation
- Optimisation problem using Hinge Loss
- Dual Formulation

Binary Classification



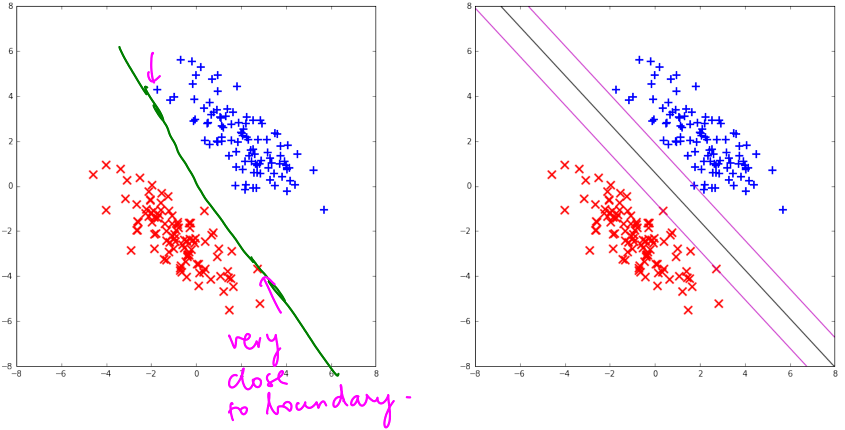
There exist several perfect classifiers

Which one to pick?

Goal: Find a linear separator

Data is linearly separable if there exists a linear separator with no classification error

Maximum Margin Principle

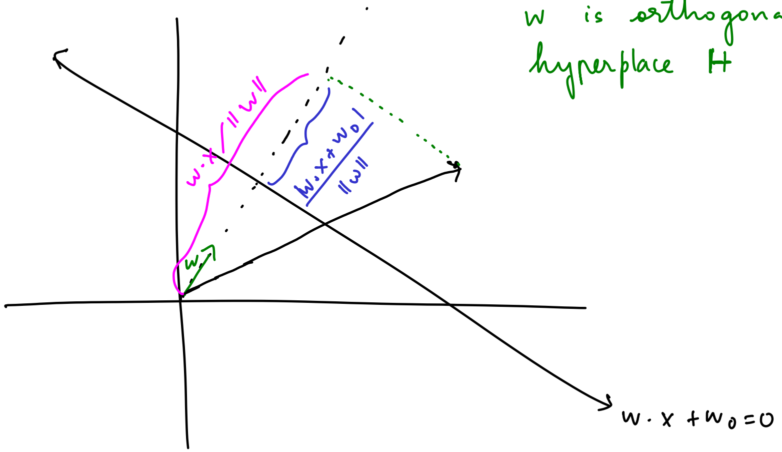


Maximise the distance of the <u>closest</u> point from the decision boundary

Geometry

Given a hyperplane: $H \equiv \mathbf{w} \cdot \mathbf{x} + w_0 = 0$ and a point $\mathbf{x} \in \mathbb{R}^n$

How far is $\mathbf{x}$ from H?



w is orthogonal to hyperplace H

Geometry

For a hyperplane: $H = \mathbf{w} \cdot \mathbf{x} + w_0 = 0$ with $\|\mathbf{w}\|_2 = 1$

The distance of point $\mathbf{x}$ from H is $|\mathbf{w} \cdot \mathbf{x} + w_0|$

If we don't restrict $\|\mathbf{w}\|_2$ to be 1, then the distance is:

$$\frac{|\mathbf{w} \cdot \mathbf{x} + w_0|}{\|\mathbf{w}\|_2}$$

SVM Formulation : Separable Case

with margin :

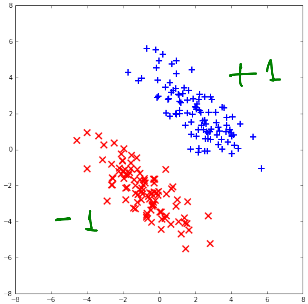
Fix $\|w\|_2 = 1$ ← Not a convex constraint

and then

max α

s.t. $y_i(w \cdot x_i + w_0) \geq \alpha \quad \forall i$

margin



Alternatively fix margin = 1 and minimize $\|w\|_2^2$.

