Support Vector Machines

Statistical Learning Reading Group

Udo Boehm

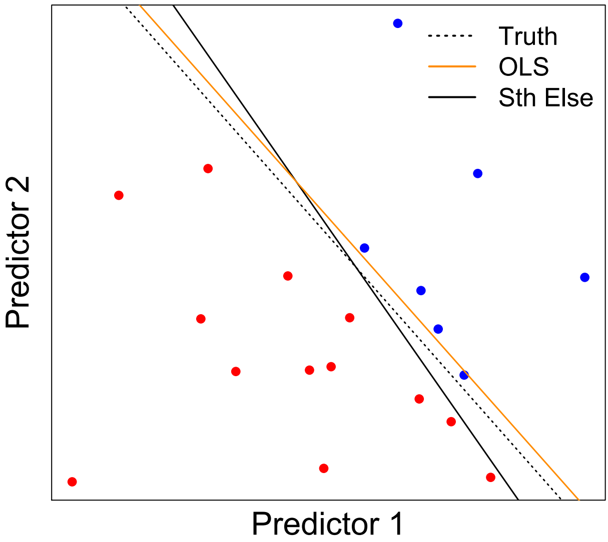
2 July 2016

Classification Problem I

Separate n-dimensional data $\boldsymbol{x}_i$ into 2 classes y_i:

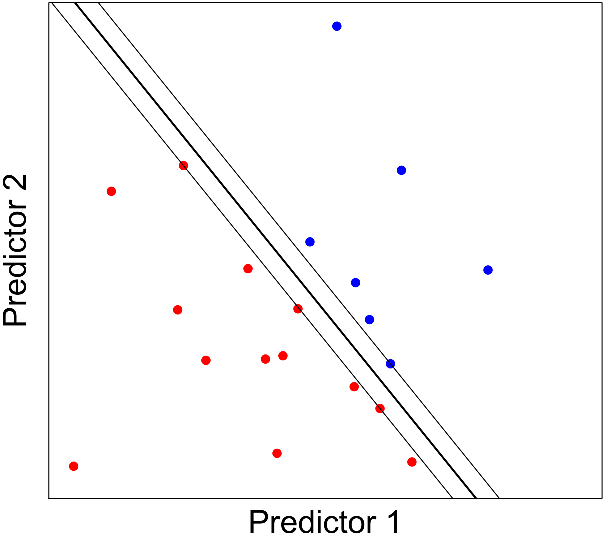
$$\hat{y}_i = \begin{cases} -1 & \text{if } \hat{f}(\boldsymbol{x}_i) < 0 \\ 1 & \text{if } \hat{f}(\boldsymbol{x}_i) > 0 \end{cases}$$

- Simplest case: separable classes


Truth
OLS
Sth Else
Predictor 1
Predictor 2

Separating Hyperplanes

- Problem: choosing a separating hyperplane
- A good criterion would be prediction performance, i.e. minimal misclassification of test data
- Maximise separating margin


Predictor 2
Predictor 1

Separating Hyperplanes

- Problem: choosing a separating hyperplane
- A good criterion would be prediction performance, i.e. minimal misclassification of test data
- Maximise separating margin

Separating hyperplane L is given by:

$$0 = \boldsymbol{\beta}^T \cdot \boldsymbol{x} + \beta_0,$$

which means $\boldsymbol{\beta}/\|\boldsymbol{\beta}\|$ is normal to the hyperplane and the signed distance of any data point $\boldsymbol{x}_i$ to the plane is given by:

$$\mathrm{d}(L, \boldsymbol{x}_i) = (\boldsymbol{\beta}^T \cdot \boldsymbol{x}_i + \beta_0)/\|\boldsymbol{\beta}\|$$

Maximising the Margin

The unsigned distance is ($y_i = \pm 1$):

$$\mathrm{d}(L, \boldsymbol{x}_i) = y_i(\boldsymbol{\beta}^T \cdot \boldsymbol{x}_i + \beta_0)/\|\boldsymbol{\beta}\|$$

and the optimisation problem is:

$$\max_{\boldsymbol{\beta}, \beta_0, \|\boldsymbol{\beta}\|=1} M$$

with inequality constraint:

$$y_i(\boldsymbol{\beta}^T \cdot \boldsymbol{x}_i + \beta_0) \geq M, i = 1 \ldots N.$$

Maximising the Margin

Setting $\|\boldsymbol{\beta}\| = 1/M$, the problem becomes:

$$\min_{\boldsymbol{\beta},\beta_0} \frac{1}{2}\|\boldsymbol{\beta}\|^2$$

with inequality constraint:

$$y_i(\boldsymbol{\beta}^T \cdot \boldsymbol{x}_i + \beta_0) \geq 1, i = 1 \ldots N.$$

Convex (quadratic) optimisation problem with linear inequality constraints.

