STK4030 - Statistical Learning: Advanced Regression and Classification

Riccardo De Bin

debin@math.uio.no

Outline of the lecture

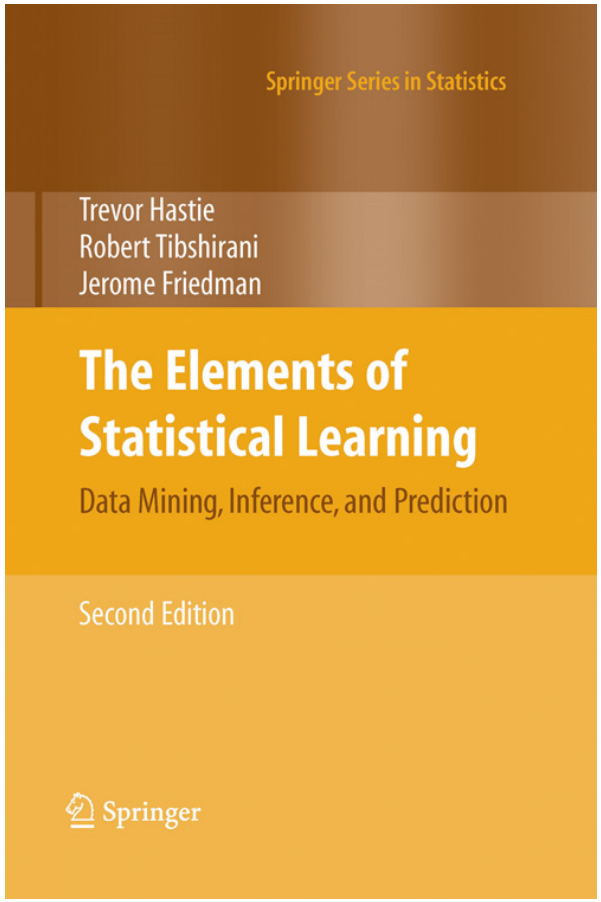
- Introduction

- Overview of supervised learning
  - Variable types and terminology
  - Two simple approaches to prediction: least square and nearest neighbors
  - Statistical decision theory
  - Local methods in high dimensions

Introduction: Elements of Statistical Learning

This course is based on the book:
"The Elements of Statistical Learning: Data Mining, Inference, and Prediction"
by T. Hastie, R. Tibshirani and J. Friedman:

- reference book on modern statistical methods;
- free online version, `https://web.stanford.edu/~hastie/ElemStatLearn/`.



Introduction: statistical learning

"We are drowning in information, but we starved from knowledge"
(J. Naisbitt)

- nowadays a huge quantity of data is continuously collected
  $\Rightarrow$ a lot of information is available;
- we struggle with profitably using it;

The goal of statistical learning is to "get knowledge" from the data, so that the information can be used for prediction, identification, understanding, ...

Introduction: email spam example

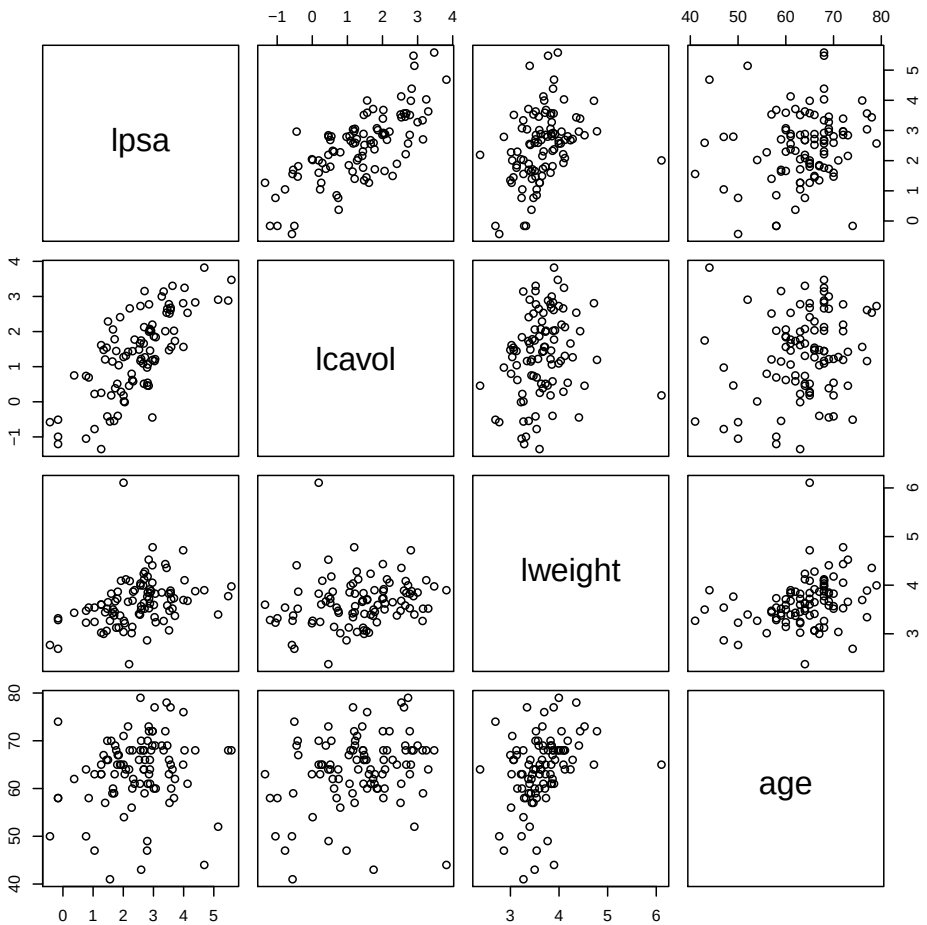
Goal: construct an automatic spam detector that block spam.
Data: information on 4601 emails, in particular,

- whether was it spam (`spam`) or not (`email`);
- the relative frequencies of 57 of the most common words or punctuation marks.

word	george	you	your	hp	free	hpl	!	...
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	...
email	1.27	1.27	0.44	0.90	0.07	0.43	0.11	...

Possible rule: if (%`george` < 0.6) & (%`you` > 1.5) then `spam`
else `email`

Introduction: prostate cancer example



- data from Stamey et al. (1989);
- goal: predict the level of (log) prostate specific antigene (`lpsa`) from some clinical measures, such as log cancer volume (`lcavol`), log prostate weight (`lweight`), age (age), . . . ;
- possible rule: $f(X) = 0.32\ \texttt{lcavol} + 0.15\ \texttt{lweight} + 0.20\ \texttt{age}$

