An introduction and overview of the principles of Machine Learning (Day 1)

Rene Eijkemans

Department of Biostatistics and Research Support
Julius Center, UMC Utrecht
The Netherlands

November, 2019

Materials for Day 1

Books:

- An Introduction to Statistical Learning – PDF (7th printing) can be downloaded at ISL*
- The Elements of Statistical Learning: Data Mining, Inference, and Prediction – PDF (12th printing) can be donwloaded at ESL*

Data for the exercises:

- Data1.zip
- Datasets within R-packages

And slides (available on the website), wikipedia, R-bloggers, etc.

Overview

- General introduction
- Supervised learning: classification and regression
- Classification and regression in High Dimensional data
- Probabilistic prediction versus hard classification
- Introduction to Support Vector Machines

Outline

What is Machine Learning?

- learning plays a key role in the fields of statistics, data mining, pattern recognition and artificial intelligence
- Many problems cannot be solved by:
  - scientific theories alone
  - programming logical 'rules', such as: If (A and B and not C) do X
- Humans learn by looking at examples, without detailed 'rules' on how and where to look.
- Can computers do the same thing?

Examples

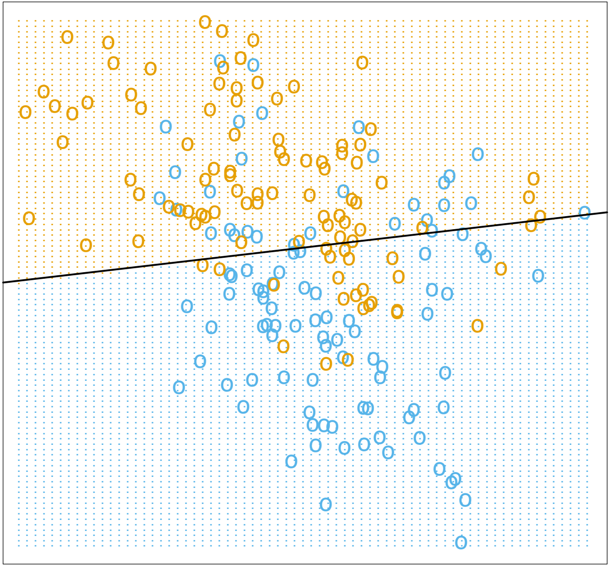
- Google Translate
- Speech recognition
- Credit approval by banks
- Spamfiltering
- Self driving cars https://vimeo.com/106226560
- Medical diagnoses
- Predictive modelling in Medicine
- ?

Characteristics of a learning problem

ML is applicable to a problem when:

- There is a pattern
- The pattern cannot be described well theoretically
- We have data to learn from

Linear Regression of 0/1 Response



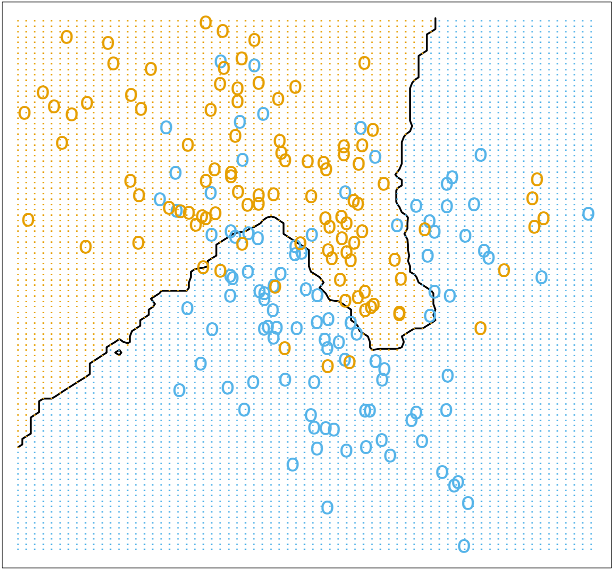
Algorithm: linear model

- The algorithm only works well on linear separable data
- But it searches an infinitely large class of models!