Maximising the Margin

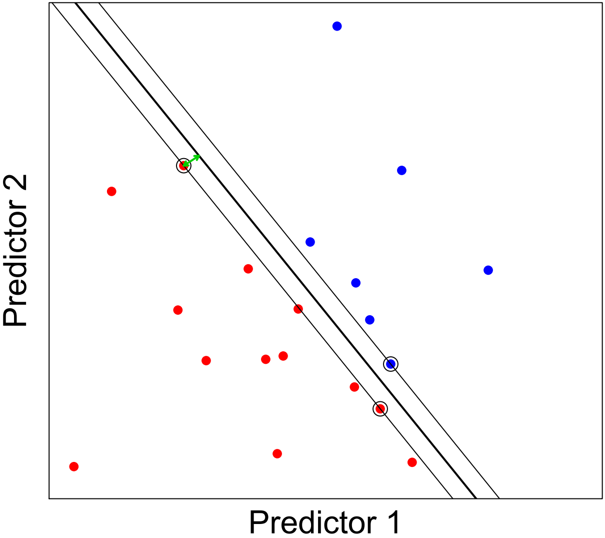
Introducing KKT-conditions and Lagrange multipliers α_i:

$$\alpha_i[y_i(\boldsymbol{\beta}^T \cdot \boldsymbol{x}_i + \beta_0) - 1] = 0 \; \forall i,$$

the solution has the form:

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i$$

$\Rightarrow$ if $\alpha_i > 0$, $\boldsymbol{x}_i$ is on the boundary (support point)
$\Rightarrow$ if $\boldsymbol{x}_i$ is not on the boundary, $\alpha_i = 0$



Predictor 2
Predictor 1

Prediction

Separating hyperplane:

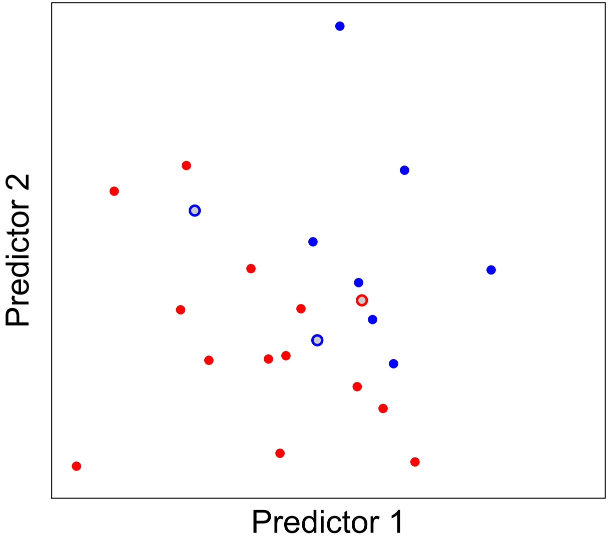
$$\hat{f}(\boldsymbol{x}) = \hat{\boldsymbol{\beta}}^T \cdot \boldsymbol{x} + \hat{\beta}_0$$

Prediction:

$$\hat{y} = \text{sign}(\hat{f}(\boldsymbol{x}))$$

Classification Problem II

- Things get more interesting when classes are not (linearly) separable
- Possible solution: allow for some violation of the margin (soft margin)


Predictor 2
Predictor 1

Support Vector Classifier

Introducing slack variables:

$$y_i(\boldsymbol{\beta}^T \cdot \boldsymbol{x}_i + \beta_0) \geq M(1 - \xi_i)$$

with constraints:

$$\sum_{i=1}^{N} \xi_i \leq K, \text{and } \xi_i \geq 0 \; \forall i.$$

- $\xi_i > 0 \Rightarrow$ observation lies within the margin
- $\xi_i > 1 \Rightarrow$ observation misclassified
- $\sum_{i=1}^{N} \xi_i \leq K$ bounds total number of misclassifications to $\lfloor K \rfloor$