Introduction: handwritten digit recognition

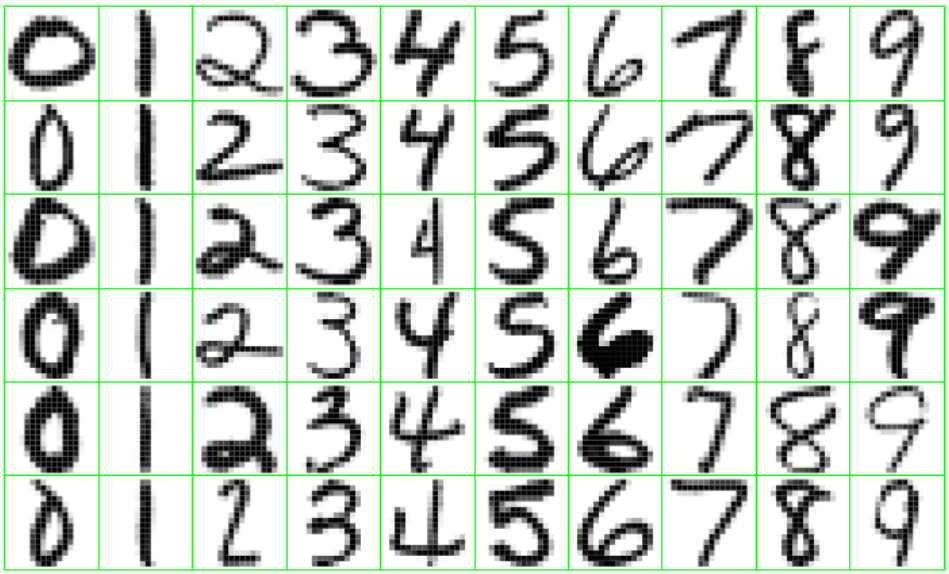


FIGURE 1.2. *Examples of handwritten digits from U.S. postal envelopes.*

- data: 16 x 16 matrix of pixel intensities;
- goal: identify the correct digit (0,, . . . , 9);
- the outcome consists of 10 classes.

Introduction: other examples

Examples (from the book):

- predict whether a patient, hospitalized due to a heart attack, will have a second heart attack, based on demographic, diet and clinical measurements for that patient;
- predict the price of a stock in 6 months from now, on the basis of company performance measures and economic data;
- identify the numbers in a handwritten ZIP code, from a digitized image;
- estimate the amount of glucose in the blood of a diabetic person, from the infrared absorption spectrum of that persons blood;
- identify the risk factors for prostate cancer, based on clinical and demographic.

Introduction: framework

In a typical scenario we have:

- an outcome Y (dependent variable, response)
  - quantitative (e.g., stock price, amount of glucose, ...);
  - categorical (e.g., heart attack/no heart attack)

that we want to predict based on

- a set of features $X_1, X_2, \ldots, X_p$ (independent variables, predictors)
  - examples: age, gender, income, ...

In practice,

- we have a training set, in which we observe the outcome and some features for a set of observations (e.g., persons);
- we use these data to construct a learner (i.e., a rule $f(X)$), which provides a prediction of the outcome ($\hat{y}$) given specific values of the features.

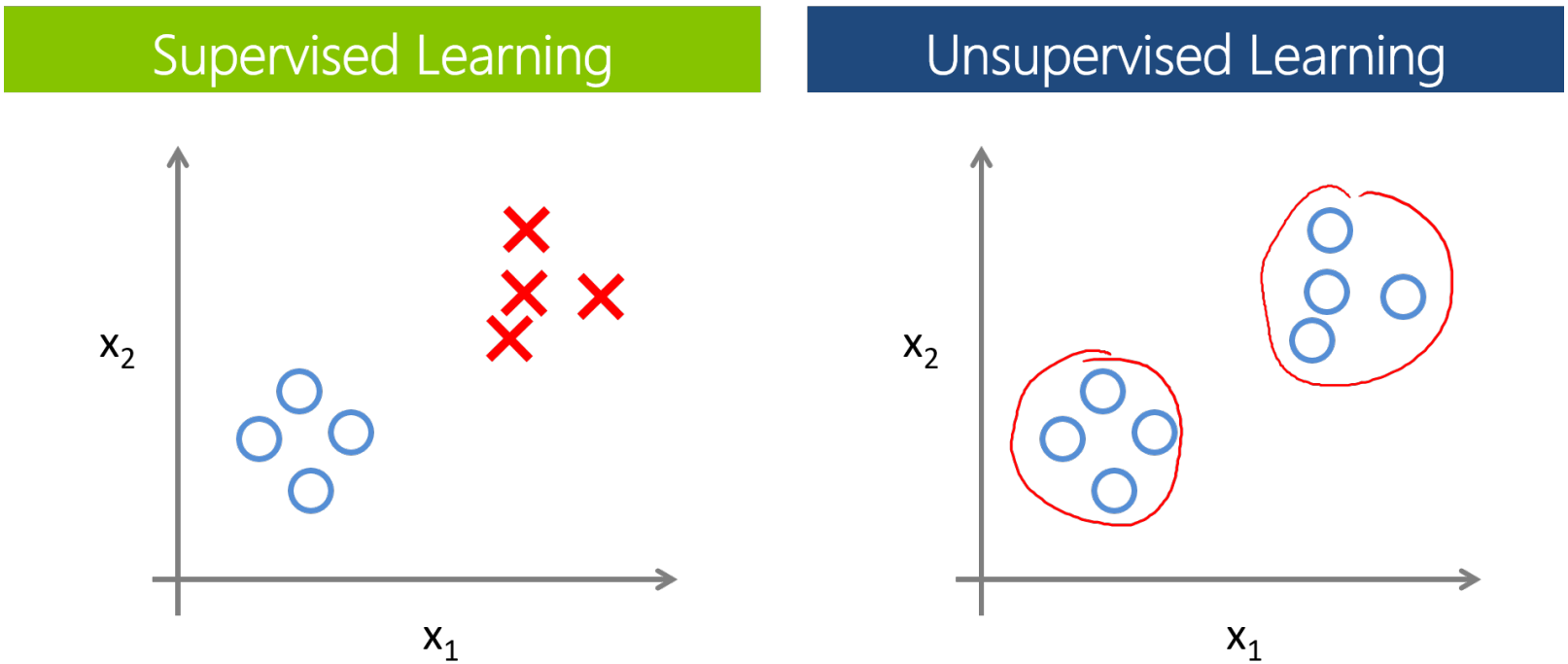
Introduction: supervised vs unsupervised learning

The scenario above is typical of a supervised learning problem:

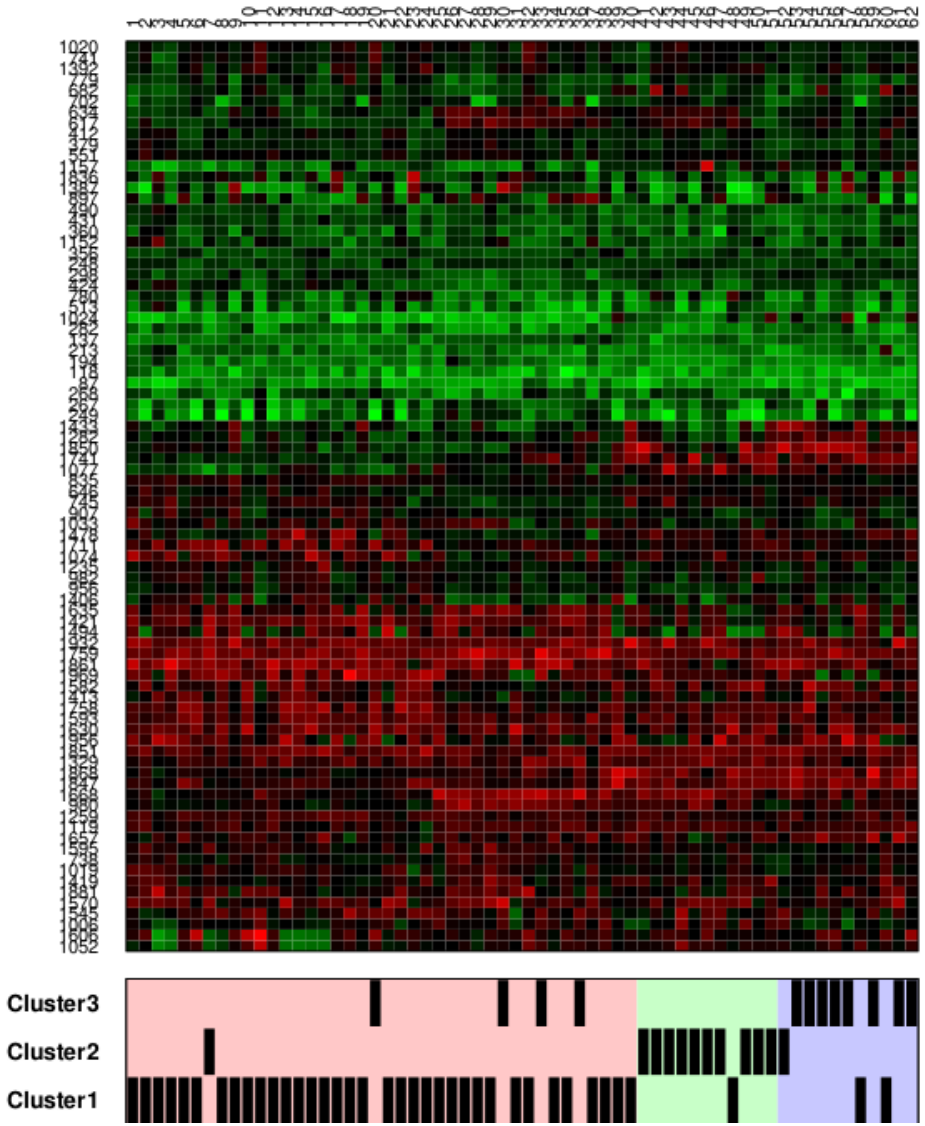
- the outcome is measured in the training data, and it can be used to construct the learner $f(X)$;