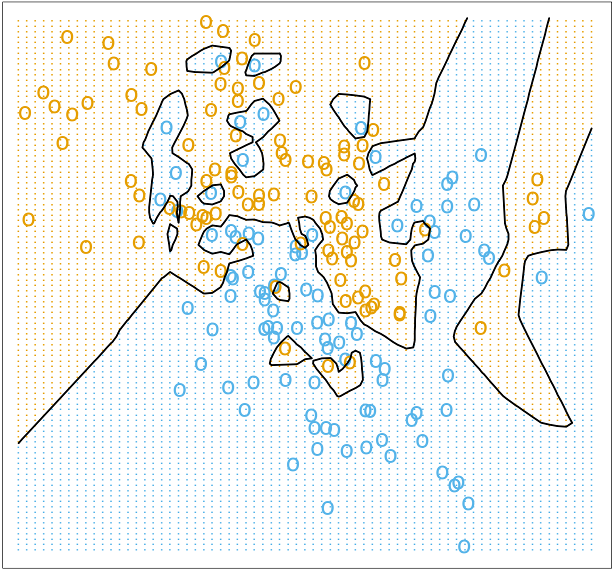
Algorithm: Nearest Neighbours

- Another combination of model space and algorithm is Nearest Neighbours
- The model space consists of (almost) all functions from inputs to outputs!
- The algorithm is very simple:
  Given a new input, find the nearest inputs from the learning data, and use their outputs

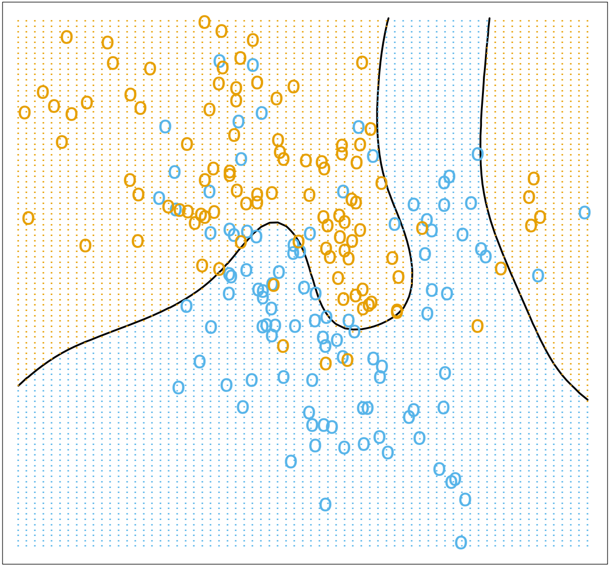
15-Nearest Neighbor Classifier

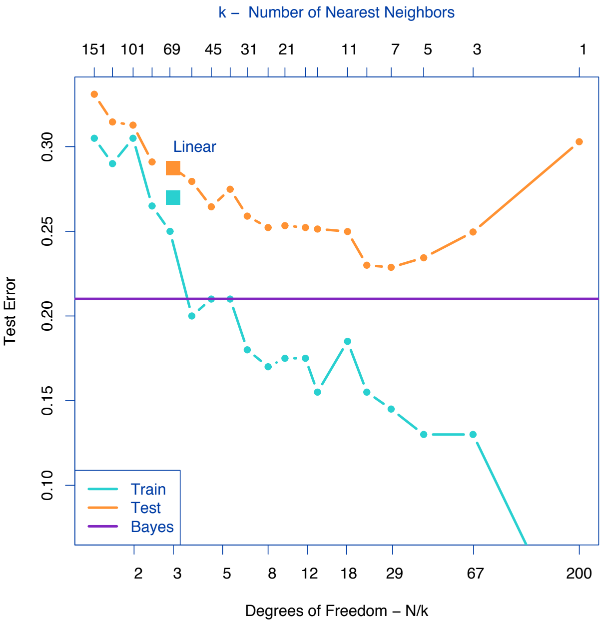


1-Nearest Neighbor Classifier



Bayes Optimal Classifier




k – Number of Nearest Neighbors
151 101 69 45 31 21 11 7 5 3 1
Linear
Test Error
0.30
0.25
0.20
0.15
0.10
Train
Test
Bayes
2 3 5 8 12 18 29 67 200
Degrees of Freedom – N/k

Example of supervised learning: kNN vs Linear

- kNN:
  - Training error become smaller with lower k (higher N/k)
  - The distance between training error and test error becomes larger with lower k: *more Overfitting*
  - There is an optimal value for k giving the lowest test error
- Linear model:
  - Training error only slightly lower than test error: *Not much overfitting*
  - Test error is higher than optimal kNN

