Statistical Computing

Zhang Shuqin
Email:zhangs@fudan.edu.cn

Address: HGX502, Mon. 3-5

February 28, 2016

- Lecture notes available on elearning
- Reference books:
  Computational Statistics, by J. E. Gentle
  Simulation, by S.M.Ross
  The EM Algorithms and Extensions, by G. J. McLachlan, T. Krishnan
  Monte-Carlo Statistical Methods, by C.P. Robert

Contents of statistical computing

- Numerical Linear Algebra
- Optimization and Nonlinear Equations
- Simulations,Data analysis,EM,MCMC...

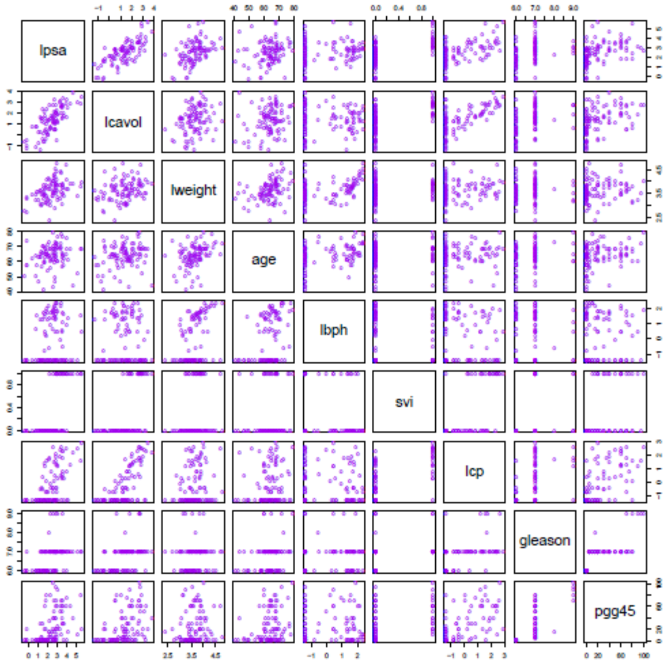
Introduction

Statistical modelling:

- Generate data,
- Mathematical, statistical modelling,
- Solve the model,
- Results explanation.

Example 1: Linear regression

- y is called the response/outcome and $x = (x_1, \cdots, x_p)$ is a set of explanatory variables.
- Given n data $\{y^i, x^i\}$, we want to determine a model relating y to x.
- If $y \in R^k$,regression problem.


lpsa
lcavol
lweight
age
lbph
svi
lcp
gleason
pgg45

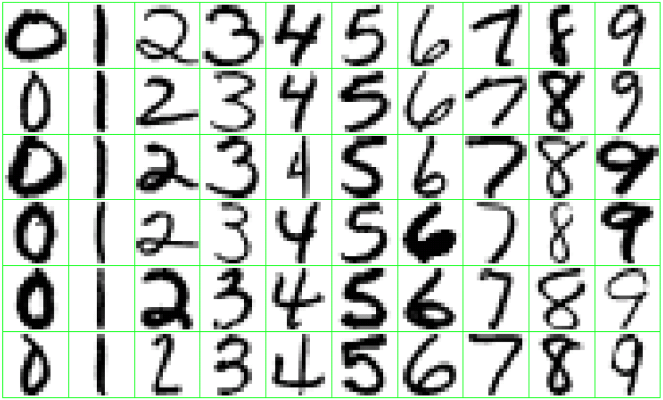
- Data: use 97 men who were about to receive a radical prostatectomy, a number of measurements including log cancer volume (lcavol), log prostate weight (lweight), age, log of benign prostatic hyperplasia amount (lbph), seminal vesicle invasion (svi), log of capsular penetration (lcp), Gleason score (gleason), percent of Gleason scores 4 or 5(pgg45).
- Problem: Predict the log of prostate specific antigen (lpsa) from the data

- $$\begin{aligned} y &= \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \epsilon, \text{where } \epsilon \sim N(0, \sigma^2) \\ &= f(x_i) + \epsilon \end{aligned}$$

- $$\min \|y - (\beta_0 + \sum_{i=1}^{p} \beta_i x_i)\|_2^2$$

Example 2: Linear classification

- y is called the response/outcome and $x = (x_1, \cdots, x_p)$ is a set of explanatory variables.
- Given n data $\{y^i, x^i\}$, we want to determine a model relating y to x.
- If y is categorical data, classification problem.