In other cases only the features are measured $\rightarrow$ unsupervised learning problems:

- identification of clusters, data simplification, ...



Introduction: gene expression example



- heatmap from De Bin & Risso (2011): 62 obs vs a subset of the original 2000 genes
  - $p >> n$ problem;
- goal: group patients with similar genetic information (cluster);
- alternatives (if the outcome was also available):
  - classify patients with similar disease (classification);
  - predict the chance of getting a disease for a new patient (regression).

Introduction: the high dimensional issue

Assume a training set $\{(x_{i1}, \ldots, x_{ip}, y_i), i = 1, ..., n\}$, where $n = 100, p = 2000$;

- possible model: $y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i$;
- least squares estimate: $\hat{\beta} = (X^T X)^{-1} X^T y$.

Exercise:

- go together in groups of 3-4;
- learn the names of the others in the group;
- discuss problems with the least squares estimate in this case;
- discuss possible ways to proceed;

Introduction: the hig dinmensional issue

Major issue: $X^T X$ is not invertible, infinitely many solutions!

Some possible directions:

- dimension reduction (reducing p to be smaller than n),
  - remove variables having low correlation with response;
  - more formal subset selections;
  - select a few "best" linear combinations of variables;
- shrinkage methods (adding constrain to β),
  - ridge regression;
  - lasso (least absolute shrinkage and selection operator)
  - elastic net.

Introduction: course information

- Course: mixture between theoretical and practical;
- evaluation: project (practical) and written exam (theoretical);
- use of computer necessary;
- based on statistical package R:
  - suggestion: use R Studio (`www.rstudio.com`), available at all Linux computers at the Department of Mathematics;
  - encouragement: follow good R programming practices, for instance consult Google's R Style Guide.

Variable types and terminology

Variable types: quantitative (numerical), qualitative (categorical).

Naming convention for predicting tasks:

- quantitative response: regression;
- qualitative response: classification.

We start with the problem of taking two explanatory variables X_1 and X_2 and predicting a binary (two classes) response G:

- we illustrate two basic approaches:
  - linear model with least squares estimator;
  - k nearest neighbors;
- we consider both from a statistical decision theory point of view.

Two simple approaches to prediction: linear regression model

The linear regression model

$$\begin{aligned} Y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon \\ &= X\beta + \varepsilon, \qquad \text{where } X = (\mathbf{1}, x_1, \ldots, x_p), \end{aligned}$$

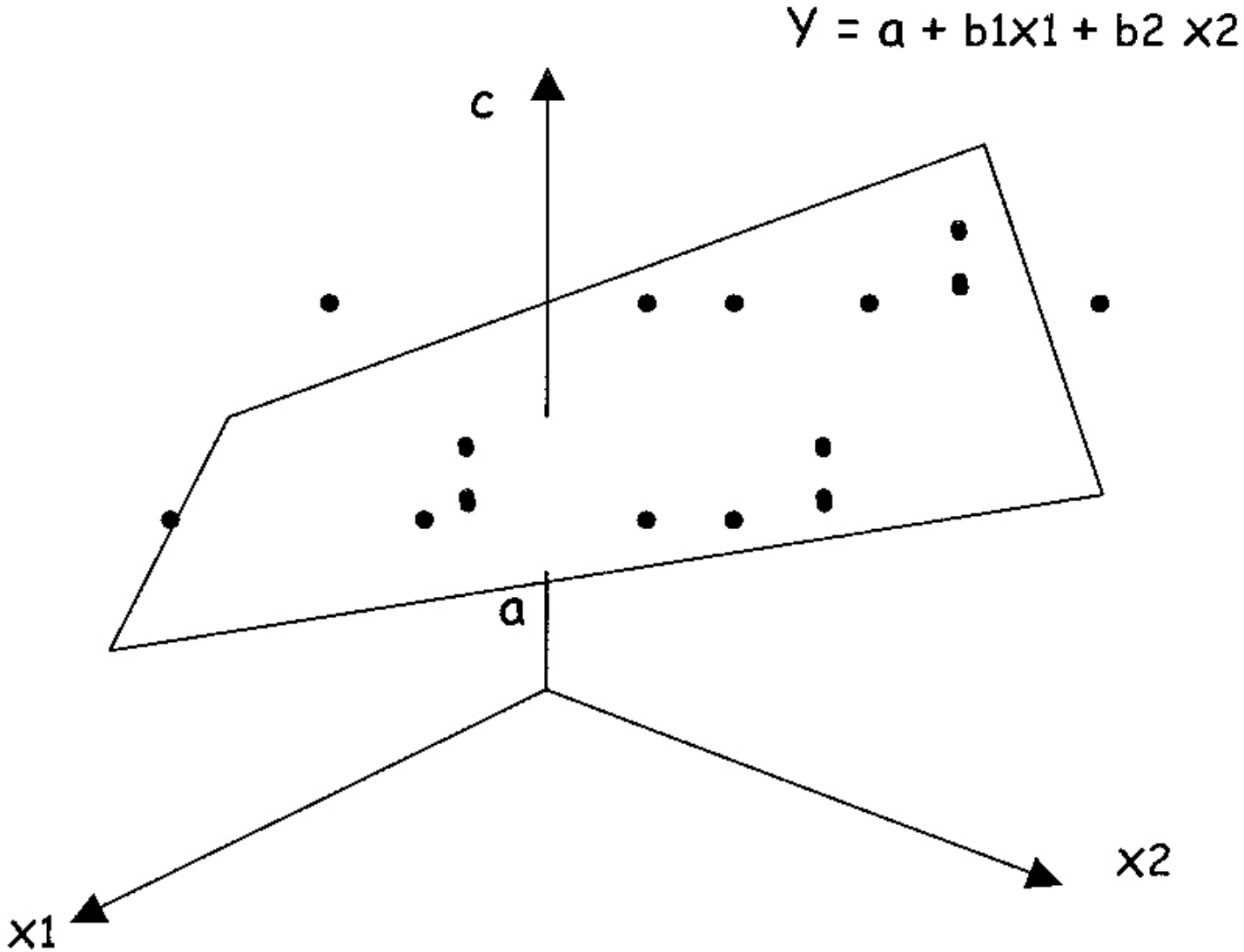
can be use to predict the outcome y given the values $x_1, x_2, \ldots, x_p$, namely

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

Properties:

- easy interpretation;
- easy computations involved;
- theoretical properties available;
- it works well in many situations.