Example of supervised learning: kNN vs Linear

- It seems that kNN is a 'super' method!
- due to it's local character, it can automatically find any separating boundary. In the training data, 1NN is almost as good as the perfect (error=0) model

BUT:

- kNN breaks down in higher dimensions (p), the 'curse of dimensionality', for several reasons:
- The expected distance per dimension needed for a certain fraction r of the data grows with the dimensionality p:

$$e_p(r) = r^{1/p}$$

  $e_2(.01) = 0.1$, $e_{10}(.01) = 0.63$, $e_{20}(.01) = 0.79$
- More points are close to the edge of the sample

Some enhancements of linear models

- Kernel methods $<>$ smoothing functions
- Local regression, e.g. Loess smoother
- Linear models on basis expansions of the inputs x
- Projection pursuit and neural networks: sums of non-linearly transformed linear models

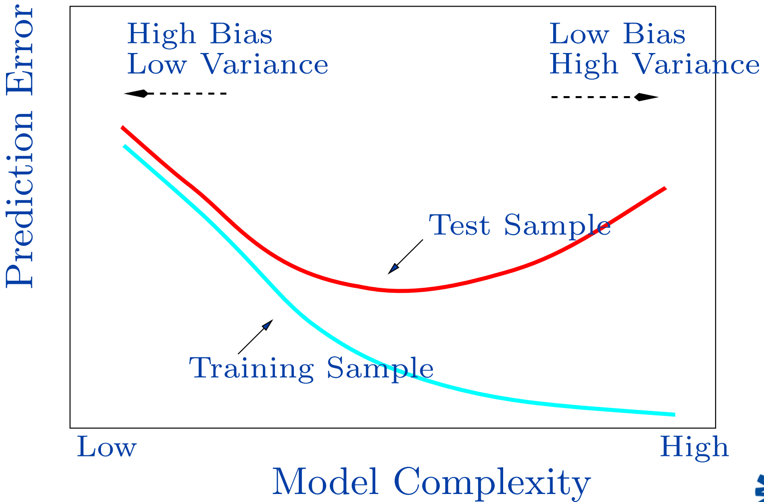
Caveats of model enhancements

- Introducing kernels etc increases model complexity
- The complexity as 1-Nearest Neighbours can be approached

BUT:

- More complex models are prone to overfit to noise in the data
- Overfitted models have poor generalisation

Bias-variance trade-off



Countermeasures to overfitting

- Models that have inherent large complexity are prone to overfitting
- Regularisation is a way to counter overfitting
- Standard method of regularisation: penalisation
- Examples:
  - Ridge regression
  - Lasso
  - Elastic net

Countermeasures to overfitting

Other ways to control model complexity:

- The width of kernels determines the degree of smoothness and thereby model complexity
- The number of basis function in basis expansions
- The number of neighbours k in k-Nearest Neighbours

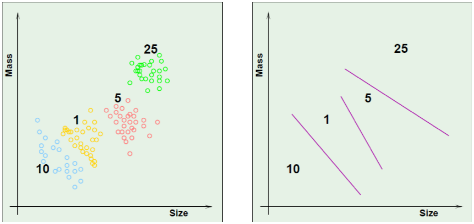
Countermeasures to overfitting

- In all these methods, a 'smoothness' or 'tuning' parameter has to be determined
- If this is done on the training data, there still is a risk of overfitting
- Solution is to use Cross-validation to find the optimal value of the tuning parameter. More about this tomorrow.

Types of learning problems

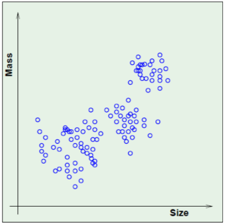
- Supervised learning

  (x, correct output y)



- Unsupervised learning

  (x)



Types of learning problems

- Active learning:
  - Learn iteratively choosing the optimal x in each iteration
- Online learning
  - Learning with x values coming in one at a time
- Reinforcement learning
  - Training data: (x, some output, grade for this output)
- Transfer learning
  - knowledge gained while solving one problem applied to a different but related problem
- ?