$$\min \tfrac{1}{2} \|w\|_2^2$$

st. $y_i(w \cdot x_i + w_0) \geq 1 \quad \forall i$

QUADRATIC PROGRAM

without margin condition

$y_i(w \cdot x_i + w_0) \geq 1$

(linear program)

No objective, just need to find a feasible point

SVM Formulation : Separable Case

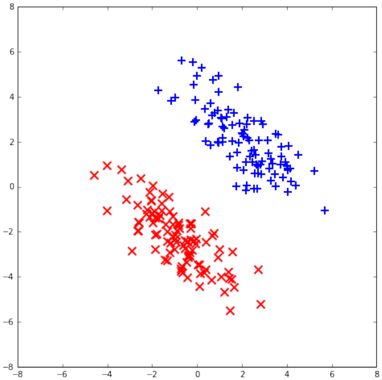
minimise: $\frac{1}{2}\|\mathbf{w}\|_2^2$

subject to:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1$$

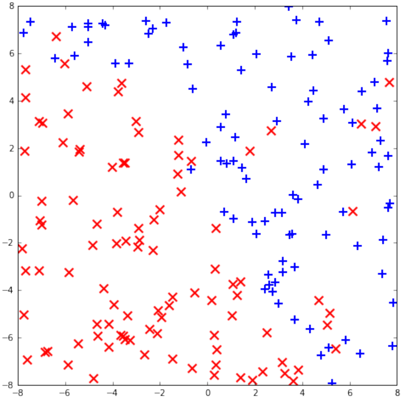
for $i = 1, \ldots, m$

Here $y_i \in \{-1, 1\}$



If data is separable, then we find a classifier with no classification error on the training set

Non-separable Data



Possible choices

(i) Least # misclassified points (NP-hard)

(ii) Sum of distances from boundary

Quadratic program on previous slide has no feasible solution

Which linear separator should we try to find?

SVM Formulation : Non-Separable Case

Penalize constraint violation.

minimise: $\frac{1}{2}\|\mathbf{w}\|_2^2 + C \sum_{i=1}^{m} \zeta_i$

subject to:

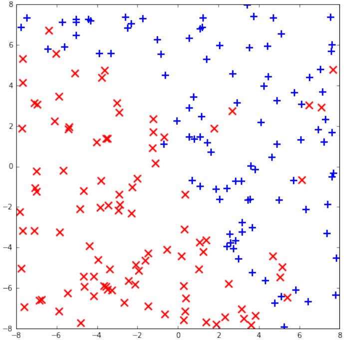
$$y_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1 - \zeta_i$$

allow slack to violate constraint

for $i = 1, \ldots, m$

Here $y_i \in \{-1, 1\}$

$\zeta_i \geq 0$



SVM Formulation : Non-Separable Case

minimise: $\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}\zeta_i$

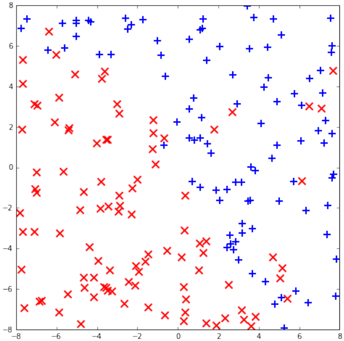
subject to:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 1 - \zeta_i$$

$$\zeta_i \geq 0$$

for $i = 1, \ldots, m$

Here $y_i \in \{-1, 1\}$



SVM Formulation : Loss Function

Regularizer Loss

minimise: $\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}\zeta_i$

subject to:

$y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 1 - \zeta_i$

$\zeta_i \geq 0$

for $i = 1, \ldots, m$

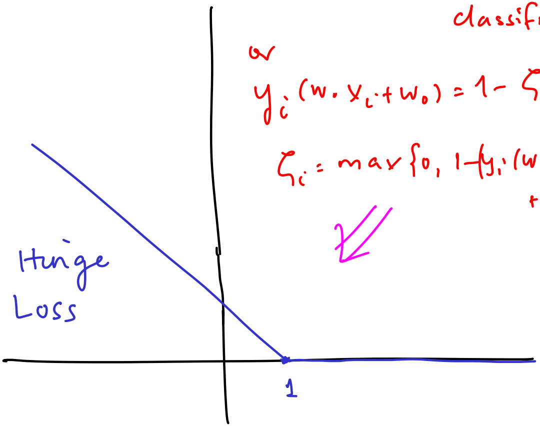
Here $y_i \in \{-1, 1\}$

Either $\zeta_i = 0$ correctly classified

or

$y_i(w \cdot x_i + w_0) = 1 - \zeta_i$

$\zeta_i = \max\{0, 1 - (y_i(w \cdot x_i + w_0))\}$



Logistic Regression: Loss Function

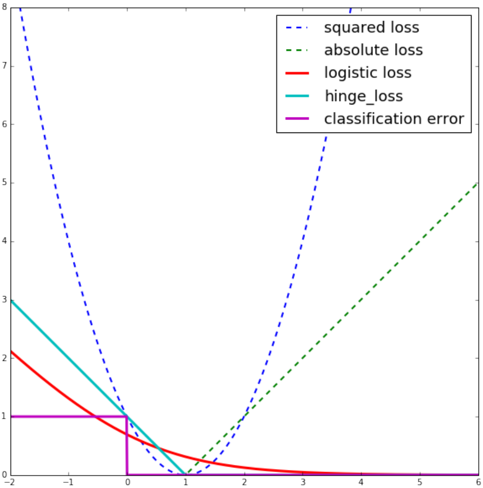
Here $y_i \in \{0, 1\}$

$z_i = -1$ if $y_i = 0$
$= +1$ if $y_i = -1$

$$\ell(\mathbf{w}; \mathbf{x}_i, y_i) = -\left(y_i \log\left(\frac{1}{1 + e^{-\mathbf{w}\cdot\mathbf{x}_i}}\right) + (1 - y_i) \log\left(\frac{1}{1 + e^{\mathbf{w}\cdot\mathbf{x}_i}}\right)\right)$$

$$= \log\left(1 + e^{-z_i \cdot (w \cdot x_i)}\right)$$

Loss Functions



SVM Formulation : Non-Separable Case

minimise: $\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}\zeta_i$

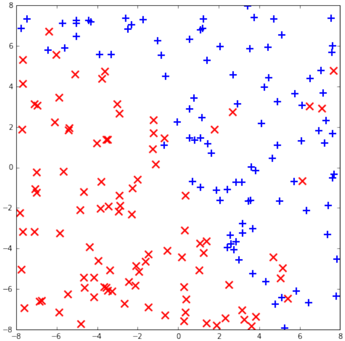
subject to:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 1 - \zeta_i$$

$$\zeta_i \geq 0$$

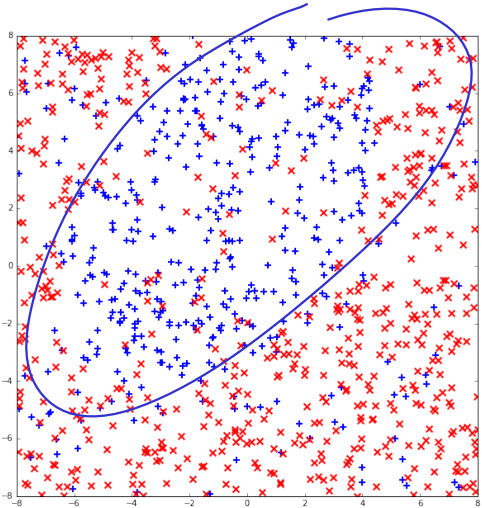
for $i = 1, \ldots, m$

Here $y_i \in \{-1, 1\}$



SVM Formulation: Non-Separable Case

What if your data looks like this?



Use basis expansion.

e.g.

