CMP-6002B - Machine Learning

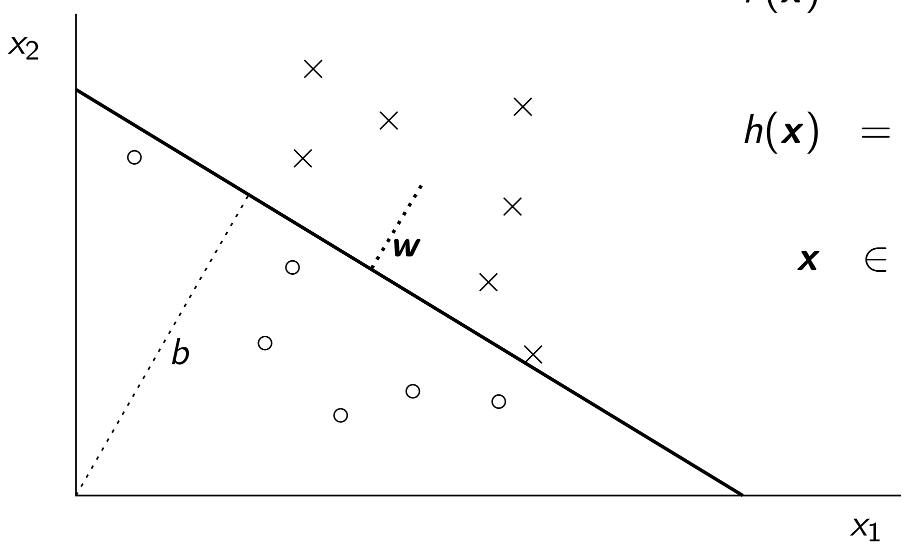
Dr Gavin Cawley

Lecture 6 - Support Vector Machines

Introduction

- Support Vector Machines (Cortes and Vapnik 1995)
  - Statistical pattern recognition method
  - Strong theoretical foundations
  - Demonstrates "state-of-the-art" performance
- Introduce the Support Vector Machine (SVM) method
  - Minimum mathematics (but not "no mathematics" ;-)
  - Emphasise key concepts
- Give overview of aplications
  - Why support vector machines are interesting
  - Applications in computational biology.
  - Applications in computer vision.

Linearly Separable Problems



$$f(\boldsymbol{x}) = \langle \boldsymbol{w} \cdot \boldsymbol{x} \rangle + b$$

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} w_i x_i + b$$

$$h(\boldsymbol{x}) = \mathrm{sign}(f(\boldsymbol{x}))$$

$$\boldsymbol{x} \in \begin{cases} \times & h(\boldsymbol{x}) \geq 0 \\ \circ & h(\boldsymbol{x}) < 0 \end{cases}$$

A Simple Training Algorithm

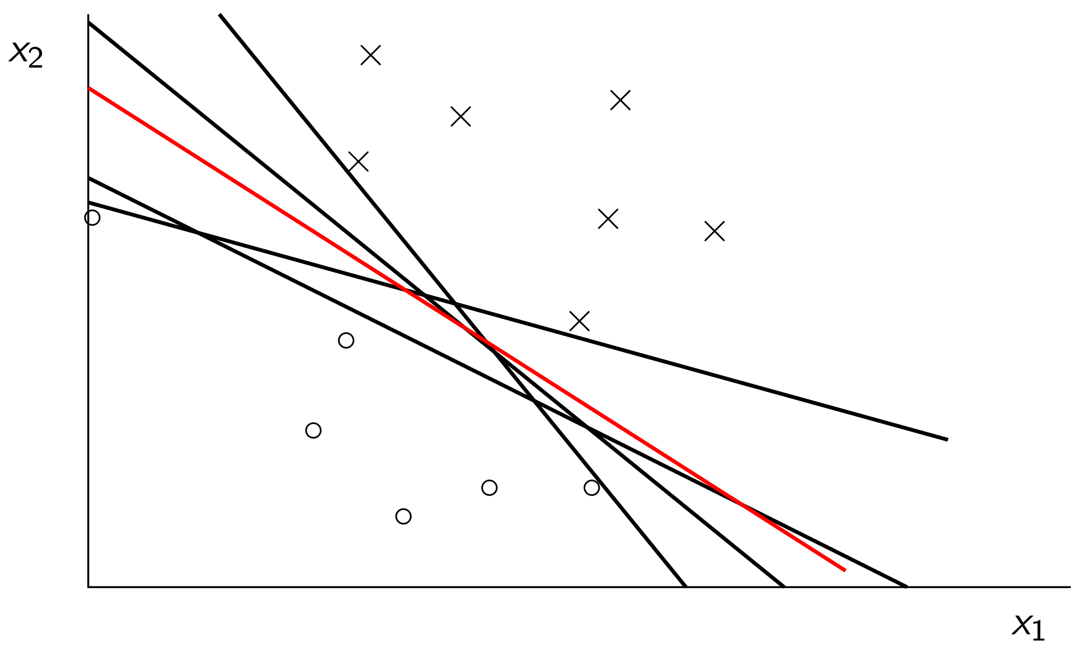
- Let $y_i = -1$ if $\boldsymbol{x}_i \in \times$ and $y_i = +1$ if $\boldsymbol{x}_i \in \circ$
- The Perceptron rule (no bias):