Two simple approaches to prediction: linear regression model



Two simple approaches to prediction: least square

How do we fit the linear regression model to a training dataset?

- Most popular method: least square;
- estimate β by minimizing the residual sum of squares

$$\text{RSS}(\beta) = \sum_{i=1}^{N}(y_i - x_i^T\beta)^2 = (y - X\beta)^T(y - X\beta)$$

where X is a $(N \times p)$ matrix and y a N-dimensional vector.

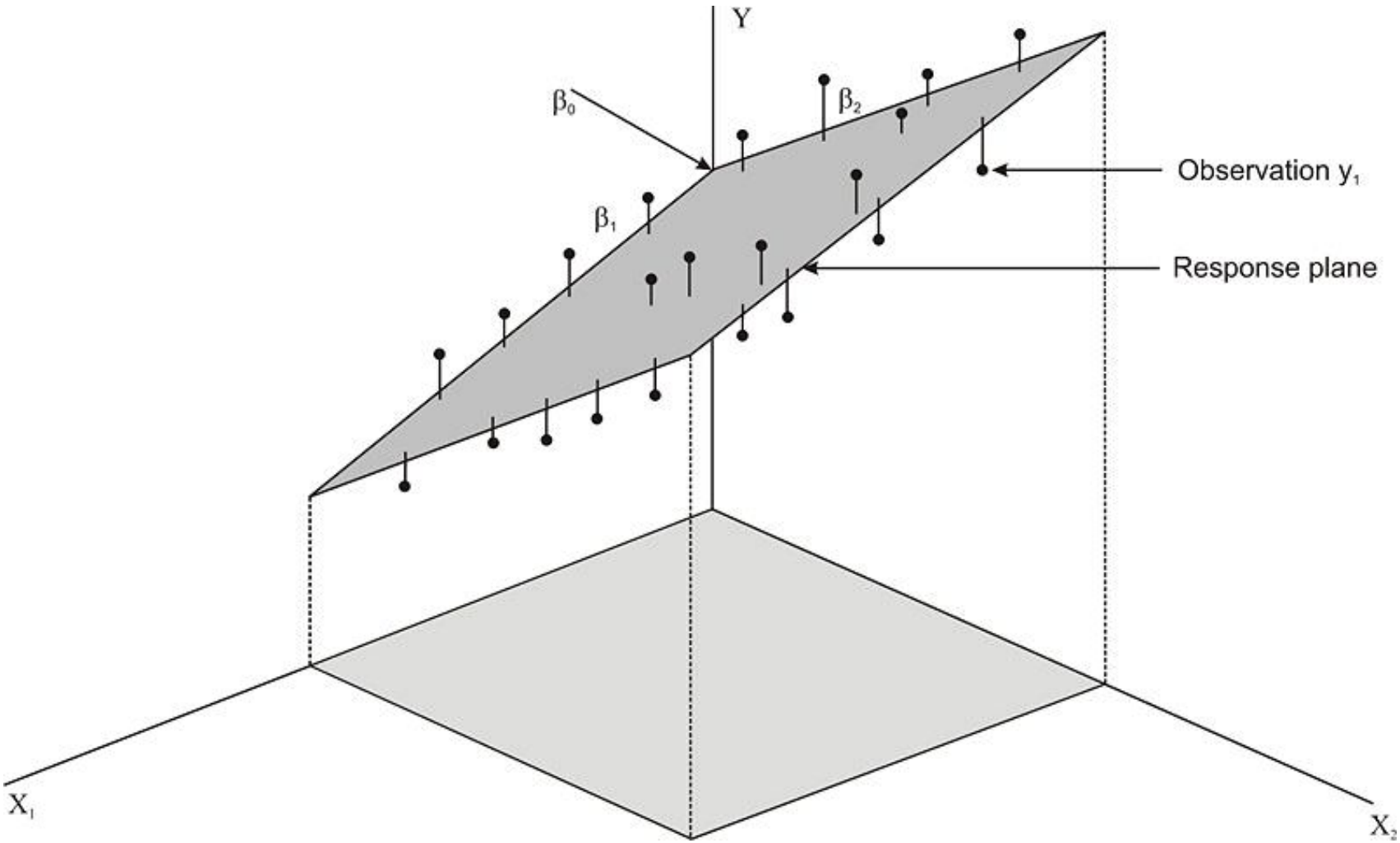
Differentiating w.r.t. β, we obtain the estimating equation

$$X^T(y - X\beta) = 0,$$

from which, when (X^TX) is non-singular, we obtain

$$\hat{\beta} = (X^TX)^{-1}X^Ty$$

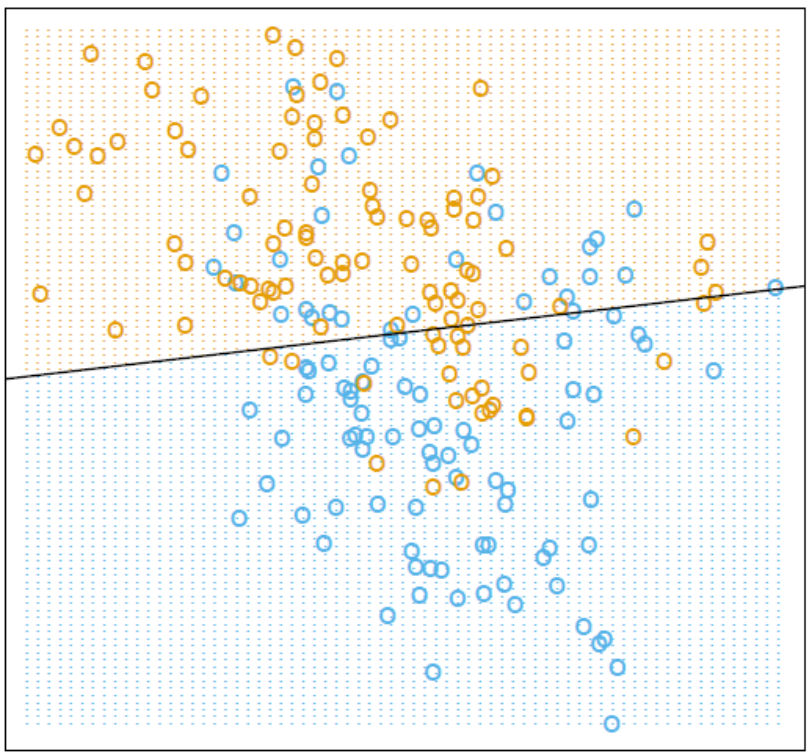
Two simple approaches to prediction: least square



Two simple approaches to prediction: least square for binary response

Simulated data with two variables and two classes:

$$Y = \begin{cases} 1 & \text{orange} \\ 0 & \text{blue} \end{cases}$$



If $Y \in \{0, 1\}$ is treated as a numerical response

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2,$$

a prediction rule

$$\hat{G} = \begin{cases} 1 \text{ (orange)} & \text{if } \hat{Y} > 0.5 \\ 0 \text{ (blue)} & \text{otherwise} \end{cases}$$

gives linear decision boundary $\{x^T \hat{\beta} = 0.5\}$

- optimal under Gaussian assumptions;
- is it better with nonlinear decision boundary?