Quadratic

SVM Formulation : Constrained Minimisation

minimise: $\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}\zeta_i$

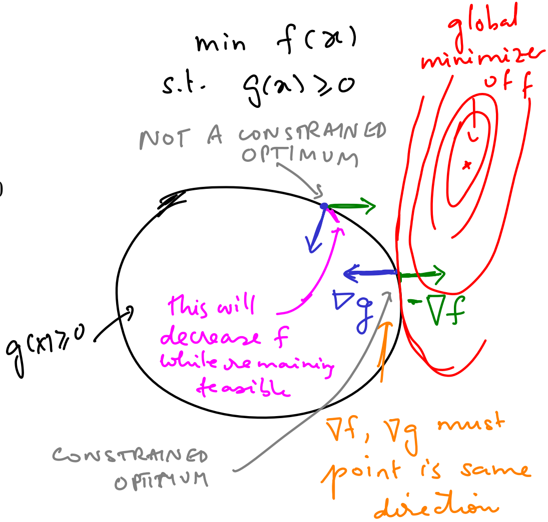
subject to:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) - (1-\zeta_i) \geq 0$$

$$\zeta_i \geq 0$$

for $i = 1, \ldots, m$

Here $y_i \in \{-1, 1\}$



SVM Formulation

minimise: $\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}\zeta_i$

subject to:

$$\begin{aligned} \alpha_i: &\quad y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) - (1-\zeta_i) \geq 0 \\ \mu_i: &\quad \zeta_i \geq 0 \end{aligned}$$

for $i = 1, \ldots, m$

Here $y_i \in \{-1, 1\}$

$\alpha_i \geq 0$

$\mu_i \geq 0$

Constrained optima are stationary points of L for suitably chosen α_i, μ_i

Lagrange Function

$$L(\mathbf{w}, w_0, \boldsymbol{\zeta}; \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\zeta_i - \sum_{i=1}^{m}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) - (1-\zeta_i)) - \sum_{i=1}^{m}\mu_i\zeta_i$$

Dual Formulation

[see chap 5 of Boyd Vandenberghe
CONVEX OPTIMIZATION)

(Idealised) Lagrange Function

$$L_{\text{ideal}}(\mathbf{w}, w_0, \boldsymbol{\zeta}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^m \zeta_i - \sum_{i=1}^m \mathbb{I}_{\geq 0}(\underline{y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) - (1-\zeta_i)}) - \sum_{i=1}^m \mathbb{I}_{\geq 0}(\zeta_i),$$

where $\mathbb{I}_{\geq 0}(x) = 0$ if $x \geq 0$ and $-\infty$ otherwise.

blows up to $+\infty$
on infeasible points

Lagrange Function

$$L(\mathbf{w}, w_0, \boldsymbol{\zeta}; \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^m \zeta_i - \sum_{i=1}^m \alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) - (1-\zeta_i)) - \sum_{i=1}^m \mu_i\zeta_i$$

For any $\boldsymbol{\alpha}, \boldsymbol{\mu} \geq 0$, we have for all $\mathbf{w}, w_0, \boldsymbol{\zeta}$

$$L(\mathbf{w}, w_0, \boldsymbol{\zeta}; \boldsymbol{\alpha}, \boldsymbol{\mu}) \leq \underline{L_{\text{ideal}}(\mathbf{w}, w_0, \boldsymbol{\zeta})}$$

Dual Formulation

say w^*, w_0^*, ζ^* are minimizers of L_{ideal}

Lagrange Function

$$L(\mathbf{w}, w_0, \boldsymbol{\zeta}; \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\zeta_i - \sum_{i=1}^{m}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) - (1-\zeta_i)) - \sum_{i=1}^{m}\mu_i\zeta_i$$

$$g(\boldsymbol{\alpha}, \boldsymbol{\mu}) = \inf_{\mathbf{w}, w_0} L(\mathbf{w}, w_0, \boldsymbol{\zeta}; \boldsymbol{\alpha}, \boldsymbol{\mu}) \leq L(w^*, w_0^*, \zeta^*; \alpha, \mu)$$

Maximising $g(\boldsymbol{\alpha}, \boldsymbol{\mu})$ will give us the "best possible" lower bound $\leq L_{ideal}(w^*, w_0^*, \zeta^*)$

$=$ value of objective

maximizing $g(\alpha, \mu)$ will give us "best" possible lower bound.