  `repeat`
      `for` $i = 1$ to ℓ
          `if` $y_i \langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle \leq 0$ `then`
              $\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta y_i \boldsymbol{x}_i$
          `end if`
      `end for`
  `until` *all patterns classified correctly*
- Convergence guaranteed for linearly separable problems

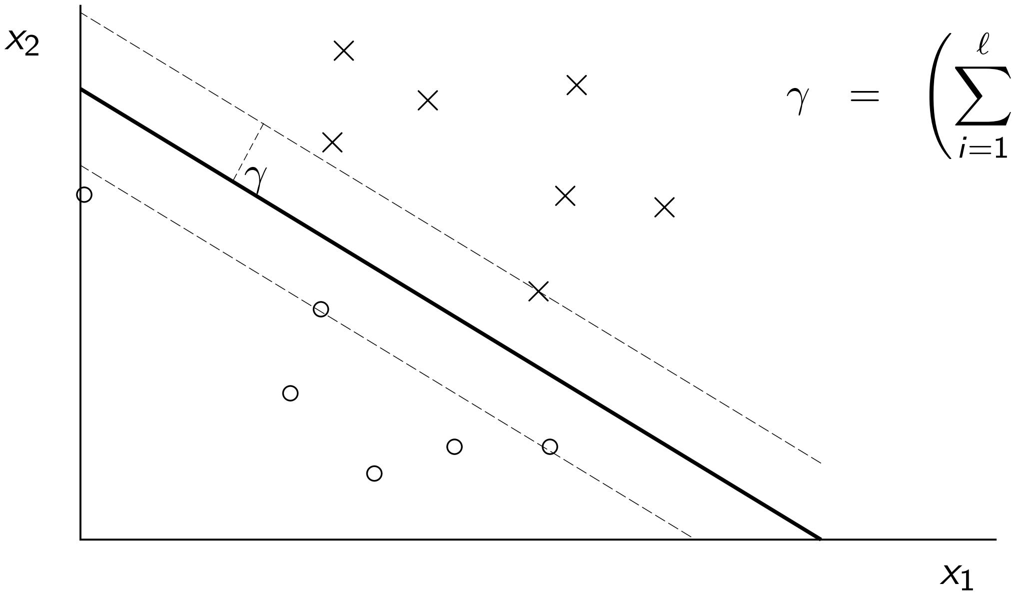
A Dual Perceptron Training Algorithm

- Weight vector is a linear combination of training patterns
  $$\boldsymbol{w} = \sum_{i=1}^{\ell} \alpha_i y_i \boldsymbol{x}_i \quad \rightarrow \quad f(\boldsymbol{x}) = \sum_{i=1}^{\ell} \alpha_i y_i \langle \boldsymbol{x}_i \cdot \boldsymbol{x} \rangle + b$$
- So... `repeat`
  - `for` $i = 1$ to ℓ
    - `if` $y_i \sum_{j=1}^{\ell} \alpha_j y_j \langle \boldsymbol{x}_j \cdot \boldsymbol{x}_i \rangle \leq 0$ `then`
      - $\alpha_i \leftarrow \alpha_i + \eta$
    - `end if`
  - `end for`
  - `until` *all patterns classified correctly*
- We can often write a training algorithm in two ways:
  - Primal - one parameter for each attribute
  - Dual - one parameter for each training pattern

What Makes a Good Decision Rule?



Maximum Margin Classifiers



$$\gamma = \frac{1}{\|\boldsymbol{w}\|}$$

$$\gamma = \left(\sum_{i=1}^{\ell} \alpha_i^2\right)^{-\frac{1}{2}}$$

A Result from Computational Learning Theory

Theorem (Error Bounds for Linear Classifiers)

Define the class F of real-valued functions on the ball of radius R. There is a constant c such that, for any probability distribution on $\mathcal{X} \times \{-1, +1\}$, with probability at least $1 - \delta$ over ℓ identically generated training examples, every $\gamma > 0$ and every function $f \in F$ with margin at least γ on all training examples, then

