An Introduction to Data Analysis and Bayesian Inference

Peter Sykacek[1]

Department of Biotechnology
Bioinfromatics Research Group
BOKU University
peter.sykacek@boku.ac.at

The Next Three Hours

- Good data analysis practise
- Why should you bother?
- Matrices and preliminaries for data analysis
- Data analysis
- Bayesian concepts
- Priors, likelihoods and inference
- Bayesian view of the t-test
- Summary and outlook

Good Analysis Practise

Requires to match data analysis (sometimes simple is too simplistic) to the application domain.

Example: Proove hypothetical genes by measurements.

What's wrong with using an RNA mix of K biological states, hybridising N arrays and declaring all genes as verified, if $n < N$ arrays show expression above a threshold δ?

1)Motivation of n - why $n = 6$ and not one more or less?

2)Motivation of δ - how is it specified?

3)We know a-priori that certain genes (e.g. regulators) show much smaller expression than others and are sometimes only involved in a few processses. The required expression level and dilution will bias the proof towards highly expressed and often used genes!

Data analysis does not fit the objective. $->$ Benchmark your ideas! (e.g. Do you produce more false negatives among known regulators?)

Good Biological Practise

Requires to match data analysis (sometimes advanced methods do not consider biological needs) to the application domain.

Example: determine functional genes.

What's wrong with using SVM (support vector machine, a powerful classifier) and some greegy search to select some optimal gene set for cancer predition and implying that this points to functionally important genes?

1) SVM does typically not outperform a much simpler linear classifier using a single gene.

2) Greedy search provides some set working well for the classification task but certainly without any claim for completeness

3) The gene set provides no ranking of functionally important genes.

The otherwise useful approach (as a diagnostic tool) fails answering the biological question.

Matrices

Important to simplify notation!
Definition n-dimensional Euclidian Space $\mathbb{R}^n$:

$$\mathbb{R}^n = \underbrace{\mathbb{R}\times\mathbb{R}\times\cdots\times\mathbb{R}}_{n \text{ times}} \quad -> \text{Cartesian product}$$

Definition of a matrix (n rows, m columns):

$$\boldsymbol{M} = \begin{pmatrix} m_{1,1} & \cdots & m_{1,m} \\ \vdots & \ddots & \vdots \\ m_{n,1} & \cdots & m_{n,m} \end{pmatrix} = (\boldsymbol{m}_1, \cdots, \boldsymbol{m}_m) \text{ and } \boldsymbol{m}_i \in \mathbb{R}^n$$

MatLab: $>> M = [[a,b,c];[d,e,f];...];$ What are the $\boldsymbol{m}_i$?

Matrix Operations

Transposition: $\boldsymbol{B} = \boldsymbol{A}^T,\ \forall n,m : b_{m,n} = a_{n,m}$
MatLab: $>> B = A';$

Addition ($\boldsymbol{A}$, $\boldsymbol{B}$ equal size):
$\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{B},\ \forall n,m : c_{n,m} = a_{n,m} + b_{n,m}$
MatLab: $>> C = A + B;$
Associative and commutative?

Matrix times constant:
$\boldsymbol{B} = \lambda\boldsymbol{A}\ \forall n,m : b_{n,m} = \lambda a_{n,m}$
MatLab: $>> B = lambda * A;$
Associative and commutative?

Matrix Multiplication I

Matrix Inner Product ($\boldsymbol{A}$'s column no. equals $\boldsymbol{B}$'s row no.): $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B}\ \forall n,m :\ c_{n,m} = \sum_i a_{n,i} b_{i,m}$
MatLab: $>> C = A * B;$
Associative and commutative?
Note: $(\boldsymbol{A}\boldsymbol{B})^T = \boldsymbol{B}^T\boldsymbol{A}^T$
However: $(\boldsymbol{A} + \boldsymbol{B})^2 = \boldsymbol{A}^2 + \boldsymbol{A}\boldsymbol{B} + \boldsymbol{B}\boldsymbol{A} + \boldsymbol{B}^2$

Hadamard Product ($\boldsymbol{A}$, $\boldsymbol{B}$ equal size):
$\boldsymbol{C} = \boldsymbol{A} \cdot \boldsymbol{B},\ \forall n,m : c_{n,m} = a_{n,m} b_{n,m}$
MatLab: $>> C = A. * B;$
Associative and commutative?