Outline

Supervised learning: classification and regression

Data are generated by $y \sim f(X)$. f is the unknown target function

- X is the matrix of features
- y is the dependent variable
  - Binary: (hard) classification or (probabilistic) prediction
  - Continuous: regression
- y multi-categorical: extension of binary hard classification: one-to-one binary classifications followed by majority voting

Supervised learning: Examples in the medical field

Concerning the dependent variable y

- Prognosis of patients
- Response to treatment
- Diagnosis

Concerning the features X

- Clinical variables
- 'omics' (high-dimensional) data
- Wearable device/sensor data
- Images (e.g. CT scans)

Supervised learning: the dilemma

Two competing goals:

- Perform well in all kinds of difficult problems: rich and flexible model class desirable (kNN)
- Overfitting should be minimised: simple model class is better (linear model)

- Many clever methods have been devised that each in their own way find a balance between these two goals

Supervised learning: the dilemma

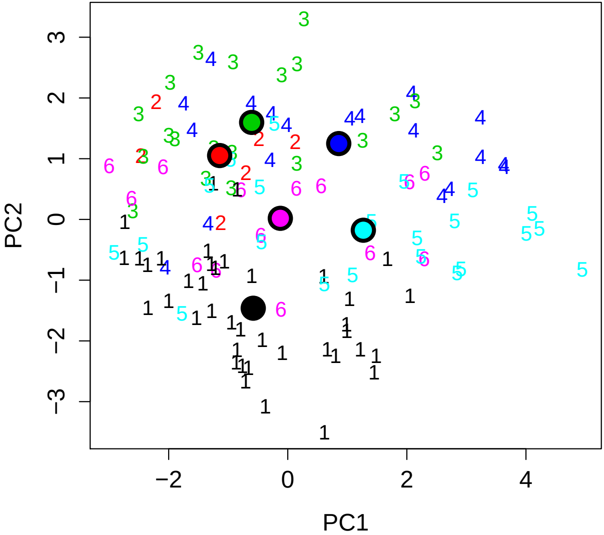
The choice of method, and the balance, depend on

- sample size (N)
- number(p) of features
- amount of structure in X (images!)
- difficulty of the problem

Supervised learning: Example Autism data

- Study of 322 subjects with six genetic symdromes associated with Autism Spectrum Disorder (ASD)
- Features (X): subset of 34 Items from the autism diagnostic interview-revised (ADI-R) interview.
- Scores per item: 0-1-2
- Research question: Do different genetic groups have different symptom profiles?
- First: exploratory analysis. Principle Components Analysis (PCA) plot, labeling subjects by genetic group. More about PCA tomorrow

PCA in study on autism. Coloring by Genetic groups 1 to 6



Supervised learning: Example Autism data

- As an example, binary classification of two of the six genetic syndromes.
- First: easy separation (according to the PCA plot): group 1 (22q11DS deletion, n=90) vs group 4
- (Supernumerary Marker chromosome 15, SMC, n=22)
- Since the smallest group (SMC) has fewer subjects (n=22) than the number of features (p=34) we are
- already in a kind of high-dimensional setting
- Therefore, use PC1 and PC2 as sole features

Outline

R Exercise 1.1: Supervised learning

- File names: Exercise1-1.nb.html, and Exercise1-1.Rmd (continued)
- The Genetic Syndrome Autism data
- Logistic regression for the group 1 vs 4 contrast on PC1 and PC2
- 1NN as an alternative approach
- Using cross validation to estimate the out-of-sample errors
- Questions:
  - Optimise k in kNN
  - Repeat the exercise for the contrast group 1 vs 5

Outline

General introduction

Supervised learning: classification and regression

Exercise 1.1: Supervised learning

Classification and regression in High-Dimensional Data

Probabilistic prediction versus hard classification

Introduction to Support Vector Machines

Exercise 1.2: Support Vector Machines

High-dimensional data

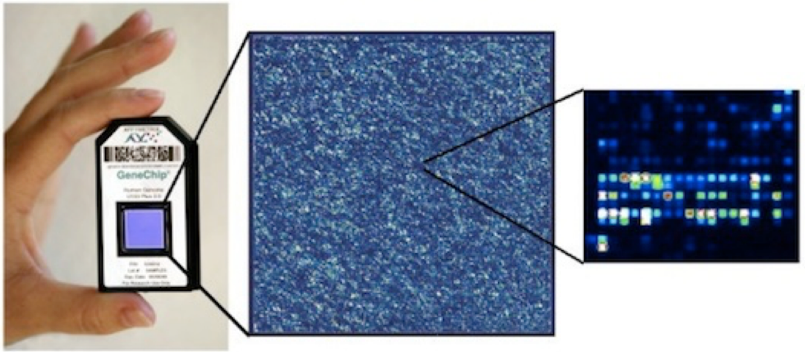
High dimensional data: $p \gg N$

Examples:

- All kinds of high-throughput 'omics' data
  - Gene expression (microarray or RNA Sequencing)
  - Proteomics
  - Methylation array data
  - Metabolomics
- Imaging data
  - X-ray
  - CT scan
  - (functional) MRI

High-dimensional data: Example microarray

Microarrays for gene expression (RNA) analysis



Typical size: 64,000 probesets per subject

Classification in high-dimensional data

- High dimensional data: $p \gg N$
- We are beyond the curse of dimensionality: we have complete separation $=>$ Even the simplest linear model will overfit
- Classification is possible through either
  - Combining classification method with feature selection method, to reduce number of features (p)
  - Only methods that have in-built protection against overfitting can be used

Complete separation in binary classification

- In 1 dimension (p=1), we can always separate 2 points
  - Find a point in between the two points
- In 2 dimensions (p=2), we can always separate max 3 points
  - Find a line with two points on one side and one point on the other side
- In 3 dimensions (p=3), we can always separate max 4 points
  - Find a 2-D plane in 3-D space that separates either 2 vs 2 or 1 vs 3

- In p=64,000 we can easily find a 64,000-1 D hyperplane that separates up to 64,000+1 points

Methods for classification in high-dimensional data

Most used classification methods that can handle high dimensional data, without preselection of features

- Regularization methods:
  - Ridge regression, Lasso, Elastic Net
- Tree based methods:
  - Bagging, Random Forest, Boosting
- Nearest Shrunken Centroids
- Supervised Principle Components/Partial least Squares (PLS)
- Support Vector Machines (SVM)

Outline

Difference between hard classification and probabilistic prediction

Hard classification:

- For automated classification tasks: e.g. read handwritten ZIP codes, self-driving cars, etc.
- In medical applications? Automated pattern recognition on medical images in radiology or pathology.
- Among competing classification algorithms: choose the one with highest accuracy/lowest error.

Probabilistic prediction:

- Produces a probability estimate that a subject belongs to a certain class
- In medical applications?

Probabilistic prediction in medicine

Clinical applications usually deal with predicting the prognosis of patients.

- Doctors typically use benefit-risk reasoning to decide on treatment:
- Only when the probability of death of a patient exceeds a certain threshold, the risk of a dangerous operation that could cure the patient becomes acceptable.
- It is of key importance that the predicted probability of death, both in the case of no operation, and the probability of death due to the operation are estimated reliably.
- A prediction model that produces reliable predicted probabilities is called well-calibrated.

Probabilistic prediction in medicine

How to choose the best probabilistic prediction algorithm?

- When choosing between alternative algorithms:
- Choose among the well calibrated ones, the one with the best discriminative ability.
- Discrimination can be measured by the Area under the ROC curve (AUC) aka c-statistic.

Probabilistic prediction

Many machine learning algorithms can either directly, or indirectly be used for probabilistic prediction.
Direct methods:

- Logistic regression, as part of the family of generalised linear models, produces probabilities as their prime output
- Regularised Logistic regression (see tomorrow's lecture)
- All tree based methods (CART, Random Forests, Boosting) can produce probabilities

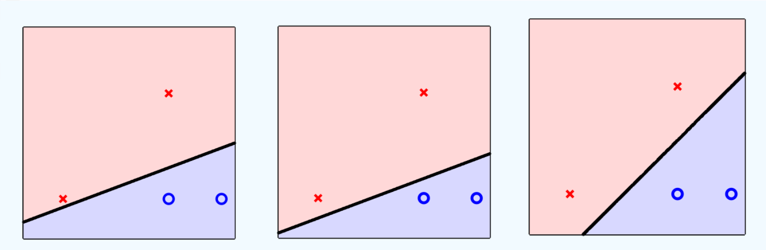
Indirect methods:

- Support Vector Machines, allow posthoc fitting of a logistic model on decision functions that give the best seperation
- All other methods (e.g. NeuralNetworks) that can be fitted using an alternative loss function:
- use Maximum likelihood(on binomial likelihood) instead of maximising accuracy.