In our case STRONG DUALITY holds

Dual Formulation

Lagrange Function

$$L(\mathbf{w}, w_0, \boldsymbol{\zeta}; \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\zeta_i - \sum_{i=1}^{m}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) - (1-\zeta_i)) - \sum_{i=1}^{m}\mu_i\zeta_i$$

For $\alpha, \mu, \geq 0$

minimize L wrt w, w_0, ζ

$\nabla_w L = w - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow w = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$

$\frac{\partial L}{\partial w_0} = -\sum_{i=1}^{m} \alpha_i y_i = 0 \Rightarrow \sum_i \alpha_i y_i = 0$

$\frac{\partial L}{\partial \zeta_i} = C - \alpha_i - \mu_i = 0 \Rightarrow \alpha_i = C - \mu_i$

$\Rightarrow 0 \leq \alpha_i \leq C$ [as μ, α are both ≥ 0].

$$\boxed{\begin{aligned} &\text{DUAL PROBLEM: } \max: -\frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_i \alpha_i \\ &\text{s.t. } \sum_i y_i \alpha_i = 0 \text{ and } 0 \leq \alpha_i \leq C \ \forall i \end{aligned}}$$

Dual Formulation

Primal Form

minimise: $\frac{1}{2}\|\mathbf{w}\|_2^2$

subject to:

$y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 1 - \zeta_i$

$\zeta_i \geq 0$

for $i = 1, \ldots, m$

Dual Form

maximise: $\sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$

subject to:

$C \geq \alpha_i \geq 0$

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

dual variables = m

objective only depends on $x_i^T \cdot x_j$

Relating primal & dual solutions

$$\boxed{w^* = \sum_i \alpha_i^* y_i \mathbf{x}_i}$$

Lagrange Function

$$L(\mathbf{w}, w_0, \boldsymbol{\zeta}; \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\zeta_i - \sum_{i=1}^{m}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) - (1-\zeta_i)) - \sum_{i=1}^{m}\mu_i\zeta_i$$

sparse if most $\alpha_i = 0$

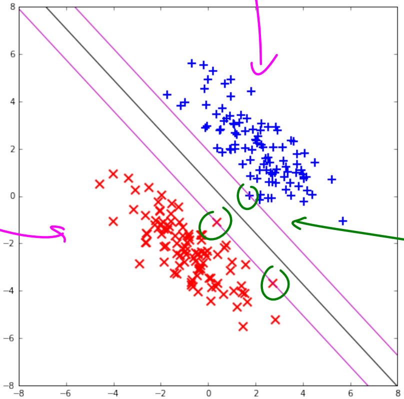
KKT Conditions

$\alpha_i^* = 0$ or $y_i(\mathbf{w}^* \cdot \mathbf{x}_i + w_0^*) - 1 = 0,$ where $\mathbf{w}^* = \sum_{i=1}^{m} \alpha_i^* y_i \mathbf{x}_i$

Support Vectors

For all other points

$\alpha_i = 0$



Support vectors

$\alpha_i \neq 0$

SVM Dual Program

maximise: $$\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

subject to:

$$C \geq \alpha_i \geq 0$$

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

- Objective depends only between dot products of training inputs
- Particularly useful if inputs are high-dimensional
- To make a new prediction only need to know dot product with support vectors

Gram Matrix

If we put the inputs in matrix $\mathbf{X}$, where the i^{th} row of $\mathbf{X}$ is $\mathbf{x}_i^T$.

$$\mathbf{K} = \mathbf{X}\mathbf{X}^T = \begin{bmatrix} \langle \mathbf{x}_1, \mathbf{x}_1 \rangle & \langle \mathbf{x}_1, \mathbf{x}_2 \rangle & \cdots & \langle \mathbf{x}_1, \mathbf{x}_m \rangle \\ \langle \mathbf{x}_2, \mathbf{x}_1 \rangle & \langle \mathbf{x}_2, \mathbf{x}_2 \rangle & \cdots & \langle \mathbf{x}_2, \mathbf{x}_m \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{x}_m, \mathbf{x}_1 \rangle & \langle \mathbf{x}_m, \mathbf{x}_2 \rangle & \cdots & \langle \mathbf{x}_m, \mathbf{x}_m \rangle \end{bmatrix}$$