Matrix Multiplication II

Kronecker (or tensor) Product:
$\boldsymbol{C} = \boldsymbol{A} \otimes \boldsymbol{B}\ \forall n,m :\ \boldsymbol{C}_{n,m} = a_{n,m}\boldsymbol{B}$
MatLab: $>> C = kron(A, B);$

$\boldsymbol{C}_{n,m}$ are submatrices of dimensions equal to $\boldsymbol{B}$

$\boldsymbol{C}$ has thus $n_A + n_B$ rows and $m_A + m_B$ columns

The Kronecker product is associative:
$\boldsymbol{A} \otimes (\boldsymbol{B} \otimes \boldsymbol{C}) = (\boldsymbol{A} \otimes \boldsymbol{B}) \otimes \boldsymbol{C}$
Is it commutative?

Matrix product of a Kronecker poduct:
$(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = (\boldsymbol{A}\boldsymbol{C}) \otimes (\boldsymbol{B}\boldsymbol{D})$

Square Matrices

Rank of a matrix $r(\boldsymbol{A})$: number of linearly independent columns (or rows, that's the same) of $\boldsymbol{A}$.

Square matrix $\boldsymbol{A}$ is a square matrix if no. rows equals no. cols, that is: $n = m$.

Square matrix $\boldsymbol{A}$ is non-singular if rank $r(\boldsymbol{A}) = n$.

Determinant of a Square Matrix

Determinant of a matrix:

$$|\boldsymbol{A}| = \sum_{\text{all permutations of } (1,..,n)} (-1)^{\Phi(j_1,..,j_n)} \prod_{i=1}^{n} a_{i,j_i}$$

where $\Phi(j_i,..,j_n)$ is the number of transpositions (interchanging two numbers) required to transform $(1,..n)$ into $(j_1,..,j_n)$. This is a consistent definition since the number of transpositions is always even or odd!
MatLab: $>> det(A)$

Remarks Regarding Determinants

$|\boldsymbol{A}| = 0$ implies that $\boldsymbol{A}$ is singular

$|\boldsymbol{A}| = \prod_n \lambda_n$, where λ_n are the eigenvalues of $\boldsymbol{A}$

if $|\boldsymbol{A}|$ is an upper or lower diagonal matrix:
$|\boldsymbol{A}| = \prod_{i=1}^{n} a_{i,i}$

Highschool math - recursive definition w.r.t j-th row (works similarly for the i-th column):

$$|\boldsymbol{A}| = \sum_{i} (-1)^{i+j} a_{i,j} |\boldsymbol{A}_{i,j}|$$

$|\boldsymbol{A}_{i,j}|$ is the determinant of the submatrix when removing the j-th row and i-th column.

Diagonal and Identity Matrices

Diagonal matrix: $\boldsymbol{A} = \mathrm{diag}(a_{1,1},..,a_{n,n})$, defines a matrix with the only non zero elements located on the main diagonal
MatLab: $>> A = diag(a);$ % places $\boldsymbol{a}$ into main diagonal of $\boldsymbol{A}$
$>> a = diag(A);$ % places main diagonal of $\boldsymbol{A}$ into $\boldsymbol{a}$

Identity matrix: $\boldsymbol{I} = \mathrm{diag}(1,..,1)$
neutral element of matrix multiplication:
$\boldsymbol{IA} = \boldsymbol{AI} = \boldsymbol{A}$
MatLab: $>> I = eye(n);$ % generates an $[n \times n]$ identity matrix.

Inverse Matrix

If matrix $\boldsymbol{A}$ is non-singular, we get a non-singular matrix $\boldsymbol{B} = \boldsymbol{A}^{-1}$, such that, $\boldsymbol{B}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{B} = \boldsymbol{I}$.
Matrix $\boldsymbol{B}$ is the inverse of $\boldsymbol{A}$

MatLab:$>> B = A^{-1}$

Remarks: $(\boldsymbol{A}^{-1})^T = (\boldsymbol{A}^T)^{-1}$ and
$(\boldsymbol{A}\boldsymbol{B})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$
Matrix $\boldsymbol{A}$ is orthonormal if $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$, hence
$\boldsymbol{A}^T = \boldsymbol{A}^{-1}$
Examples: projection to principal axis (PCA)
Permutation matrix $\boldsymbol{P}$ in every row and column one 1 entry, otherwise 0

Using Inverse Matrices

Consider:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

with $\boldsymbol{A}$ square $[n \times n]$, then:

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

Note that this type of operation is typically found in many data analysis scenarios, e.g. in finding least squares solutions.