- Data: Postal code on the envelopes from US postal mails. The images are 16×16 eight-bit grayscale maps, with each pixel ranging in intensity from 0 to 255.
- Problem: Predict the identity of each image $(0, 1, \cdots, 9)$ quickly and accurately from the 16×16 matrix of pixel intensities.

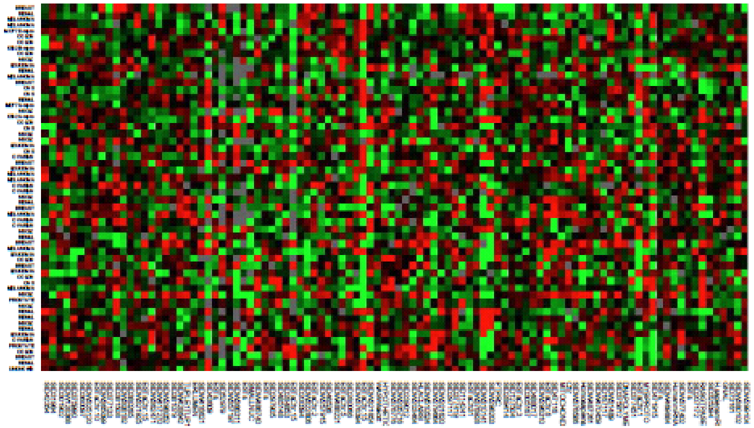
- Logistic regression: $y \in \{1, 2, \cdots, K\}$.

$$\log \frac{Pr(G = k|X = x)}{Pr(G = l|X = x)} = \beta_0^{(k,l)} + \sum_{j=1}^{p} \beta_j^{(k,l)} x_j$$

-

$$\begin{aligned} \max L(\beta) &= \sum_{i=1}^{N} \log p_{g_i}(x_i; \beta) \\ &= \sum_{i=1}^{N} [\bar{\beta}_{g_i}^T x_i - \log(1 + \sum_{l=1}^{k-1} e^{\bar{\beta}_l^T x_i})] \end{aligned}$$

Example 3: Clustering



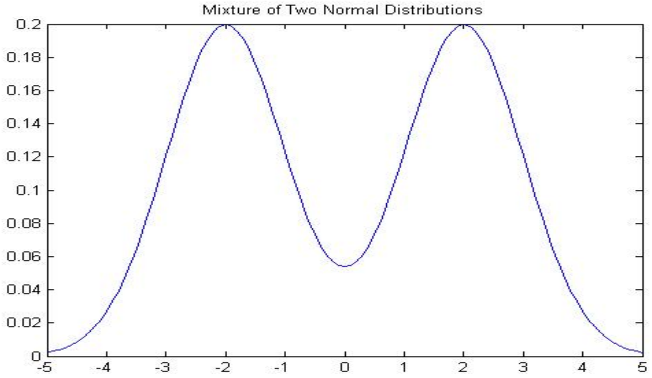
- Data: DNA microarray data:expression matrix of 6830 genes and 64 samples for the human tumor data,14 cancers.
- Problems:
  - Which samples are most similar to each other, in terms of their expression profiles across genes?
  - Which genes are most similar to each other, in terms of their expression profiles across samples?
  - Do certain genes show very high expression for certain cancer samples?

- Cluster the points having high similarities.
- Data: $(x_1, x_2, \cdots, x_n)$, K sets.
- $$\min_S \sum_{i=1}^{K} \sum_{x_i \in S_i} \|x_j - c_i\|_2^2$$

Example 4: Finite Mixture Models

$$f(x) = \sum_{i=1}^{k} p_i f_i(x)$$

- Finite mixture models appear everywhere and are used for modelling multimodal distributions.
- Allows to model populations as mixture of several subpopulations.
- Data clustering.