Two simple approaches to prediction: Nearest neighbor methods

A different approach consists in looking at the closest (in the input space) observations to x and, based on their output, form $\hat{Y}(x)$.

The k nearest neighbors prediction of x is the mean

$$\hat{Y}(x) = \frac{1}{k} \sum_{i:x_i \in N_k(x)} y_i,$$

where $N_k(x)$ contains the k closest points to x.

- less assumptions on $f(x)$;
- we need to decide k;
- we need to define a metric (for now, consider the Euclidean distance).

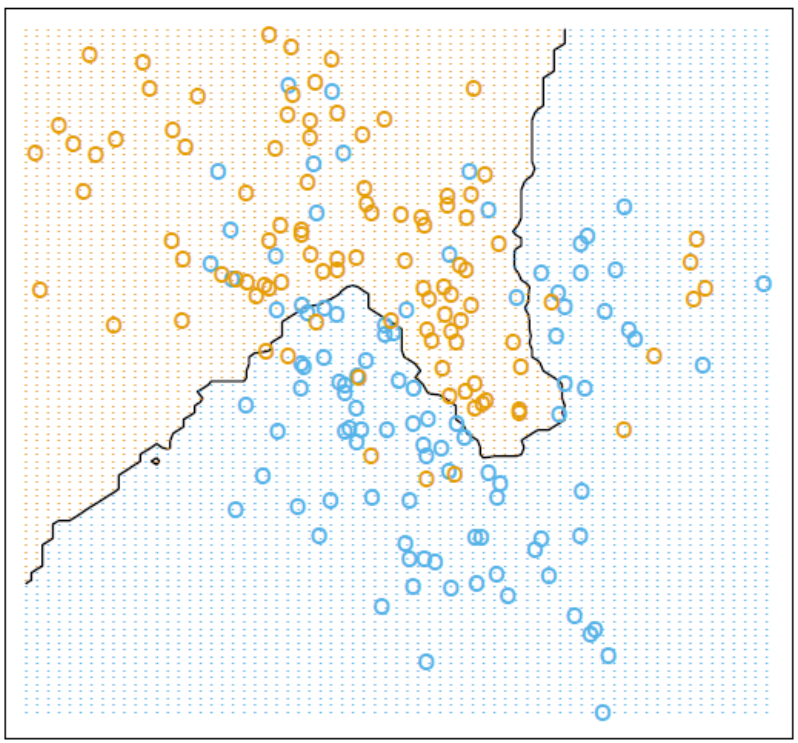
Two simple approaches to prediction: nearest neighbor methods

Use the same training data (simulated) as before:

$$Y = \begin{cases} 1 & \text{orange} \\ 0 & \text{blue} \end{cases}$$

Classify to orange, if there are mostly orange points in the neighborhood:

$$\hat{G} = \begin{cases} 1 \text{ (orange)} & \text{if } \hat{Y} > 0.5 \\ 0 \text{ (blue)} & \text{otherwise} \end{cases}$$

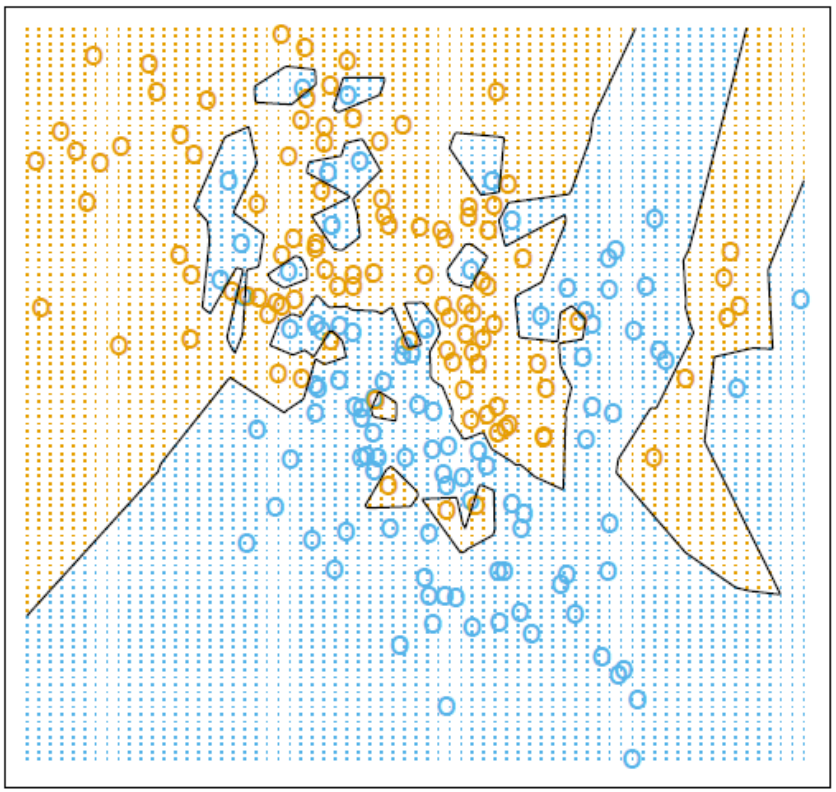


- $k = 15$;
- flexible decision boundary;
- better performance than the linear regression case:
  - fewer training observations are missclassified;
  - is this a good criterion?

Two simple approaches to prediction: nearest neighbor methods

Using the same data as before:

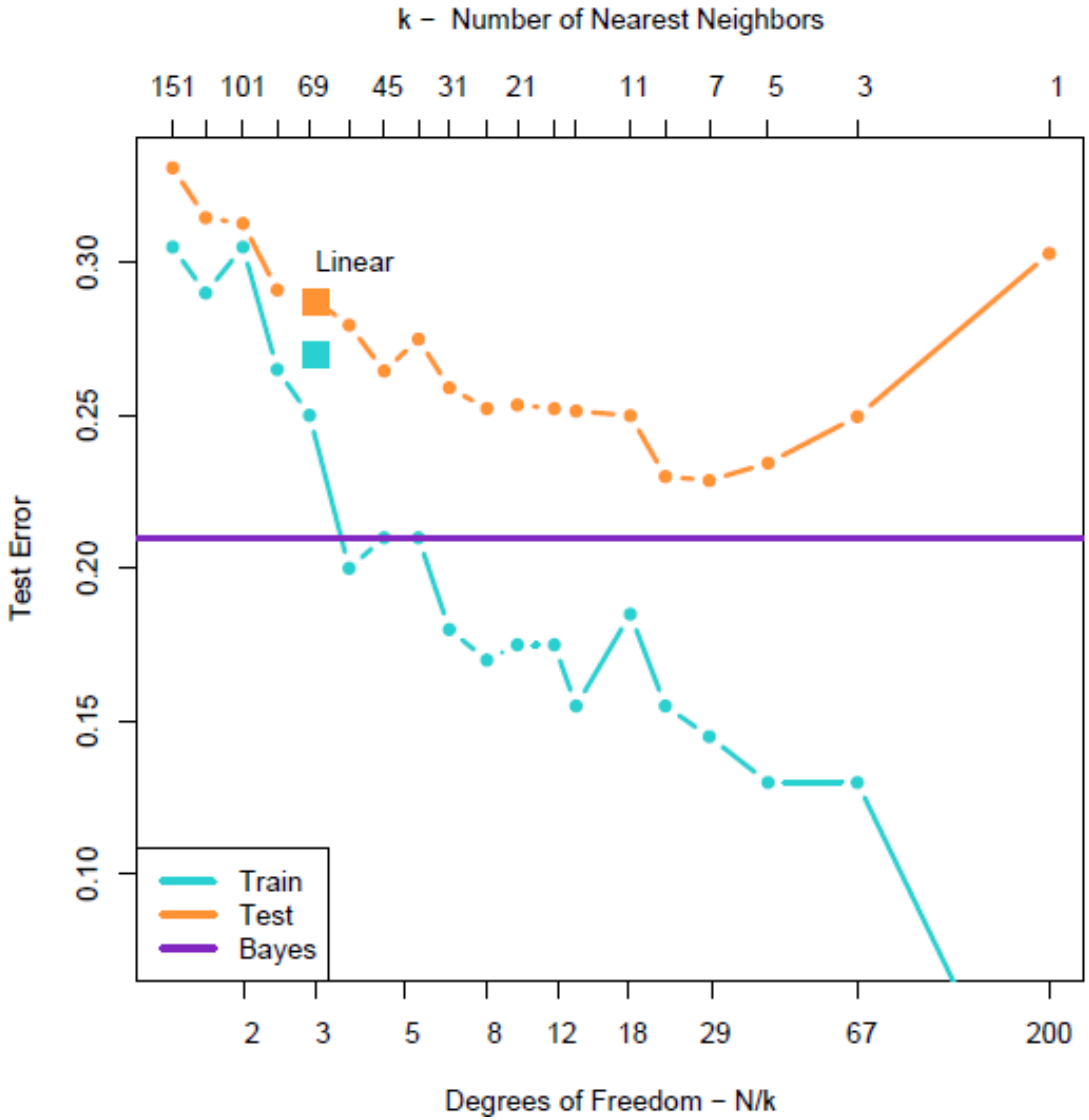
$$Y = \begin{cases} 1 & \text{orange} \\ 0 & \text{blue} \end{cases}$$



Note:

- same approach, with $k = 1$;
- no training observations are missclassified!!!
- Is this a good solution?
  - the learner works greatly on the training set, but what its prediction ability? (remember this term: overfitting);
  - It would be preferable to evaluate the performance of the methods in an independent set of observations (test set);
- bias-variance trade-off.

Two simple approaches to prediction: how many neighbors in KNN?



Two simple approaches to prediction: alternatives

Most of the modern techniques are variants of these two simple procedures:

- kernel methods that weight data according to distance;
- in high dimension: more weight on some variables;
- local regression models;
- linear models of functions of X;
- projection pursuit and neural network.

Statistical decision theory: theoretical framework

Statistical decision theory gives a mathematical framework for finding the optimal learner.

Let:

- $X \in \mathbb{R}^p$ be a p-dimensional random vector of inputs;
- $Y \in \mathbb{R}$ be a real value random response variable;
- $p(X, Y)$ be their joint distribution;

Our goal is to find a function $f(X)$ for predicting Y given X:

- we need a loss function $L(Y, f(X))$ for penalizing errors in $f(X)$ when the truth is Y,
  - example: squared error loss, $L(Y, f(X)) = (Y - f(X))^2$.

Statistical decision theory: expected prediction error

Given $p(X, Y)$, it is possible to derive the expected prediction error of $f(X)$:

$$\text{EPE}(f) = E\left[L(Y, f(X))\right] = \int_{x,y} L(y, f(x))p(x, y)dxdy;$$

we have now a criterion for choosing a learner: find f which minimizes $\text{EPE}(f)$.

The aforementioned squared error loss,

$$L(Y, f(X)) = (Y - f(X))^2,$$

is by far the most common and convenient loss function. Let us focus on it!

Statistical decision theory: squared error loss

If $L(Y, f(X)) = (Y - f(X))^2$, then

$$\begin{aligned}\text{EPE}(f) =& E_{X,Y}[(Y - f(X))^2] \\ =& E_X E_{Y|X}[(Y - f(X))^2|X]\end{aligned}$$

It is then sufficient to minimize $E_{Y|X}[(Y - f(X))^2|X]$ for each X:

$$f(x) = \text{argmin}_c E_{Y|X}[(Y - c)^2|X = x],$$

which leads to

$$f(x) = E[Y|X = x],$$

i.e., the conditional expectation, also known as regression function.

Thus, by average squared error, the best prediction of Y at any point $X = x$ is the conditional mean.

Statistical decision theory: estimation of optimal f

In practice, $f(x)$ must be estimated.

Linear regression:

- assumes a function linear in its arguments, $f(x) \approx x^T\beta$;
- $\text{argmin}_\beta E[Y - X^T\beta] \to \beta = E[XX^T]^{-1}E[XY]$;
- replacing the expectations by averages over the training data leads to $\hat{\beta}$.
- Note:
  - no conditioning on X;
  - we have used our knowledge on the functional relationship to pool over all values of X (model-based approach);
  - less rigid functional relationship may be considered, e.g.

$$f(x) \approx \sum_{j=1}^{p} f(x_j).$$

Statistical decision theory: estimation of optimal f

K nearest neighbors:

- uses directly $f(x) = E[Y|X = x]$:
- $\hat{f}(x_i) = \text{Ave}(y_i)$ for observed x_i's;
- normally there is at most one observation for each point x_i;
- uses points in the neighborhood,

$$\hat{f}(x) = \text{Ave}(y_i|x_i \in N_k(x))$$

- there are two approximations:
  - expectation is approximated by averaging over sample data;
  - conditioning on a point is relaxed to conditioning on a neighborhood.

Statistical decision theory: estimation of optimal f

- assumption of k nearest neighbors: $f(x)$ can be approximated by a locally constant function;
- for $N \to \infty$, all $x_i \in N_k(x) \approx x$;
- for $k \to \infty$, $\hat{f}(x)$ is getting more stable;
- under mild regularity condition on $p(X, Y)$,

$$\hat{f}(x) \to E[Y|X = x] \text{ for } N, k \to \infty \text{ s.t. } k/N \to 0$$

- is this an universal solution?
  - small sample size;
  - curse of dimensionality (see later)

Statistical decision theory: other loss function

- It is not necessary to implement the squared error loss function (L_2 loss function);
- a valid alternative is the L_1 loss function:
  - the solution is the conditional median

$$\hat{f}(x) = \text{median}(Y|X = x)$$

  - more robust estimates than those obtained with the conditional mean;
  - the L_1 loss function has discontinuities in its derivatives $\rightarrow$ numerical difficulties.

Statistical decision theory: other loss functions

What happens with a categorical outcome G?

- similar concept, different loss function;
- $G \in \mathcal{G} = \{1, \ldots, K\} \rightarrow \hat{G} \in \mathcal{G} = \{1, \ldots, K\}$;
- $L(G, \hat{G}) = L_{G,\hat{G}}$ a $K \times K$ matrix, where $K = |G|$;
- each element of the matrix l_{ij} is the price to pay to missallocate category g_i as g_j
  - all elements on the diagonal are 0;
  - often non-diagonal elements are 1 (zero-one loss function).