Maximising the Soft Margin

The optimisation problem is:

$$\max_{\boldsymbol{\beta}, \beta_0, \|\boldsymbol{\beta}\|=1} M$$

with:

$$y_i(\boldsymbol{\beta}^T \cdot \boldsymbol{x}_i + \beta_0) \geq M(1 - \xi_i), \text{ and}$$

$$\sum_{i=1}^{N} \xi_i \leq K, \; \xi_i \geq 0 \; \forall i.$$

Maximising the Soft Margin

Setting $\|\boldsymbol{\beta}\| = 1/M$, this becomes:

$$\min_{\boldsymbol{\beta},\beta_0} \frac{1}{2}\|\boldsymbol{\beta}\|^2$$

with inequality constraints:

$$y_i(\boldsymbol{\beta}^T \cdot \boldsymbol{x}_i + \beta_0) \geq 1 - \xi_i, \text{ and}$$

$$\sum_{i=1}^{N} \xi_i \leq K, \ \xi_i \geq 0 \ \forall i.$$

Maximising the Soft Margin

For computational optimisation, constraints on the slack variables are added to the objective function:

$$\min_{\boldsymbol{\beta}, \beta_0} \frac{1}{2} \|\boldsymbol{\beta}\|^2 + C \sum_{i=1}^{N} \xi_i$$

(explicit minimisation w.r.t. ξ_i)

- C is the cost for boundary violations
- $C = \infty$ forces perfect separation
- C provides tradeoff between fit and generalisability
- Optimal C can be estimated by cross-validation

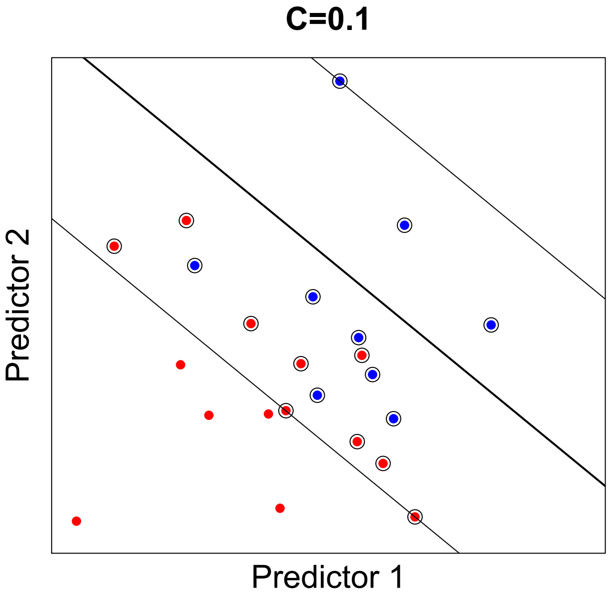
Convex (quadratic) optimisation problem with linear inequality constraints.

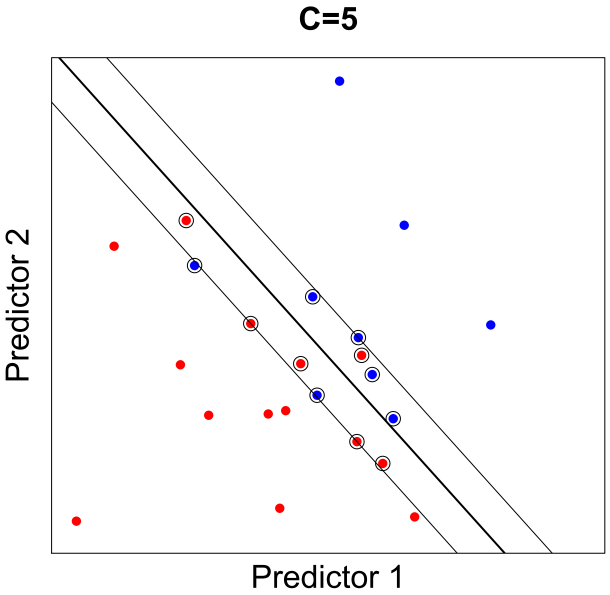
Optimal Solution

Introducing KKT conditions and Lagrange multipliers, the solution has the form:

$$\hat{\beta} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

- Only observations on or within the margin contribute to $\hat{\beta}$ (support points)
- Points on the margin contribute with weight $0 < \alpha_i < C$
- Points within the margin contribute with weight $\alpha_i = C$