Mixture of Two Normal Distributions
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-5
-4
-3
-2
-1
0
1
2
3
4
5

- Finite Mixture of Gaussians

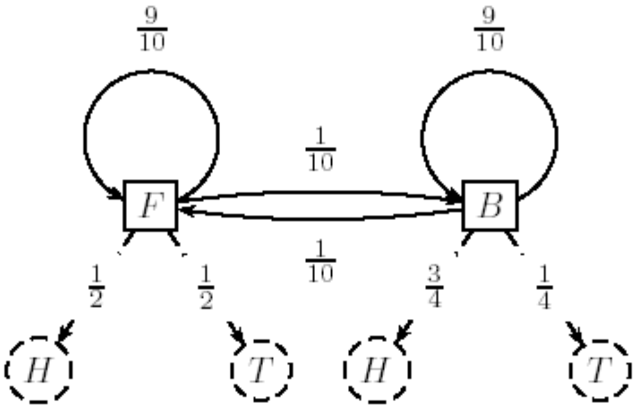
$$f(x|\theta) = \sum_{i=1}^{k} p_i N(x; \mu_i, \sigma_i),$$

where $\theta = (\mu_i, \sigma_i^2, p_i)$ is estimated from some data $(x_1, x_2, \cdots, x_n)$.
- A standard approach consists of finding a local maximum of the log-likelihood

$$\max \sum_{i=1}^{n} \log f(x_i|\theta).$$

- Solve the problem: EM algorithm

Hidden Markov Models



- A hidden/unobserved Markov process is defined, call it x_t. It can take values in a finite space or a continuous space.
- We only have access to an observation process y_t. The observations are conditionally independent given x_t.
- Examples: HMM are used for example to model DNA sequences, speech processing, etc..

- Model:

$$p(x_1, y_1, x_2, y_2, \cdots, x_T, y_T) = p(x_1)p(y_1|x_1)\Pi_{i=2}^{T}p(x_i|x_{i-1})p(y_i|x_i)$$

- When x_i is known, MLE
- When x_i is unknown, how to solve the model?

Errors

- Empirical measurements error
  – Random error $\epsilon \sim N(0, \sigma^2)$.
  – Systematic error $\epsilon \sim N(\mu_0, \sigma^2)$, μ_0 is the systematic error.
- Modelling error
- Computational error

Computational error

The systematic approximations that occur during computation include

- Truncation or discretization: Some features of a mathematical model may be omitted or simplified.
  - Replacing infinite processes with finite processes, such as replacing integrals or infinite series with finite sums, or derivatives with finite difference quotients
  - Replacing general matrices with matrices having a simpler form
  - Replacing complicated functions with simple functions, such as polynomials
  - Replacing nonlinear problems with linear problems
  - Replacing differential equations with algebraic equations
  - Replacing infinite-dimensional spaces with finite-dimensional spaces
- Rounding: The computer representation of real numbers and arithmetic operations upon them is generally inexact.

Error Analysis

A typical problem can be viewed as the computation of the value of a function, say $f : R \Rightarrow R$. Denote the true value of the input data by x, so that the desired true result is $f(x)$. Suppose that we must work with inexact input, say $\hat{x}$, and we can compute only an approximation to the function, say $\hat{f}$. Then

$$\begin{aligned} \text{Total} \quad \text{error} &= \hat{f}(\hat{x}) - f(x) \\ &= (\hat{f}(\hat{x}) - f(\hat{x})) + (f(\hat{x}) - f(x)) \\ &= \text{computational} \quad \text{error} + \text{propagated} \quad \text{data} \quad \text{error} \end{aligned}$$

Truncation Error and Rounding Error

- Truncation error is the difference between the true result (for the actual input) and the result that would be produced by a given algorithm using exact arithmetic. It is due to approximations such as truncating an infinite series, replacing a derivative by a finite difference quotient, replacing an arbitrary function by a polynomial, or terminating an iterative sequence before convergence.
- Rounding error is the difference between the result produced by a given algorithm using exact arithmetic and the result produced by the same algorithm using finite-precision, rounded arithmetic.