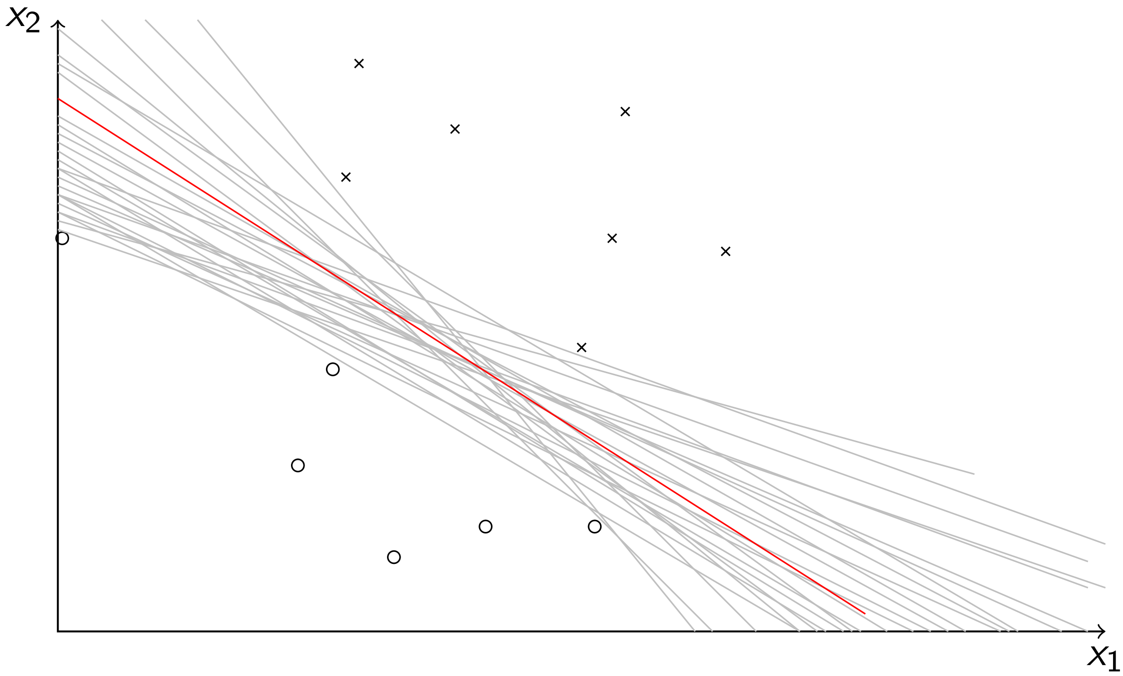
$$\mathrm{err}(f) \leq \frac{c}{\ell}\left\{\frac{R^2}{\gamma^2}\log^2\left(\frac{\ell}{\gamma}\right) + \log\left(\frac{1}{\delta}\right)\right\},$$

where $\mathrm{err}(f)$ is the expected test error rate for f.

- Maximum margin approach justified by "worst case" analysis
  - Any sensible approach will work on easy problems
  - Maximum margin should be safe for awkward problems

An Intuitive Bayesian Justification

- Take the average of all possible models (marginalisation)



Putting Theory into Practice

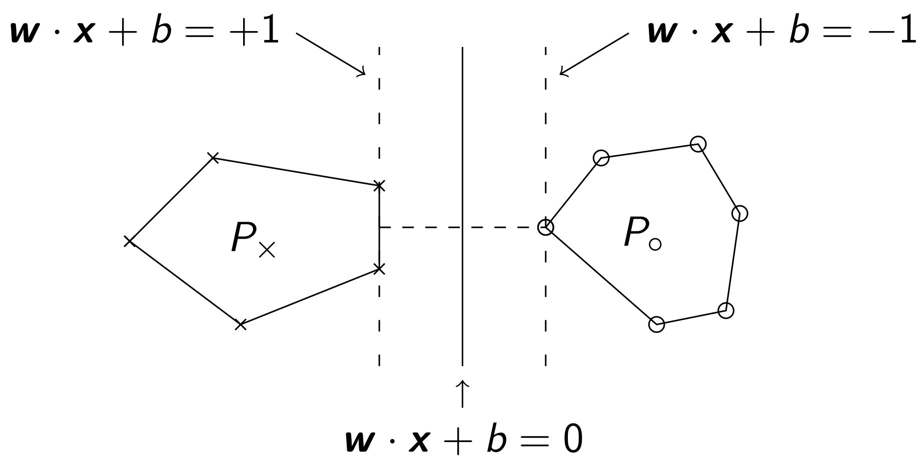
- Primal optimisation problem (in terms of $\boldsymbol{w}$ rather than $\boldsymbol{\alpha}$):
  minimise
  $$\mathcal{L}(\boldsymbol{w}) = \|\boldsymbol{w}\|^2$$
  subject to
  $$y_i(\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b) \geq 1, \quad \forall\ i \in 1, 2, \ldots, \ell$$
- Directly maximise the margin
  - Equivalently minimise squared norm of the weight vector $\boldsymbol{w}$
- Constraints ensure all data lie on or outside the margins
- Quadratic optimisation problem with bound constraints
  - Single global minimum
  - Can find the optimal model parameters exactly (unlike e.g. neural nets!)