Outline

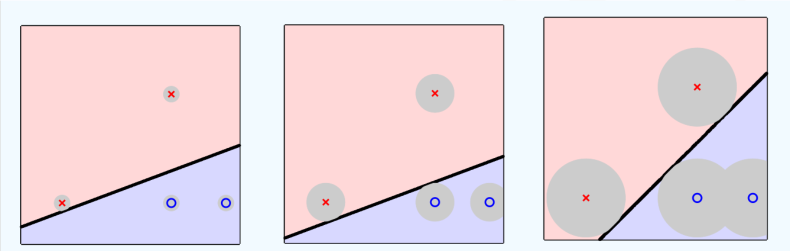
Support vector Machines

Which linear classifier is better?



Support vector Machines

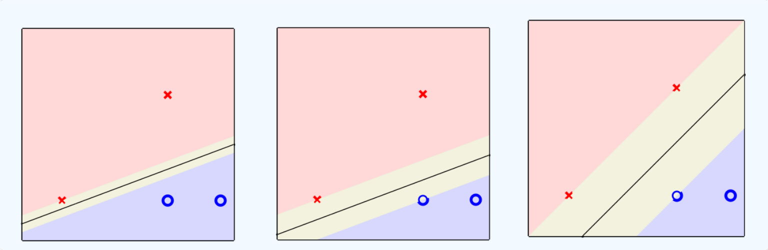
Why is the right one the best one? Think of noise in the x-values:



Clearly, the right one is more robust to noise, and therefore will generalize better

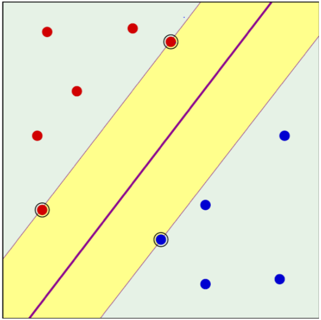
Support Vector Machines: Fat margin

Equivalently: from the perspective of the separator:



Support vectors

- The data points that determine the fattest margin are called Support Vectors.
- Note that the solution only depends on these support vectors, the rest of the data doesn't play a role!



Importance of Support vectors

Thought experiment:

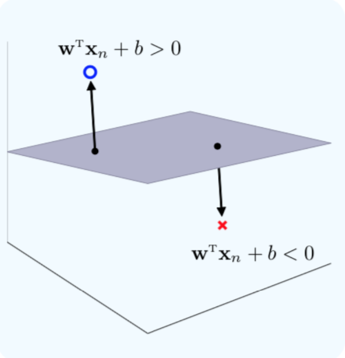
- Determine the Leave-One-Out Cross Validation (LOOCV) Error
- Only when one of the Support Vectors is left out, will the optimal separator change, with possible misclassification of this left out subject.
- Otherwise it will not change and the left-out point remains correctly classified.
- The LOOCV error is therefore at most #Support Vectors/N
- This is a good upper bound on the out-of-sample error
- When we add non-linear transformations to the data, the #Support Vectors grows more slowly than the number of dimensions added, making SVM a very robust method.

Optimizing the margin

The linear separating (hyper)plane can be expressed as

$$\mathbf{w}^T\mathbf{x} + b = 0$$

Observed datapoints $\mathbf{x_n}$ have $\mathbf{w}^T\mathbf{x_n} + b > 0$ on one side and $\mathbf{w}^T\mathbf{x_n} + b < 0$ on the other side of the separating hyperplane



Optimizing the margin(2)

If we recode y_n as $\{-1, +1\}$, concurrent of the sign of $\mathbf{w}^T\mathbf{x_n} + b$, we find that

$$y_n(\mathbf{w}^T\mathbf{x_n} + b) > 0, \text{ for } n = 1, ..., N$$

- The separating hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ will stay the same if we multiply $y_n(\mathbf{w}^T\mathbf{x_n} + b)$ by any arbitrary number ρ.
- So we can choose ρ such that $y_n(\mathbf{w}^T\mathbf{x_n} + b) = 1$ for the closest point(s) to the hyperplane (Support vectors).
- We can show that $\mathbf{w}$ is a normal vector to the separating hyperplane, and the distance of a point $\mathbf{x_n}$ to the hyperplane is inversely related to the norm of $\mathbf{w}$.