Moore Penrose Pseudo Inverse

Inverting ill conditioned (close to singular) matrices involves quantities close to machine precision. Results derived from such inverse matrices result in large numerical errors.

Pracactical rule - never use matrix inversion, always use the Moore Penrose pseudo inverse.

$$\boldsymbol{A}^{+} = \lim_{\delta \to 0}(\boldsymbol{A}^T\boldsymbol{A} + \delta\boldsymbol{I})^{-1}\boldsymbol{A}^T$$

MatLab: $>> A\_plus = pinv(A);$
If $\boldsymbol{A}$ square and not ill-conditioned:
$\boldsymbol{A}^{+}\boldsymbol{A} = \boldsymbol{A}^{-1}(\boldsymbol{A}^T)^{-1}\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$

Sherman-Morrison-Woodbury Formula

Data analysis sometimes (e.g. Kalman filter) requires inverting matrices incrementally. We know $\boldsymbol{A}^{-1}$ and have two $[n \times p]$ matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ with $p \ll n$ and seek $(\boldsymbol{A} + \boldsymbol{U}\boldsymbol{V}^T)^{-1}$. The following matrix inversion lemma helps:

$$(\boldsymbol{A}+\boldsymbol{U}\boldsymbol{V}^T)^{-1} = \boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{I}+\boldsymbol{V}^T\boldsymbol{A}^{-1}\boldsymbol{U})^{-1}\boldsymbol{V}^T\boldsymbol{A}^{-1}$$

Inversion then only requires inverting a $[p \times p]$ matrix. For a column vector $\boldsymbol{x}$, a further simplification of $(\boldsymbol{A} + \boldsymbol{x}\boldsymbol{x}^T)^{-1}$ is possible. Why??

Matrices and Data Analysis

Just to make sure that you see the connection between matrices and data analysis, here an example:

Assume N samples of k “input” measurements collected in $\boldsymbol{x}_n$ and one dependent variable y_n, which we intend modelling as a function $f(\boldsymbol{x}_n, \boldsymbol{\Theta})$ parameterised by $\boldsymbol{\Theta}$. This type of modelling is called *regression*.

We can only move on deciding on a particular $f(\boldsymbol{x}_n, \boldsymbol{\Theta})$.

For simplicity we assume that the best guess of y_n is obtained as linear combination of $\boldsymbol{x}_n$. This allows writing:

$$y_n = \sum_k \boldsymbol{x}_n[k]\boldsymbol{\Theta}[k], \text{ or } y_n = \boldsymbol{x}_n^T\boldsymbol{\Theta}$$

Model fitting I

is, withthout giving this much thought, often done by “minimising least squares differences”

$$\hat{\boldsymbol{\Theta}} = \mathrm{argmin}_{\boldsymbol{\Theta}} \left(\sum_n \left(y_n - \boldsymbol{x}_n^T\boldsymbol{\Theta} \right)^2 \right)$$

Importance of thinking in terms of matrices:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_N^T \end{pmatrix} \text{ and } \boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_N \end{pmatrix}$$

Model fitting II

We can then immediately write

$$\mathrm{lsd} = \sum_n \left(y_n - \boldsymbol{x}_n^T\boldsymbol{\Theta} \right)^2 = (\boldsymbol{X\Theta} - \boldsymbol{y})^T(\boldsymbol{X\Theta} - \boldsymbol{y})$$

which contains no sums any more and is an extremely convenient method for deriving model fitting procedures and code for numerical tools like MatLab.

MatLab:$>> y\_d = X * theta - y;$

$>> LSD = y\_d' * y\_d;$

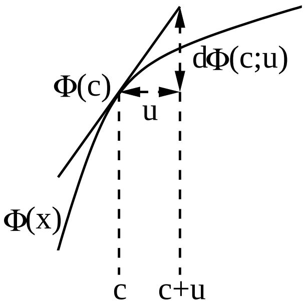
two lines to be more efficient!