A Dual Training Algorithm

- Use of Lagrange multipliers gives the dual optimisation problem:
  minimise
  $$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$$
  subject to
  $$\alpha_i \geq 0 \quad \forall\ i \in 1, 2, \ldots, \ell \quad \text{and} \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0$$
- Quadratic program with bound and linear equality constraints
  - Efficient algorithms are available (e.g. interior points)
- The Karush-Kuhn-Tucker conditions show the solution is sparse
  $\alpha_i = 0 \quad \rightarrow \quad y_i(\langle \mathbf{w} \cdot \mathbf{x}_i)\rangle + b) \geq 1, \quad \text{where} \quad \mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}_i$
- This is known as the "Hard Margin Support Vector Machine"

Sparsity : A Geometric Interpretation

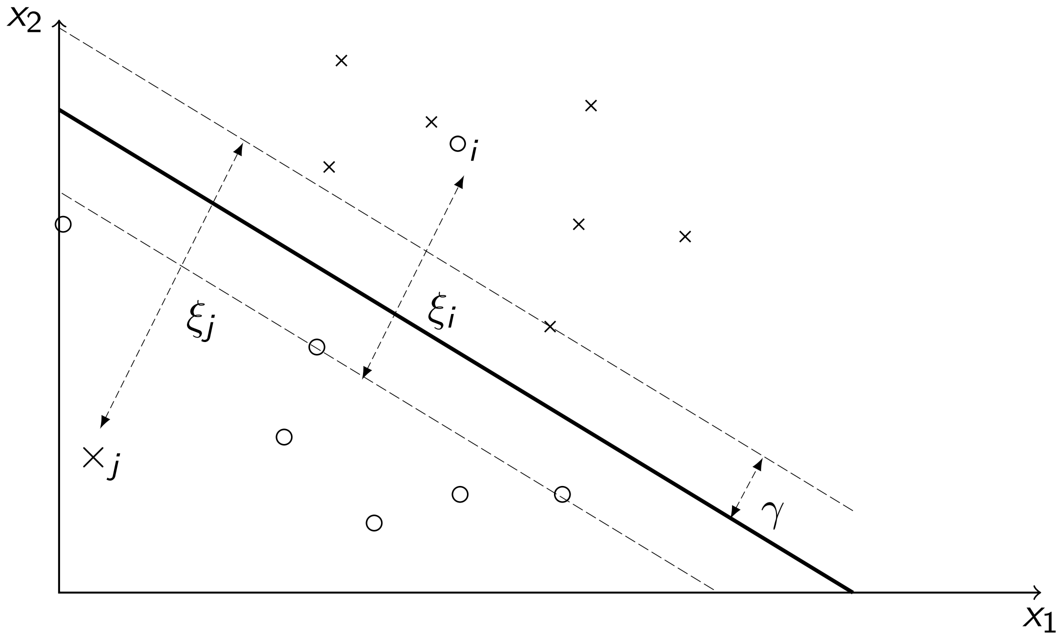
- Decision rule defined by the closest points on two convex hulls



- Not all training patterns define closest points!
  - Those that do are known as "Support Vectors"

Dealing with Non-Linearly Separable Problems

- Introduce "slack" variables, $\boldsymbol{\xi}$, to allow misclassification errors



More Computational Learning Theory!

Theorem (Error Bounds for Linear Classifiers)

Define the class F of real-valued functions on the ball of radius R. There is a constant c such that, for any probability distribution on $\mathcal{X} \times \{-1, +1\}$, with probability at least $1 - \delta$ over ℓ identically generated training examples, every $\gamma > 0$ and every function $f \in F$ with margin at least γ on all training examples, then

$$\mathrm{err}(f) \leq \frac{c}{\ell} \left\{ \frac{R^2 + \|\boldsymbol{\xi}\|_1^2 \log(1/\gamma)}{\gamma^2} \log^2(\ell) + \log\left(\frac{1}{\delta}\right) \right\},$$

where $\boldsymbol{\xi}$ is the margin slack vector with respect to f and γ.