C=0.1
Predictor 2
Predictor 1


C=5
Predictor 2
Predictor 1

Prediction

Prediction as before:

$$\hat{f}(\boldsymbol{x}) = \hat{\boldsymbol{\beta}}^T \cdot \boldsymbol{x} + \hat{\beta}_0$$

$$\hat{y} = \text{sign}(\hat{f}(\boldsymbol{x}))$$

Classification Problem III

- Things get more interesting when classes are not (linearly) separable
- Possible solution: allow for some misclassification
- Further extension through non-linear boundaries

Enlarged Feature Space

Goal: improved classification

Procedure:

- Add transformations of input features $h_m(\boldsymbol{x}),\ m = 1, \ldots, M$ to basis
- Fit SV classifier to enlarged feature space $\boldsymbol{h}(x_i) = (h_1(x_i), h_2(x_i), \ldots, h_M(x_i))$
- Linear boundary in enlarged space = nonlinear boundary in original space
- Potential problems are computational costs for $\boldsymbol{h}$ and overfitting

Reproducing Kernel Hilbert Space

Hilbert Space $\mathcal{H}$ of functions over some bounded domain $X \subset \mathbb{R}^k$, and for each $\boldsymbol{x} \in X$, the evaluation functionals $\mathcal{F}_{\boldsymbol{x}}$:

$$\mathcal{F}_{\boldsymbol{x}}[f] = f(\boldsymbol{x})$$

are linear, bounded functionals, i.e. $\exists U = U_{\boldsymbol{x}} \in \mathbb{R}^+ :$

$$|\mathcal{F}_{\boldsymbol{x}}[f]| = |f(\boldsymbol{x})| \leq U\|f\|.$$

Then there is a unique positive definite function $K(\boldsymbol{x}, \boldsymbol{y})$, the reproducing kernel, with reproducing property:

$$f(\boldsymbol{x}) = \langle f(\boldsymbol{y}), K(\boldsymbol{x}, \boldsymbol{y}) \rangle_{\mathcal{H}} \ \forall f \in \mathcal{H}$$

Constructing an RKHS

For linearly independent functions $\phi_n(\boldsymbol{x})$,

$$f(\boldsymbol{x}) = \sum_{m=0}^{\infty} a_m \phi_m(\boldsymbol{x})$$

and

$$K(\boldsymbol{x}, \boldsymbol{y}) = \sum_{m=0}^{\infty} \lambda_m \phi_m(\boldsymbol{x}) \phi_m(\boldsymbol{y}).$$

Define the scalar product:

$$\langle f(\boldsymbol{x}), g(\boldsymbol{x}) \rangle_{\mathcal{H}} = \langle \sum_{m=0}^{\infty} a_m \phi_m(\boldsymbol{x}), \sum_{m=0}^{\infty} d_m \phi_m(\boldsymbol{x}) \rangle_{\mathcal{H}} = \sum_{m=0}^{\infty} \frac{a_m d_m}{\lambda_m},$$

which gives the reproducing property:

$$\langle f(\boldsymbol{y}), K(\boldsymbol{y}, \boldsymbol{x}) \rangle_{\mathcal{H}} = \sum_{m=0}^{\infty} \frac{a_m \lambda_m \phi_m(\boldsymbol{x})}{\lambda_m} = f(\boldsymbol{x})$$

Constructing an RKHS

and norm:

$$\|f\|_K^2 = \sum_{m=0}^{\infty} \frac{a_m^2}{\lambda_m}.$$

Example

For $\boldsymbol{x} = [x_1, x_2] \in \mathbb{R}^2$ and basis $h(\boldsymbol{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$ (2nd degree polynomial):

$$\begin{aligned}
\langle h(\boldsymbol{x}), h(\boldsymbol{y}) \rangle &= \sum_{m=1}^{6} h_m(\boldsymbol{x}) h_m(\boldsymbol{y}) \\
&= 1 + 2x_1y_1 + 2x_2y_2 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1y_1x_2y_2 \\
&= (\langle \boldsymbol{x}, \boldsymbol{y} \rangle + 1)^2 \\
&= K(\boldsymbol{x}, \boldsymbol{y}) \text{ with } K = (\langle \boldsymbol{x}, \boldsymbol{y} \rangle + 1)^2
\end{aligned}$$

Inner products in the enlarged feature space can be computed through the Kernel function.