Matrix Functions

Scalar functions map vectors and matrices to $\mathbb{R}$			
a)	$\Phi(\boldsymbol{x})$	$\mathrm{S} \in \mathbb{R}^n$	$\mapsto \mathbb{R}$
b)	$\Phi(\boldsymbol{A})$	$\mathrm{S} \in \mathbb{R}^{[n \times m]}$	$\mapsto \mathbb{R}$
a) Fourier synthesis, b) $\lVert\boldsymbol{A}\rVert$, quadratic form $\boldsymbol{x}^T\boldsymbol{Ax}$			
Vector functions map vectors and matrices to $\mathbb{R}^q$			
c)	$f(\boldsymbol{x})$	$\mathrm{S} \in \mathbb{R}^n$	$\mapsto \mathbb{R}^q$
d)	$f(\boldsymbol{A})$	$\mathrm{S} \in \mathbb{R}^{[n \times m]}$	$\mapsto \mathbb{R}^q$
c) linear projection $\boldsymbol{Ax}$ (parameter $\boldsymbol{x}$), d) $\mathrm{tr}\boldsymbol{A}$			
Matrix functions map vectors and matrices to $\mathbb{R}^{[q \times p]}$			
e)	$F(\boldsymbol{x})$	$\mathrm{S} \in \mathbb{R}^n$	$\mapsto \mathbb{R}^{[q \times p]}$
f)	$F(\boldsymbol{A})$	$\mathrm{S} \in \mathbb{R}^{[n \times m]}$	$\mapsto \mathbb{R}^{[q \times p]}$
e) $\boldsymbol{xx}^T$ (expand this!), f) the inverse matrix $\boldsymbol{A}^{-1}$			

The vec opeartor (stacking of column vectors), converts matrix functions to vector functions $f(\boldsymbol{A}) = \mathrm{vec}\ F(\boldsymbol{A})$.

Definition Differential



Derivative: $\Phi'(c) = \lim_{u \to 0} \frac{\Phi(c+u)-\Phi(c)}{u}$ implies a linear approximation of $\Phi(x)$ at c:

$\Phi(c+u) = \Phi(c) + d\Phi(c;u) + r_c(u)$

The differential $d\Phi(c;u)$ is the difference between $\Phi(c)$ and $\Phi(c+u)$ based on a linear expansion around c.

First identification theorem: $d\Phi(c;u) = \Phi'(c)u$

Least Squares Optimum I

is, as previously discussed, obtained as

$$\hat{\boldsymbol{\Theta}} = \mathrm{argmin}_{\boldsymbol{\Theta}}((\boldsymbol{X}\boldsymbol{\Theta} - \boldsymbol{y})^T(\boldsymbol{X}\boldsymbol{\Theta} - \boldsymbol{y}))$$

Analogous to high school optimisation, we get the $\mathrm{argmin}_{\boldsymbol{\Theta}}$ by setting the gradient of $\Phi(\boldsymbol{\Theta}) = ((\boldsymbol{X}\boldsymbol{\Theta} - \boldsymbol{y})^T(\boldsymbol{X}\boldsymbol{\Theta} - \boldsymbol{y}))$ zero and solving for $\boldsymbol{\Theta}$. Using $\boldsymbol{e} = (\boldsymbol{X}\boldsymbol{\Theta} - \boldsymbol{y})$, we require the differential $d(\boldsymbol{e}^T\boldsymbol{e})$:

$$d(\boldsymbol{e}^T\boldsymbol{e}) = (d\boldsymbol{e}^T)\boldsymbol{e} + \boldsymbol{e}^T(d\boldsymbol{e}) = 2\boldsymbol{e}^T(d\boldsymbol{e})$$

$$= 2(\boldsymbol{X}\boldsymbol{\Theta} - \boldsymbol{y})^T\boldsymbol{X}d\boldsymbol{\Theta}$$

Least Squares Optimum II

The first identification theorem gives now the Jacobian matrix (actually a row vector):

$$D\Phi(\boldsymbol{\Theta}) = 2(\boldsymbol{\Theta}^T\boldsymbol{X}^T - \boldsymbol{y}^T)\boldsymbol{X}$$

and thus the gradient:

$$\nabla\Phi(\boldsymbol{\Theta}) = 2(\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\Theta} - \boldsymbol{X}^T\boldsymbol{y}).$$

The solution is thus:

$\boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{\Theta}} = \boldsymbol{X}^T\boldsymbol{y}$ or $\hat{\boldsymbol{\Theta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$

Does this look familiar?