- Can improve generalisation by
  - Maximising the width of the margin γ
  - Reducing magnitude of margin slack variables $\|\boldsymbol{\xi}\|_1$

Soft Margin Support Vector Machines

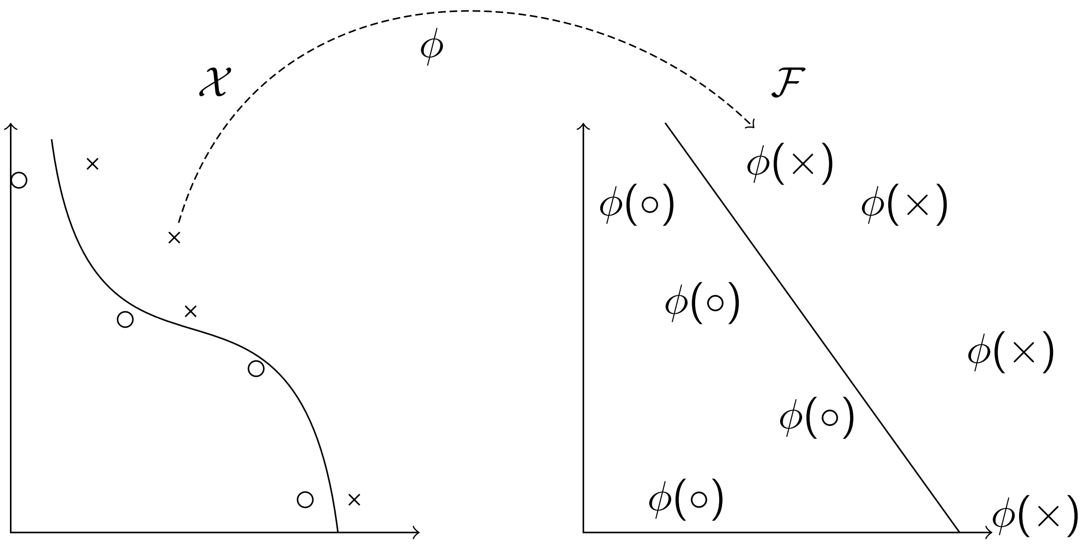
- Minimise combination of norm of weights and slack variables:
  minimise
  $$\mathcal{L}(\boldsymbol{w}) = \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{\ell} \xi_i$$
  subject to
  $$y_i(\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b) \geq 1 \quad \forall\, j \in 1, 2, \ldots, \ell$$
  and
  $$\xi_j \geq 0 \quad \forall\, j \in 1, 2, \ldots, \ell$$
- "Soft Margin" as slacks allow margin constraint to be violated
- C is a regularisation parameter
  - Small C - concentrate on maximising the margin
  - Large C - concentrate on reducing the slacks on training set
  - Best value somewhere in the middle (bias-variance tradeoff)

A Dual Training Algorithm

- Use of Lagrange multipliers gives the dual optimisation problem:
  minimise
  $$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i \cdot \boldsymbol{x}_j \rangle$$
  subject to
  $$0 \leq \alpha_i \leq C \quad \forall\, i \in 1, 2, \ldots, \ell \quad \text{and} \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0$$
- Optimisation problem virtually unchanged
  - Bound constraints replaced by a box constraint
  - Efficient algorithms still exist
  - Still has a single global minimum
  - Solution still sparse

Forming a Non-Linear Decision Rule

- Fixed transformation from input space $\mathcal{X}$ to a "feature space" $\mathcal{F}$
- Linear model in $\mathcal{F}$ corresponds to a non-linear model in $\mathcal{X}$



The "Kernel Trick"

- Project data, $\{\boldsymbol{x}_i \in \mathcal{X}\}_{i=1}^{\ell}$, into a feature space $\mathcal{F}(\boldsymbol{\phi} : \mathcal{X} \to \mathcal{F})$
  - Construct SVM in $\mathcal{F}$ rather than $\mathcal{X}$
  - If $\mathcal{F}$ high dimensional, data is likely to be linearly separable
- Kernel function defines inner product $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{\phi}(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{x}') \rangle$
  - Dual algorithms use data only in the form of inner products
  - Substitute $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for $\langle \boldsymbol{x}_i \cdot \boldsymbol{x}_j \rangle$
  - No need to evaluate $\boldsymbol{\phi}(\boldsymbol{x}_i)$ explicitly
- Not a kernels to give rise to valid feature spaces
  - Must obey Mercer's condition
  - Must have a semi-positive definite Gram matrix

$$\boldsymbol{K} = [k_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^{\ell} \quad \to \quad \boldsymbol{v}'\boldsymbol{K}\boldsymbol{v} \geq 0 \quad \forall\ \boldsymbol{v}$$

Common Kernel Functions

- Homogeneous polynomial - $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = (\langle \boldsymbol{x} \cdot \boldsymbol{x}' \rangle)^d$
  - $\mathcal{F}$ consists of all monomials of order d
  - For 2 input variables and $d = 2$ then $\phi(\boldsymbol{x}) = \left(x_1^2, x_2^2, \sqrt{2}x_1x_2\right)$
  - For 256 input variables and $d = 5$ then $\mathcal{F} \subset \mathbb{R}^{\approx 10^{10}}$
- Inhomogeneous polynomial - $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = (\langle \boldsymbol{x} \cdot \boldsymbol{x}' \rangle + 1)^d$
  - $\mathcal{F}$ contains all monomials of order d or less
- Hyperbolic tangent - $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \tanh(\kappa \langle \boldsymbol{x} \cdot \boldsymbol{x}' \rangle + \theta)$
  - Only for some values of κ and θ
- Radial Basis Function (RBF) - $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left\{-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{\sigma^2}\right\}$
  - $\mathcal{F}$ has an infinite number of dimensions!
  - SVM is then always able to reduce training set error to zero