Statistical decision theory: other loss functions

Mathematically:

$$\begin{aligned} EPE =& E_{G,X}[L(G, \hat{G}(X))] \\ =& E_X \left[E_{G|X}[L(G, \hat{G}(X))] \right] \\ =& E_X \left[\sum_{k=1}^{K} L(g_k, \hat{G}(X)) \Pr(G = g_k | X) \right] \end{aligned}$$

which is sufficient to be minimized pointwise, i.e,

$$\hat{G} = \text{argmin}_{g \in \mathcal{G}} L(g_k, g) \Pr(G = g_k | X = x).$$

When using the 0-1 loss function

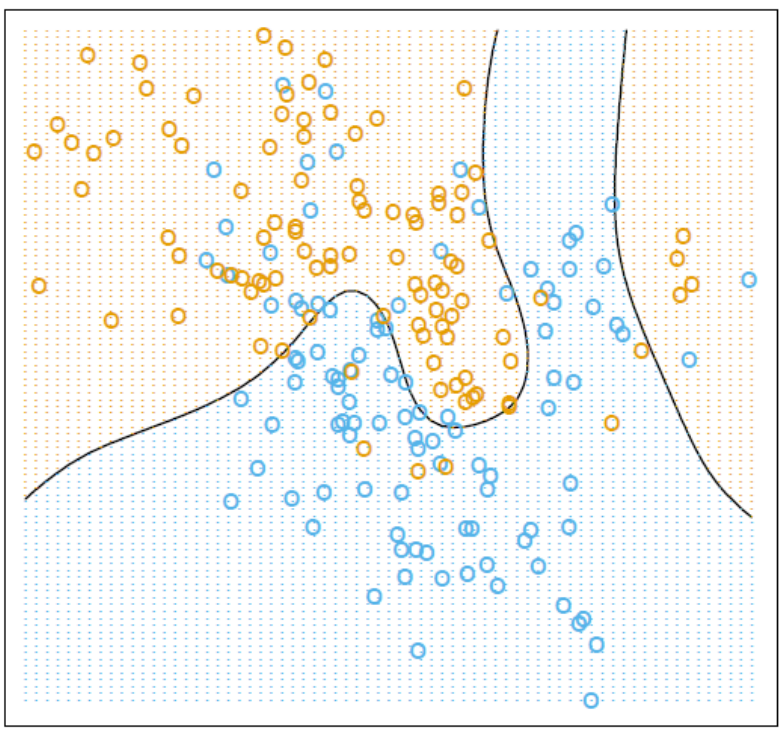
$$\begin{aligned} \hat{G} =& \text{argmin}_{g \in \mathcal{G}} \sum_{k=1}^{K} \{1 - I(G = g_k))\} \Pr(G = g_k | X = x) \\ =& \text{argmin}_{g \in \mathcal{G}} \{1 - \Pr(G = g_k | X = x)\} \\ =& \text{argmax}_{g \in \mathcal{G}} \Pr(G = g_k | X = x) \end{aligned}$$

Statistical decision theory: other loss functions

Alternatively,

$$\hat{G}(x) = g_k \text{ if } P(G = g_k|X = x) = \max_{g \in \mathcal{G}} \Pr(G = g|X = x),$$

also known as Bayes classifier.



- k nearest neighbor:
  - $\hat{G}(x) =$ category with largest frequency in k nearest samples;
  - approximation of this solution.
- regression:
  - $E[Y_k|X] = \Pr(G = g_k|X)$;
  - also approximates the Bayes classifier.

Local methods in high dimensions

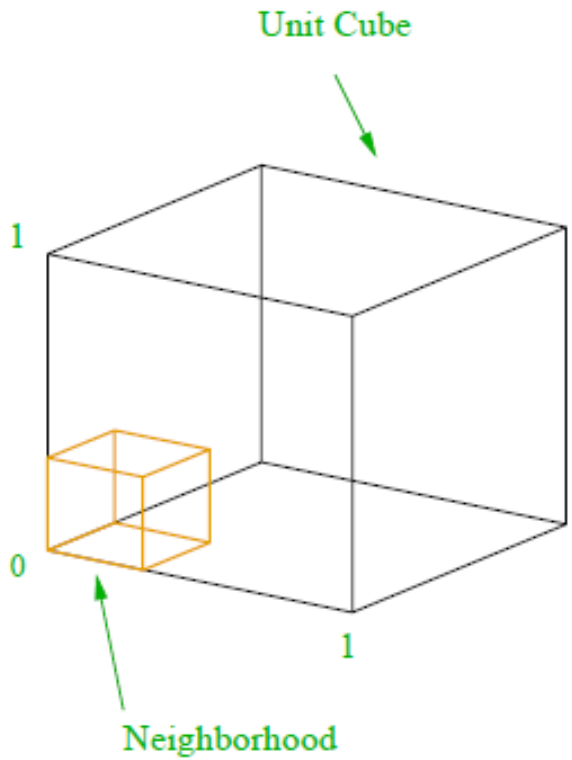
The two (extreme) methods seen so far:

- linear model, stable but biased;
- k-nearest neighbor, less biased but less stable.

For large set of training data:

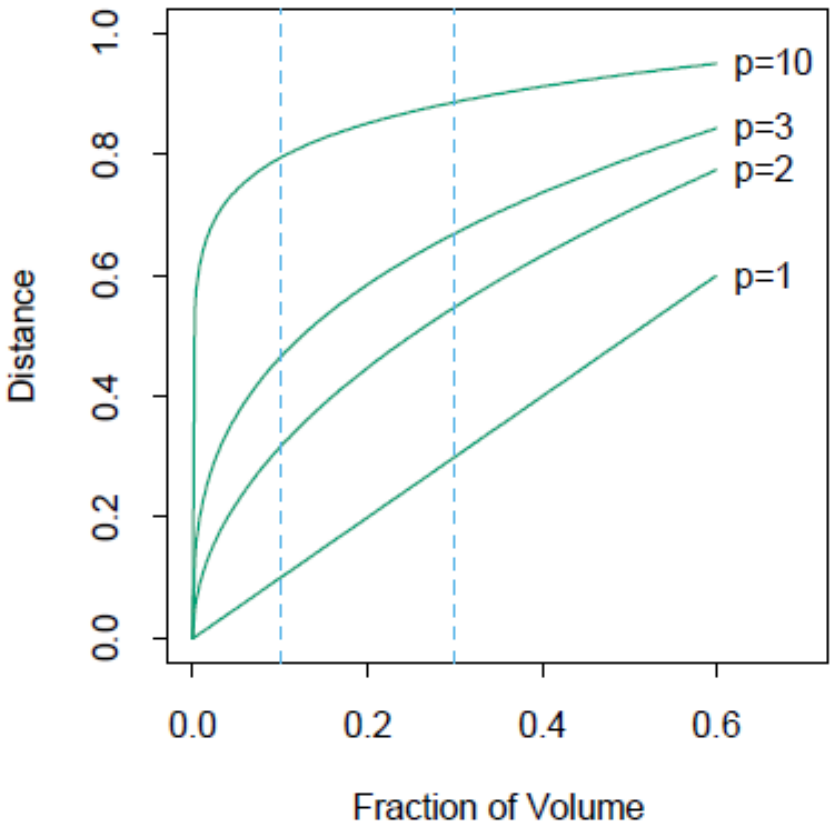
- always possible to use k nearest neighbors?
- Breaks down in high dimensions $\rightarrow$ curse of dimensionality (Bellman, 1961).