Kernel Functions

Regularisation Network	Kernel Function
Polynomial of degree d	$K(\boldsymbol{x}, \boldsymbol{y}) = (\langle \boldsymbol{x}, \boldsymbol{y} \rangle + 1)^d$
Gaussian radial basis	$K(\boldsymbol{x}, \boldsymbol{y}) = exp(-\gamma \lVert \boldsymbol{x} - \boldsymbol{y} \rVert^2)$
Thin plate spline	$K(\boldsymbol{x}, \boldsymbol{y}) = \lVert \boldsymbol{x} - \boldsymbol{y} \rVert^{2n-1}$
	$K(\boldsymbol{x}, \boldsymbol{y}) = \lVert \boldsymbol{x} - \boldsymbol{y} \rVert^{2n} \log(\lVert \boldsymbol{x} - \boldsymbol{y} \rVert)$
Multilayer perceptron	$K(\boldsymbol{x}, \boldsymbol{y}) = \tanh(\langle \boldsymbol{x}, \boldsymbol{y} \rangle - \theta)$

(See Evgeniou et al., 1999 for more examples)

Non-Linear Boundaries as Inner Products

We want to solve the optimisation problem:

$$\min_{\boldsymbol{\beta},\beta_0} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N}\xi_i$$

on the enlarged feature space $h(\boldsymbol{x})$ instead of $\boldsymbol{x}$.
Introducing KKT conditions and Lagrange multipliers, the solution has the form:

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{N}\alpha_i y_i h(\boldsymbol{x}_i)$$

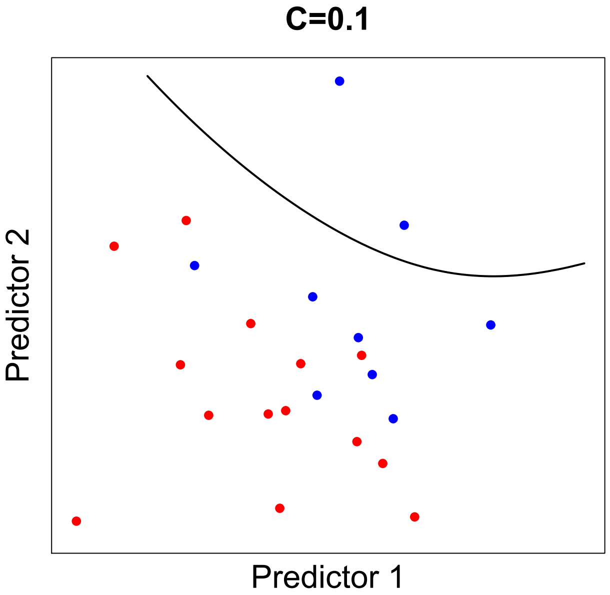
which can be rewritten as a scalar product:

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{N}\alpha_i y_i \langle h(\boldsymbol{x}), h(\boldsymbol{x}_i)\rangle = \sum_{i=1}^{N}\alpha_i y_i K(\boldsymbol{x},\boldsymbol{x}_i).$$

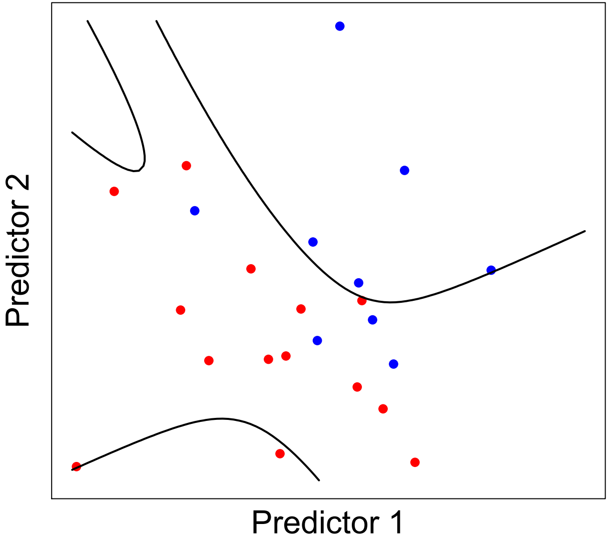
Non-Linear Boundaries as Inner Products

Role of the cost parameter C:

- Large C $\Rightarrow$ wiggly boundary (overfit)
- Small C $\Rightarrow$ smooth boundary