- The matrix K is positive definite (if $n > m$ and $\mathbf{x}_i$ are linearly independent)
- If we do feature expansion first

$$\phi : \mathbb{R}^n \to \mathbb{R}^N$$

then replace entries by $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$

Kernel Trick

Suppose we do basis expansion with up to degree two terms (say $n = 2$)

$$\phi((x_1, x_2)) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2)$$

If $\mathbf{x} = (x_1, x_2)$ and $\mathbf{x}' = (x_1', x_2')$ then what is $\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$?

$$\langle \phi(x), \phi(x') \rangle = 1 + 2x_1x_1' + 2x_2x_2' + x_1^2(x_1')^2 + x_2^2 \cdot (x_2')^2 + 2x_1x_2x_1'x_2'$$

$$= (1 + x_1x_1' + x_2x_2')^2$$

$$= (1 + \langle x, x' \rangle)^2$$

To compute matrix entries, only need $O(n)$ time!

Kernel Trick

We can use a symmetric positive definite matrix (Mercer Kernels)

$$\mathbf{K} = \begin{bmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \kappa(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_1, \mathbf{x}_m) \\ \kappa(\mathbf{x}_2, \mathbf{x}_1) & \kappa(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_2, \mathbf{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_m, \mathbf{x}_1) & \kappa(\mathbf{x}_m, \mathbf{x}_2) & \cdots & \kappa(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix}$$

Here $\kappa(\mathbf{x}, \mathbf{x}')$ is some measure of similarity between $\mathbf{x}$ and $\mathbf{x}'$

The dual program becomes

$$\text{maximise} \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K_{i,j} \qquad \text{subject to} : 0 \leq \alpha_i \leq C$$

To make prediction on new $\mathbf{x}_{\text{new}}$, need to compute $\kappa(\mathbf{x}_i, \mathbf{x}_{\text{new}})$ for support vectors $\mathbf{x}_i$

Polynomial Kernels

Rather than perform basis expansion,

$$\kappa(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x} \cdot \mathbf{x}')^d$$

This gives all terms of degree up to d

If we use $\kappa(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^d$, we get only degree d terms

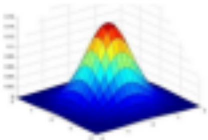
<u>Linear Kernel</u>: $\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x} \cdot \mathbf{x}'$

All of these satisfy the Mercer or positive-definite condition

Gaussian or RBF Kernel

Radial Basis Function (RBF) or Gaussian Kernel

$$\kappa(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$



σ^2 is known as the bandwidth

Can generalise to more general covariance matrices

Results in a Mercel kernel

Kernels on Discrete Data : Cosine Kernel

For text documents: let $\mathbf{x}$ denote bag of words

Cosine Similarity

$$\kappa(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x} \cdot \mathbf{x}'}{\|\mathbf{x}\|_2 \|\mathbf{x}'\|_2}$$

Term frequency $\text{tf}(c) = \log(1 + c)$, c word count for some word w

Inverse document frequency $\text{idf}(w) = \log\left(\frac{N}{1+N_w}\right)$, N_w #docs containing w

$\text{tf-idf}(\mathbf{x})_w = \text{tf}(x_w)\text{idf}(w)$

Kernels on Discrete Data : String Kernel

Let $\mathbf{x}$ and $\mathbf{x}'$ be strings over some alphabet $\mathcal{A}$

$$\mathcal{A} = \{A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$$

IPTSALVKETLALLSTHRTLLIANETLRIPVPVHKNHQLCTEEIFQGIGTLESQTVQGGTV
ERLFKNLSLIKKYIDGQKKKCGEERRRVNQFLDYLQEFLGVMNTEWI

PHRRDLCSRSIWLARKIRSDLTALTESYVKHQGLWSELTEAERLQENLQAYRTFHVLLA
RLLEDQQVHFTPTEGDFHQAIHTLLLQVAAFAYQIEELMILLEYKIPRNEADGMLFEKK
LWGLKVLQELSQWTVRSIHDLRFISSHQTGIP

$\kappa(\mathbf{x}, \mathbf{x}') = \sum_s w_s \phi_s(\mathbf{x}) \phi_s(\mathbf{x}')$

$\phi_s(\mathbf{x})$ is the number of times s appears in $\mathbf{x}$ as substring

How to choose a good kernel?