Exponential Function and Logarithm

Two important functions in data analysis are the exponential function and the logarithm.

$y = \exp(x)$ and the inverse $x = \log(y)$

Important Relations:

$$\begin{aligned}
&\exp(a)\exp(b) = \exp(a+b)\\
&\frac{\exp(a)}{\exp(b)} = \exp(a-b)\\
&(\exp(a))^n = \exp(na)\\
&\log(ab) = \log(a) + \log(b)\\
&\log(\frac{a}{b}) = \log(a) - \log(b)\\
&x^\alpha = \exp(\alpha\log(x))
\end{aligned}$$

Gamma and Digamma Functions

that occur occasionally in Bayesian data analysis. Commonality: only implicit definitions with numerical implementations in most numerical packages.

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} \exp(-x) dx$$

is known as *gamma function*. A related function is the *digamma function*

$$\Psi(\alpha) = \frac{d}{dx} \log(\Gamma(x))|_{x=\alpha} = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)}$$

Integrals

Bayesian data analysis is inherently coupled with solving integrals. There are two ways dealing with those:

1) Solve either exact or approximate versions of the integral analytically.
2) Solve by Monte Carlo Integration, i.e. by

$$f(x) = \int_{\boldsymbol{\Theta}} f(x;\boldsymbol{\Theta}) p(\boldsymbol{\Theta}) d\boldsymbol{\Theta} \approx \frac{1}{N} \sum_{n=1}^{N} f(x;\boldsymbol{\Theta}_n),$$

where $\boldsymbol{\Theta}_n \sim p(\boldsymbol{\Theta})$.

1) has the disadvantage of being analytically more challenging and often requiring *systematic approximations*. solutions are though computationally much less involved than 2).

The general rule for 1) is avoiding integration by transformations to known integrals.

Example for Avoiding Integration

An approximation technique useful for 1) leads in many situations to the following integral:

$$\int_{\lambda=0}^{\infty} \log(|\lambda|) \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{(\alpha-1)} \exp(-\beta\lambda) d\lambda$$

this is the expectation of $\log(\lambda)$ under a *Gamma distribution*. Trick: $\log(\lambda)\lambda^{(\alpha-1)} = \frac{d}{d\alpha}\lambda^{(\alpha-1)}$, we solve:

$$\frac{\beta^\alpha}{\Gamma(\alpha)} \frac{d}{d\alpha} \underbrace{\int_{\lambda=0}^{\infty} \lambda^{(\alpha-1)} \exp(-\beta\lambda)}_{\text{1/normalisation constant !!}}$$

Avoiding Interation Ctd.

The initial Integral is therefore eqivalent to

$$\frac{\beta^\alpha}{\Gamma(\alpha)} \frac{d}{d\alpha} \frac{\Gamma(\alpha)}{\beta^\alpha}$$

or

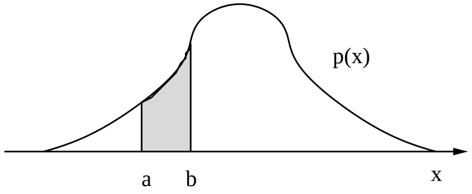
$$\frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\Gamma'(\alpha)\beta^\alpha + \Gamma(\alpha)\log(\beta)\beta^\alpha}{\beta^{2\alpha}} = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} + \log(\beta)$$

and finally $\Psi(\alpha) + \log(\beta)$, where Ψ is the digamma function.

Random Variable and PDF

Random variable: a non deterministic quantity where repeated observations being different though generated according to some overall property. Properties of random variables are for example captured by an associated probability density function (pdf).

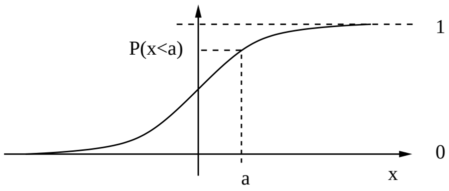
The pdf allows deducing the probability that a new realisation falls into a particular set, $P(x \in [a, b]) = \int_{x=a}^{b} p(x)dx$.



Cumulative Distribution Function

An equivalent characterisation for univariate random variables is provided by the so called cumulative distribution function (cdf).

The cdf $F(x)$ denotes the probability that a realisation of the random variable is smaller than x. $F(a)$ is thus the probability $P(x < a)$.