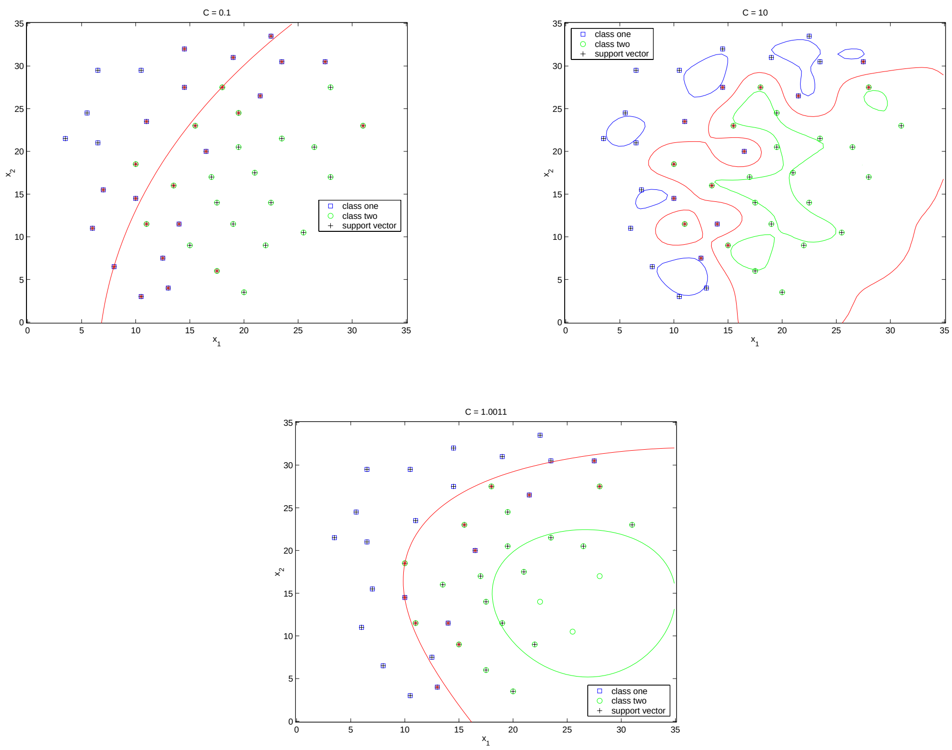
Putting all the Pieces Together

- Using the "kernel trick" gives the dual optimisation problem: minimise
  $$\mathcal{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} \alpha_i\alpha_j y_i y_j \mathcal{K}(\boldsymbol{x}_i \cdot \boldsymbol{x}_j)$$
  subject to
  $$0 \leq \alpha_i \leq C \quad \forall\, i \in 1, 2, \ldots, \ell \quad \text{and} \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0$$
- Optimisation problem essentially unchanged
- Also due to duality:
  $$\boldsymbol{w} = \sum_{i=1}^{\ell} \alpha_i y_i \phi(\boldsymbol{x}_i)$$
  and
  $$f(\boldsymbol{x}) = \sum_{i=1}^{\ell} \alpha_i y_i \langle \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}) \rangle + b = \sum_{i=1}^{\ell} \alpha_i y_i \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

Key Concepts

- Maximal margin classification
  - Makes good practical sense
  - Agrees with Bayesian intuition
  - Minimises an upper bound on test error rate
- Based on constrained quadratic optimisation
  - Efficient algorithms available
  - Guarantee of single global optimum
- Use of "slack" variables to deal with non-separable problems
  - Retains strong theoretical justification
- Powerful non-linear classifier via the "kernel trick"
  - Retains theoretical/practical benefits of linear SVM

A Toy Example



A Real-World Example

MNIST handwritten character benchmark (60000 training & 10000 test examples, 28×28)

Classifier	Test Error	Reference
Linear Classifier	8.4%	Bottou *et al.*, 1994
3-Nearest Neighbour	2.4%	Bottou *et al.*, 1994
SVM	1.4%	Burges and Schölkopf, 1997
Tangent Distance	1.1%	Simard *et al.*, 1993
LeNet4	1.1%	LeCun *et al.*, 1998
Boosted LeNet4	0.7%	LeCun *et al.*, 1998
Translation Invariant SVM	0.56%	DeCoste and Schölkopf, 2000

Note: the SVM used a polynomial kernel of degree 9, corresponding to a feature space of dimension $\approx 3.2 \times 10^{20}$.