Not always easy to tell whether a function is a Mercer kernel

For any finite set of inputs, the Kernel matrix should be positive definite

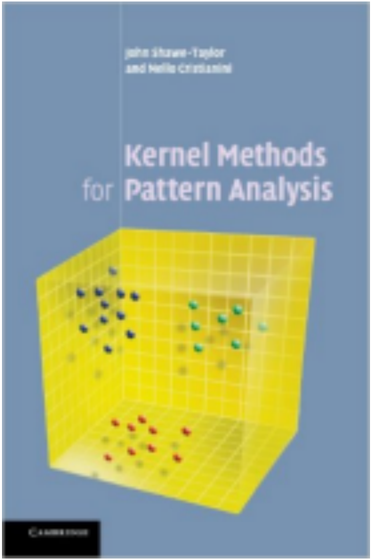
If the following hold:

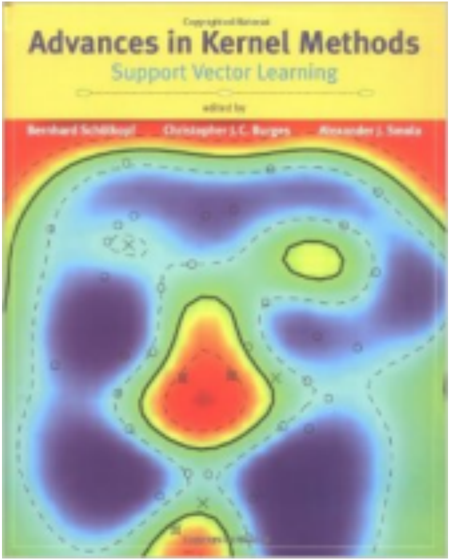
- κ_1, κ_2 are Mercer kernels for points in $\mathbb{R}^n$
- $f : \mathbb{R}^n \to \mathbb{R}$
- $\phi : \mathbb{R}^n \to \mathbb{R}^N$
- κ_3 is a Mercer kernel on $\mathbb{R}^N$

the following are Mercer kernels

- $\kappa_1 + \kappa_2$, $\kappa_1 \cdot \kappa_2$, $\alpha\kappa_1$ for $\alpha \in \mathbb{R}^+$
- $\kappa(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})f(\mathbf{x}')$
- $\kappa_3(\phi(\mathbf{x}), \phi(\mathbf{x}'))$
- $\kappa(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{A}\mathbf{x}'$ for $\mathbf{A}$ positive definite

Books


Kernel Methods
for Pattern Analysis


Advances in Kernel Methods
Support Vector Learning

Kernel Trick in Linear Regression

Recall the loss function for linear regression (least squares)

$$\mathrm{L}(\mathbf{w}) = \sum_{i=1}^{m} (\mathbf{x}_i^T \mathbf{w} - y_i)^2$$

and the solution $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{y})$.

Can write $\hat{\mathbf{w}} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i$. Why?

Suppose $\hat{w} = \tilde{w} + u$. where $u \perp x_i \ \forall i$.
and $u \perp \tilde{w}$

Then $\langle \hat{w}, x_i \rangle = \langle \tilde{w}, x_i \rangle$.

So $\tilde{w}$ is identical solution.

If we had regularized, then u is forced to be 0, since $\|\hat{w}\|^2 = \|\tilde{w}\|^2 + \|u\|^2$

Support Vector Regression

ϵ-sensitive loss function

$$\ell_\epsilon(y, \hat{y}) = \begin{cases} 0 & \text{if } |y - \hat{y}| < \epsilon \\ |y - \hat{y}| - \epsilon & \text{otherwise} \end{cases}$$

Loss function:

$$\mathrm{L}(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \ell_\epsilon(\mathbf{x}_i^T \mathbf{w}, y_i)$$