Optimizing the margin(3)

SVM can find the linear separator with the fattest margin by solving a quadratic programming problem. The general formula for a linear separator is:

$$\min_{b,\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$\text{subject to: } y_n(\mathbf{w}^T\mathbf{x_n} + b) \geq 1 \text{ for } n = 1, ..., N$$

- The inequality constraints can be dealt with by introducing Langrangian multipliers α_n.
- In a dual formulation, the quadrating programming becomes a problem of finding the optimal α_n, with equality constraints for $\mathbf{w}$ and b.
- The equations to optimise the α_n, contain the matrix of all inner products between the x-vectors ${x_i}^T \cdot x_j$.

Kernel Trick

- Arbitrary non-linear transformations can automatically be generated by SVM, using the 'Kernel trick'
- Idea is to apply non-linear transformations of the X data, $Z = \phi(X)$, mapping X to a much higher dimensional space Z, e.g. by a polynomial of degree Q=10.
- In the dual quadratic programming formulation in the high dimensional Z-space, all the inner products between the z-vectors,$z_i{}^T \cdot z_j$, would be needed
- This would be problematic because of the potentially very high dimensionality (crossproducts of polynomial terms).

Kernel Trick

- A kernel function $K(x_i, x_j) = (1 + {x_i}^T \cdot x_j)^Q$ can be shown to generate all these terms of the inner product in z-space, avoiding the need to explicitly calculate ${z_i}^T \cdot z_j$.
- The transformation $\phi(X)$ may even be into an infinite dimensional space, without problems.
- Example of this: the rgb kernel $K(x_i, x_j) = exp(-\gamma\|x_i - x_j\|^2)$
- The number of Support vectors in z space may still be limited, guaranteeing robustness of the solution.

Soft margin

SVM can also find an optimal linear separator when the data are not linearly separable due to overlapping classes, by introducing a cost parameter C:

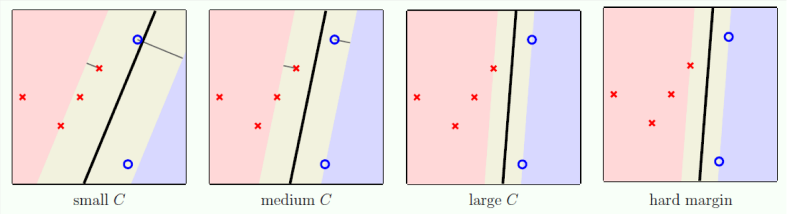
$$\min_{b,\mathbf{w},\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n$$

$$\text{subject to } y_n(\mathbf{w}^T\mathbf{x_n} + b) \geq 1 - \xi_n \quad \xi_n \geq 0 \qquad \text{for } n = 1, ..., N$$

This is still a quadratic programming problem that can easily be solved.

Soft margin SVM

- The C parameter acts as a kind of regularisation parameter
- Small C, much regularisation: the margin has to be very wide, even at the cost of increasing the in-sample error
- Large C, little regularisation: go for minimal in-sample error, even at the cost of a narrow margin



Support Vector Machines

- SVMs are very powerful, and easy to fit
- This made them superior to any other method.
- In Imaging applications, they have been overtaken by Deep Learning
- In High-Dimensional data, SVMs work very well, and are often the best option
- Also when the data do not have complete separation, the #Support Vectors remains limited: only the points that fall within or on the boundary of the Fat margin are Support Vectors

Outline

R Exercise 1.2: Support Vector Machines

- File names: Exercise1-2.nb.html, and Exercise1-2.Rmd (continued)
- Yet again the Genetic Syndrome Autism data
- SVM on the ADI-R items, group 1 vs 5 contrast
- Tune the Cost parameter in the linear kernel model, using LOOCV
- Compare linear kernel with radial (Gaussian) kernel