Local methods in high dimensions: curse of dimensionality



- Assume $X \sim \text{Unif}[0,1]^p$;
- define e_p the expected length size of a hypercube containing a fraction r of input points;
- $e_p(r) = r^{1/p}$ $\left(e^p = r \Leftrightarrow e = r^{1/p}\right)$;

Local methods in high dimensions: curse of dimensionality



- Expected length: $e_p(r) = r^{1/p}$

p	1	2	3	5
$e_p(0.01)$	0.01	0.10	0.22	0.63
$e_p(0.1)$	0.10	0.32	0.46	0.79

Local methods in high dimensions: curse of dimensionality

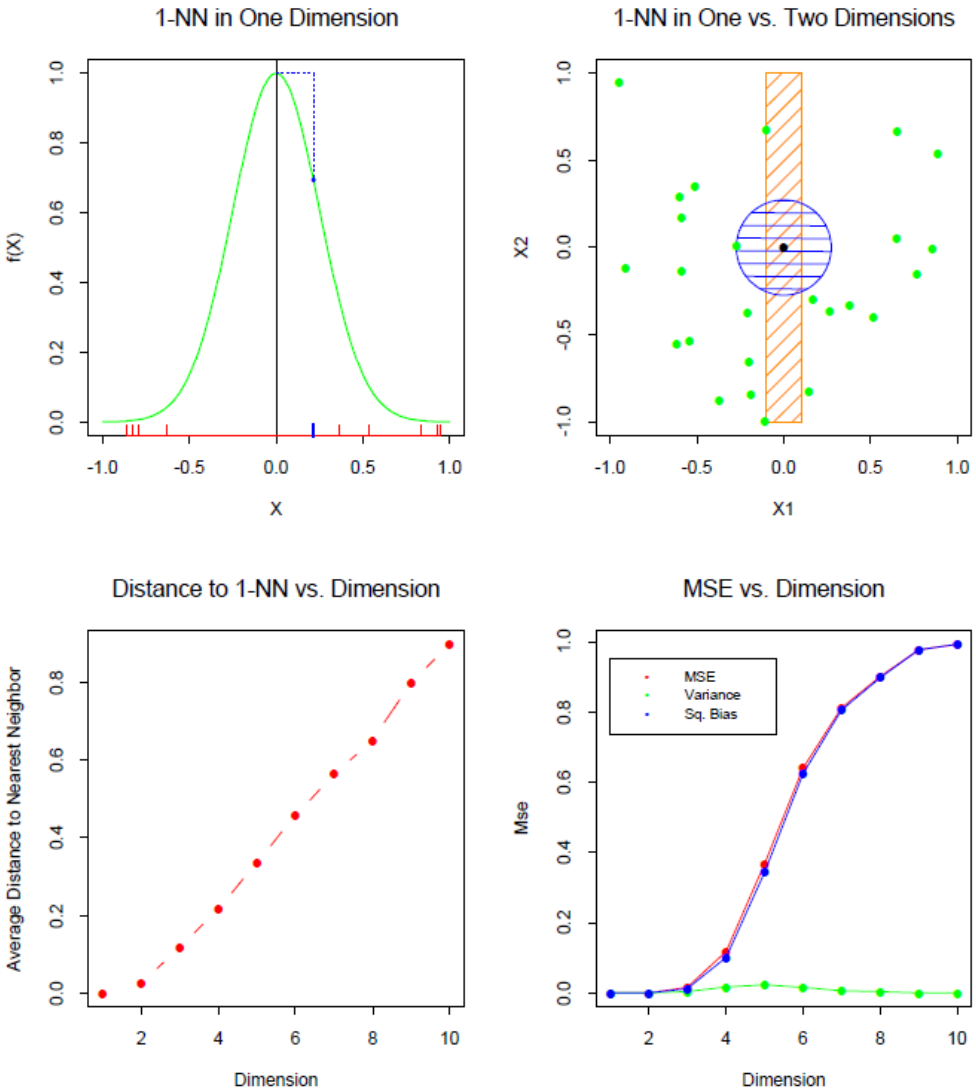
Assume $Y = f(X) = e^{-8||X||^2}$

and use the 1-nearest neighbor to predict y_0 at $x_0 = 0$, i.e.

$\hat{y}_0 = y_i$ s.t. x_i nearest observed

$$
\begin{aligned}
\text{MSE}(x_0) = & \\
&= E_{\mathcal{T}}[\hat{y}_0 - f(x_0)]^2 \\
&= E_{\mathcal{T}}[\hat{y}_0 - E_{\mathcal{T}}(\hat{y}_0)]^2 \\
&\quad + [E_{\mathcal{T}}(\hat{y}_0) - f(x_0)]^2 \\
&= \text{Var}(\hat{y}_0) + \text{Bias}^2(\hat{y}_0)
\end{aligned}
$$

NB: we will see often this bias-variance decomposition!



Local methods in high dimensions: EPE in the linear model

- Assume now $Y = X^T\beta + \varepsilon$
- we want to predict $y_0 = x_0^T\beta + \varepsilon_0$ with x_0 fixed
- $\hat{y}_0 = x_0^T\hat{\beta}$ where $\hat{\beta} = (X^TX)^{-1}X^Ty$

$$\begin{aligned}
\text{EPE}(x_0) &= E(y_0 - \hat{y}_0)^2 \\
&= E\left[(y_0 - E[y_0|x_0] + E[y_0|x_0] - E[\hat{y}_0|x_0] + E[\hat{y}_0|x_0] - \hat{y}_0)^2\right] \\
&= E(y_0 - E[y_0|x_0])^2 + (E[y_0|x_0] - E[\hat{y}_0|x_0])^2 \\
&\quad + E(\hat{y}_0 - E[\hat{y}_0|x_0])^2 \\
&= \text{Var}(y_0|x_0) + \text{Bias}^2(\hat{y}_0) + \text{Var}(\hat{y}_0)
\end{aligned}$$

True and assumed linear model

- Bias=0
- $\text{Var}(\hat{y}_0) = x_0^TE(X^TX)^{-1}x_0\sigma^2$
- $\text{EPE}(x_0) = \sigma^2 + x_0^TE(X^TX)^{-1}x_0\sigma^2$

Local methods in high dimensions: EPE in the linear model

- $\text{EPE}(x_0) = \sigma^2 + x_0^T E(X^T X)^{-1} x_0 \sigma^2$
- If x's drawn from a random distribution with $E(X) = 0$, $X^T X \to N\text{Cov}(X)$
- Assume also x_0 drawn from same distribution:

$$\begin{aligned} E_{x_0}\left[\text{EPE}(x_0)\right] \approx & \sigma^2 + E_{x_0}[x_0^T]\text{Cov}(X)^{-1} x_0 N^{-1}\sigma^2 \\ = & \sigma^2 + N^{-1}\sigma^2 \text{trace}[\text{Cov}(X)^{-1} E_{x_0}[x_0 x_0^T]] \\ = & \sigma^2 + N^{-1}\sigma^2 \text{trace}[\text{Cov}(X)^{-1}\text{Cov}(x_0)] \\ = & \sigma^2 + N^{-1}\sigma^2 \text{trace}[I_p] \\ = & \sigma^2 + N^{-1}\sigma^2 p \end{aligned}$$

- It increases linearly with p!

References I

Bellman, R. (1961). *Adaptive control process: a guided tour*. Princeton University Press, London.

De Bin, R. & Risso, D. (2011). A novel approach to the clustering of microarray data via nonparametric density estimation. *BMC Bioinformatics* **12**, 49.

Stamey, T. A., Kabalin, J. N., McNeal, J. E., Johnstone, I. M., Freiha, F., Redwine, E. A. & Yang, N. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. ii. Radical prostatectomy treated patients. *The Journal of Urology* **141**, 1076–1083.