Model Selection

- Minimise an upper bound on the leave-one-out error.
  - Non-support vectors are never misclassified,

$$E_{\mathrm{loo}} \leq \frac{N_{\mathrm{SV}}}{\ell}.$$

  - The radius-margin bound (much tighter),

$$E_{\mathrm{loo}} \leq \frac{1}{\ell}\frac{R^2}{\gamma^2}.$$

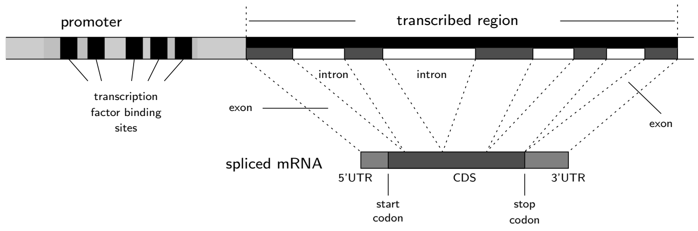
  - The Span bound (tighter still),

$$E_{\mathrm{loo}} \leq \frac{1}{\ell}\sum_{p=1}^{\ell}\Psi\left(\alpha_p^0 S_p^2 - 1\right),$$

where $\Psi(\cdot)$ is the unit step function and

$$S_p = \min_{\lambda \in \Lambda_p}\left(\|\phi(\boldsymbol{x}_i) - \lambda\|\right), \quad \Lambda_p = \left\{ \sum_{1 \neq p, \alpha_i^0 > 0} \zeta_i \phi(\boldsymbol{x}_i) \;\middle|\; \sum_{i \neq p} \zeta_i = 1 \right\}.$$

Promoter Based Gene Classification



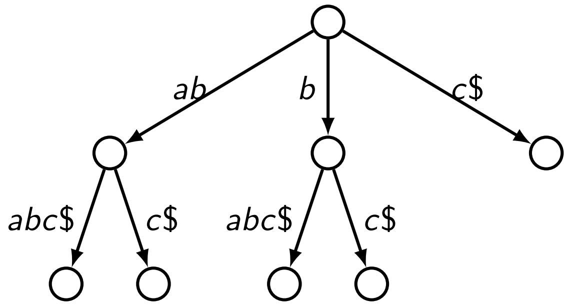
- Every gene is preceded by a promoter region
  - Transcription factors control the *expression* of the gene
- Identify co-regulated genes using microarray data
- Find features distinguishing up- and down- regulated genes
  - PLACE database of known TF binding sites
  - All possible k-mers

Using PLACE Features

- PLACE database contains binding sites for known TFs
- Using linear kernel $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x} \cdot \boldsymbol{x}'$
  - Feature space identical to input space
  - Number of occurrences of motifs better than simple presence/absence of motifs
  - TELOBOX motif implicated in glucose responsive expression.
- Using polynomial kernel $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x} \cdot \boldsymbol{x}' + 1)^d$
  - Feature space is space of all monomials of order d or less
  - Features correspond to combinations of features
  - $\approx$ 280 place motifs, $d = 3 \implies \approx$ 22 million features!
  - Combinations of motifs did not significantly improve performance.

The Spectrum Kernel

- Feature space consists of the number of occurrences of all possible substrings of length k.
- Substrings represent the boundary "features" of the shape
- Efficient implementation using *suffix trees*
  - Complexity $\mathcal{O}(kN)$.
  - Strings need not be of the same length.
  - Easily extended to allow *mismatches*.
  - Linear complexity predictions.



Suffix Tree of ababc

- Performance similar to that obtained with PLACE features

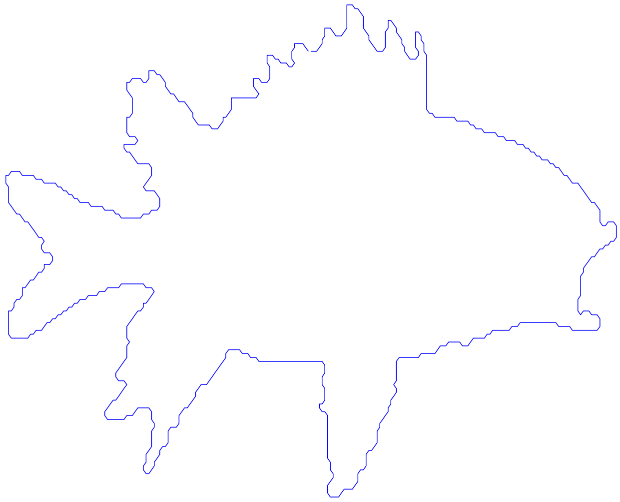
Kernel Methods For Computer Vision

- SVMs have already proved very successful in applications using "unstructured" image data
  - Optical character recognition
  - Face detection
  - Biometrics (e.g. face/fingerprint recognition)
- Traditional classifiers have difficulty with "structured data"
  - Chain-code representations

    representation of shape
    representation of movement e.g. handwriting
  - Trees based representations of objects in images
  - Generative models e.g. Hidden Markov Models (HMMs)