Compute sin($\pi/8$)

$$\sin(\pi/8) \approx \sin(3/8) \approx 3/8 = 0.375$$

Suppose the correct answer to four decimal digits is:

$$\sin(\pi/8) \approx 0.3827$$

Total error:

$$\hat{f}(\hat{x}) - f(x) \approx 0.3750 - 0.3827 = -0.0077$$

The correct answer for the purturbed input is:

$$f(\hat{x}) = sin(3/8) \approx 0.3663,$$

The propagated data error:

$$f(\hat{x}) - f(x) = \sin(3/8) - sin(\pi/8) \approx 0.3663 - 0.3827 = -0.0164$$

The computational error caused by truncating the infinite series is:

$$\hat{f}(\hat{x}) - f(\hat{x}) = 3/8 - \sin(3/8) \approx 0.3750 - 0.3663 = 0.0087$$

Absolute Error and Relative Error

Absolute error=approximate value-true value;
Relative error=absolute error/true value;

Sensitivity and Conditioning

A problem is said to be insensitive, or well-conditioned, if a given relative change in the input data causes a reasonably commensurate relative change in the solution. A problem is said to be sensitive, or ill-conditioned, if the relative change in the solution can be much larger than that in the input data.

$$Cond = \frac{|\text{relative change in solution}|}{|\text{relative change in input data}|} = \frac{|(f(\hat{x}) - f(x))/f(x)|}{|(\hat{x} - x)/x|};$$

Forward,Backward Error Analysis

- Forward error: Suppose we compute the value of $y = f(x)$, where $f : R \to R$, but we obtain $\hat{y}$.

$$\Delta y = \hat{y} - y$$

- Backward error: Consider the approximate solution to be the exact solution for a modified. The backward error is how much data error in the initial input would be required to explain all of the error in the final computed result.

$$\Delta x = \hat{x} - x, \text{where } f(\hat{x}) = \hat{y}.$$

$$\begin{tikzcd}
x \arrow[r, "f"] \arrow[dr, "\hat{f}"] \arrow[d, leftrightarrow, "\text{backward error}"'] & f(x) \arrow[d, leftrightarrow, "\text{forward error}"] \\
\hat{x} \arrow[r, "f"] & f(\hat{x}) = \hat{f}(x)
\end{tikzcd}$$

Example

We approximate the cosine function $y = f(x) = cos(x)$ for $x = 1$. The cosine function is given by:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots ,$$

We truncate the series after two terms and get the approximation:

$$\hat{y} = \hat{f}(x) = 1 - x^2/2.$$

The forward error is given by

$$\Delta y = \hat{y} - y = \hat{f}(x) - f(x) = 1 - x^2/2 - \cos(x).$$

For $x = 1$, foward error $\Delta y = \hat{y} - y \approx 0.5 - 0.5403 = -0.0403$

To get the backward error, we must find the input value $\hat{x}$ for f that given the output $\hat{y}$.

$$\hat{x} = \arccos(\hat{f}(x)) = \arccos(\hat{y}).$$

For $x = 1$, we have

$$y = f(1) = \cos(1) \approx 0.5403$$

$$\hat{y} = \hat{f}(1) = 1 - 1^2/2 = 0.5$$

$$\hat{x} = \arccos(\hat{y}) = \arccos(0.5) \approx 1.0472$$

Backward error $\Delta x = \hat{x} - x \approx 1.0472 - 1 = 0.0472$.

Summarize the desirable characteristics that such software should possess

- Reliability: always works correctly for easy problems
- Robustness: usually works for hard problems, but fails gracefully and informatively when it does fail
- Accuracy: produces results as accurate as warranted by the problem and input data, preferably with an estimate of the accuracy achieved
- Efficiency: requires execution time and storage that are close to the minimum possible for the problem being solved
- Maintainability: is easy to understand and modify
- Portability: adapts with little or no change to new computing environments
- Usability: has a convenient and well-documented user interface
- Applicability: solves a broad range of problems