C=0.1
Predictor 2
Predictor 1

C=15



SVMs as a Penalisation Method

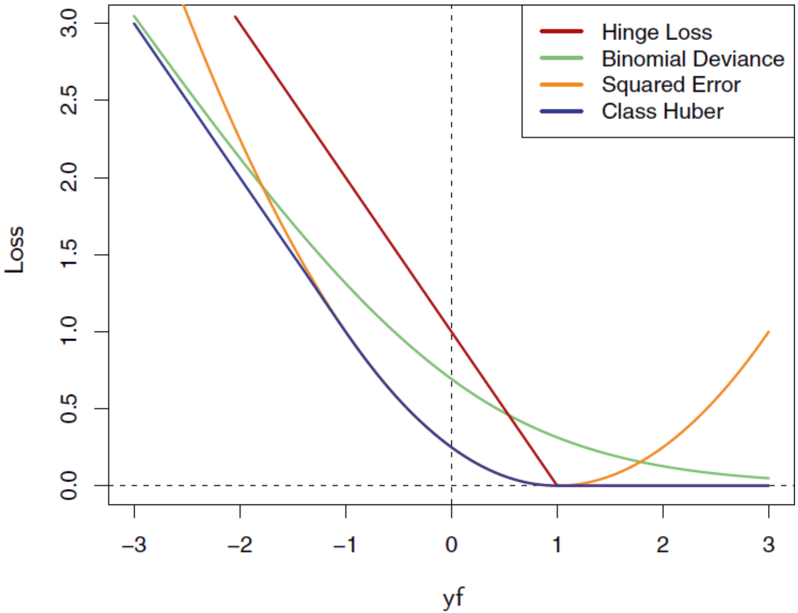
The optimisation problem:

$$\min_{\boldsymbol{\beta}, \beta_0} \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N}\xi_i$$

is equivalent to the problem:

$$\min_{\boldsymbol{\beta}, \beta_0} \sum_{i=1}^{N}(1 - y_i f(\boldsymbol{x}_i))_+ + \frac{\kappa}{2}\|\boldsymbol{\beta}\|^2,$$

which is of the form *loss* + *penalty*.
Hinge loss is preferable to other loss functions, e.g., squared loss, that also penalise correctly classified points.


Hinge Loss
Binomial Deviance
Squared Error
Class Huber
Loss
0.0
0.5
1.0
1.5
2.0
2.5
3.0
−3
−2
−1
0
1
2
3
yf

SVMs and the Curse of Dimensionality

The optimisation problem:

$$\min_{\boldsymbol{\beta},\beta_0} \sum_{i=1}^{N} (1 - y_i f(\boldsymbol{x}_i))_+ + \frac{\kappa}{2} \|\boldsymbol{\beta}\|^2,$$

can be expressed in terms of the (infinite-dimensional) basis of the expanded feature space:

$$\min_{\boldsymbol{\beta},\beta_0} \sum_{i=1}^{N} \Big(1 - y_i\big(\beta_0 + \sum_{m=1}^{\infty} \theta_m \phi(\boldsymbol{x}_i)\big)\Big)_+ + \frac{\kappa}{2} \sum_{m=1}^{\infty} \frac{\theta_m^2}{\lambda_m},$$

(using $h_m(\boldsymbol{x}) = \frac{\phi_m(\boldsymbol{x})}{a_m}$ and $\theta_m = \frac{1}{a_m}\beta_m$)

κ controls complexity of $\hat{f}$; larger $\kappa \Rightarrow$ smoother $\hat{f}$

SVMs and the Curse of Dimensionality

$$\min_{\boldsymbol{\beta},\beta_0} \sum_{i=1}^{N} \Big(1 - y_i\big(\beta_0 + \sum_{m=1}^{\infty} \theta_m \phi(\boldsymbol{x}_i)\big)\Big)_+ + \frac{\kappa}{2} \sum_{m=1}^{\infty} \frac{\theta_m^2}{\lambda_m},$$

This problem has a finite-dimensional solution under relatively general conditions.

Finding the solution might still be computationally expensive and requires adaptive methods (or substantial prior knowledge)

Summary

- Goal: find function that separates 2 classes
- Maximise separating margin for best generalisation to new data
- Support Vector Classifier separates classes using soft margin
- C parameter controls complexity (smaller C $\Rightarrow$ greater flexibility)
- Further flexibility through non-linear boundaries
- Kernel property (and some mild assumptions) guarantees finite-dimensional solution
- Finding the solution might still be computationally expensive

Thank You

More about SVMs: `http://www.kernel-machines.org`

Intro to RKHS and SVMs: Evgeniou, T., Pontil, M., & Poggio, T. (2000). Regularization Networks and Support Vector Machines. *Advances in Computational Mathematics, 13*(1), 1-50. DOI: 10.1023/A:1018946025316