Shape Recognition Via Chain Codes

- Represent a shape by a set of steps to the N, E, S and W

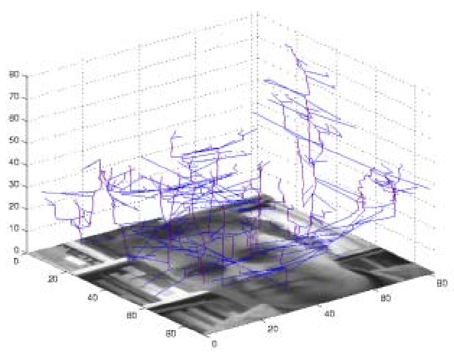


- Shapes become variable length strings from a fixed alphabet

`NEWSWENWWNEEENSNNSNWNWNNENNENWNNSNNE...`

Tree-based Representation of Objects in Images

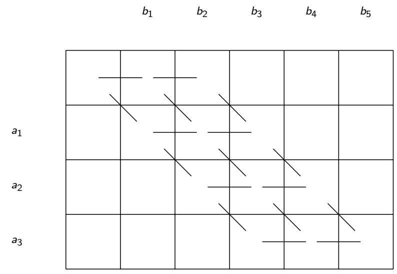
- The sieve can provide a tree relating the components of an object at difference scales
- A tree can be folded to form a string

- Current project

Sequence Alignment Kernel

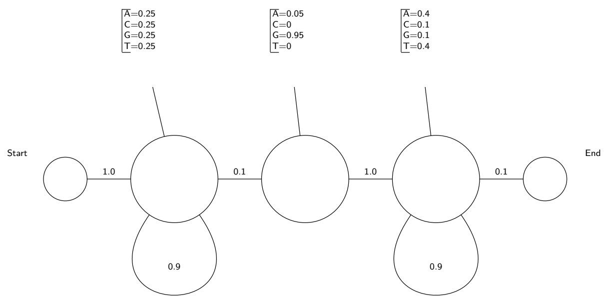
- Smith-Waterman dynamic programming algorithm computes optimal pairwise alignment between sequences
- Optimal alignment cost does not give a valid kernel :-(



- Unless we average over all possible alignments
  - Can easily be computed using dynamic programming
  - Similar sequences will have higher average alignment scores
- Have been used for finding transcription start sites (TSSs)

Exploiting Generative Models

- "Fisher" kernels constructed from a generative model $P(\boldsymbol{x}|\boldsymbol{\theta})$
- Hidden Markov Models
- $\boldsymbol{\theta}$ represents transition/emission probabilities of each state



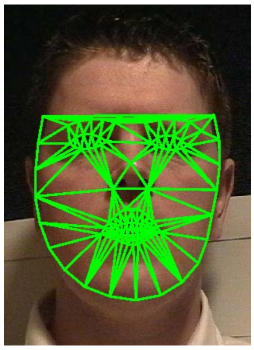
- Feature space defined by the *Fisher Score* vector $\boldsymbol{u_x}$

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{u}_{\boldsymbol{x}}^T \boldsymbol{I}^{-1} \boldsymbol{u}_{\boldsymbol{x}'}, \quad \boldsymbol{I} = \mathcal{E}\left\{\boldsymbol{u}_{\boldsymbol{x}} \cdot \boldsymbol{u}_{\boldsymbol{x}'}^T\right\}, \quad \boldsymbol{u}_{\boldsymbol{x}} = \nabla_{\boldsymbol{\theta}} \log P(\boldsymbol{x}|\boldsymbol{\theta})$$

  - Often ignore Fisher information matrix, I.
- Similarity metric accounting for the distribution of the data

Generative Models in Computer Vision

- "Fisher" kernels constructed from a generative model $P(\boldsymbol{x}|\boldsymbol{\theta})$
  - Active appearance model,
    - Face recognition
  - Hidden Markov Models,
    - Lip-reading
    - Gesture recognition
    - Gait recognition



- Again, use feature space defined by Fisher kernel.
- Similarity metric accounting for the distribution of the data

Summary

- Brief review of basic theory of Support Vector Machines
- Why should SVMs be interesting?
  - State-of-the-art performance on many real-world problems
  - Strong theoretical justification
  - Efficient large-scale training algorithms
  - Can cope with "awkward" representations of image data
- There are related approaches to other settings
  - Regression (e.g. Relevance Vector Machine)
  - Clustering/Segmentation (similar to the Normalised Cuts)
  - Non-linear principal/independent component analysis