$- >$ the pdf $p(x) = \frac{dF(\xi)}{d\xi}|_{\xi=x}$ is the derivative of the cdf at x.

Essential Rules of Probability Calculus

- If A and B represent mutually exclusive events with probabilities $P(A)$ and $P(B)$, the probability that either event occurs is $P(A) + P(B)$.
- The joint probability over A and B is: $P(A, B) = P(A)P(B|A) = P(B)P(A|B)$. If A and B are independent we have $P(B|A) = P(B)$ and $P(A|B) = P(A)$.
- Given $P(A, B, C) = P(A)P(B|A)P(C|A, B)$, we obtain $P(C, A) = \int_B P(A, B, C)dB$ by integration (here also referred to as marginalisation).

Why Bother With Data Analysis?

Moore's Law:

PC 1984	5 MB Hard Drive
PC 2007	2 TB Hard Drive (4*500 GB) ≈ 400 Euro

How much paper on one PC in 2007 assuming 10.000 (single byte) characters per page ?

Why Bother With Data Analysis?

Moore's Law:

PC 1984 5 MB Hard Drive

PC 2007 2 TB Hard Drive (4*500 GB) $\approx$ 400 Euro

How much paper on one PC in 2007 assuming 10.000 (single byte) characters per page ?

It is actually a stack of paper 20 km high!

2 TB $\approx$ $2 * 10^{12}$ byte

$= 2 * 10^8$ pages, assuming 1000 pages = 10 cm

a stack $2 * 10^5 * 10$ cm = $2 * 10^4$ m = 20 km

What About Data Generation?

Medical monitoring 1:

20 channels EEG+physiological signals 8 hours sleep at 200 Hz and 16 Bit :

$20 * 8 * 3600 * 200 * 2 \approx 230,410^6$ byte ≈ 250 MB.

A single sleep lab with 8 recording units, operated at nights only, will generate one TB in just over a year.

What About Data Generation?

Medical monitoring 1:

20 channels EEG+physiological signals 8 hours sleep at 200 Hz and 16 Bit :

$20 * 8 * 3600 * 200 * 2 \approx 230,410^6$ byte ≈ 250 MB.

A single sleep lab with 8 recording units, operated at nights only, will generate one TB in just over a year.

Medical monitoring 2:

An FMRI scanner, 1dm^3 volume, 10s temporal and 1mm^3 spatial resolution, 16 bit.

One scanner generates $10^6 * 360 * 2$ byte ≈ 720 MB per hour which fills 1 TB in about 58 days.

What About Data Generation?

Medical monitoring 1:

20 channels EEG+physiological signals 8 hours sleep at 200 Hz and 16 Bit :

$20 * 8 * 3600 * 200 * 2 \approx 230,410^6$ byte ≈ 250 MB.

A single sleep lab with 8 recording units, operated at nights only, will generate one TB in just over a year.

Medical monitoring 2:

An FMRI scanner, 1dm^3 volume, 10s temporal and 1mm^3 spatial resolution, 16 bit.

One scanner generates $10^6 * 360 * 2$ byte ≈ 720 MB per hour which fills 1 TB in about 58 days.

High throughput molecular biology:

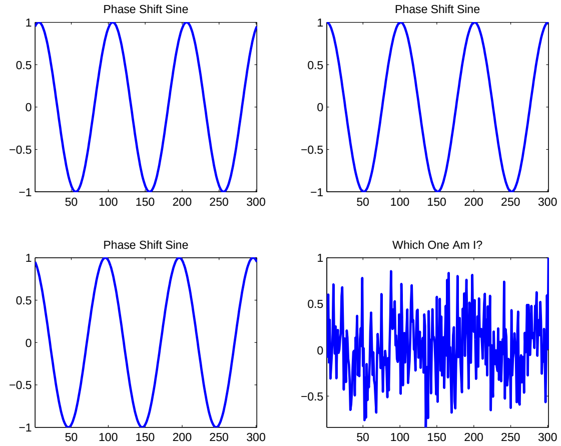
A small lab produces up to 12 slides per 24 hours. One slide can contain up to 30.000 probes with ≈ 300 pixels/probe at 16 bit. Since we scan the entire array this is about 240 MB per 24 hours.

Such data can for two reasons not be analysed manually:

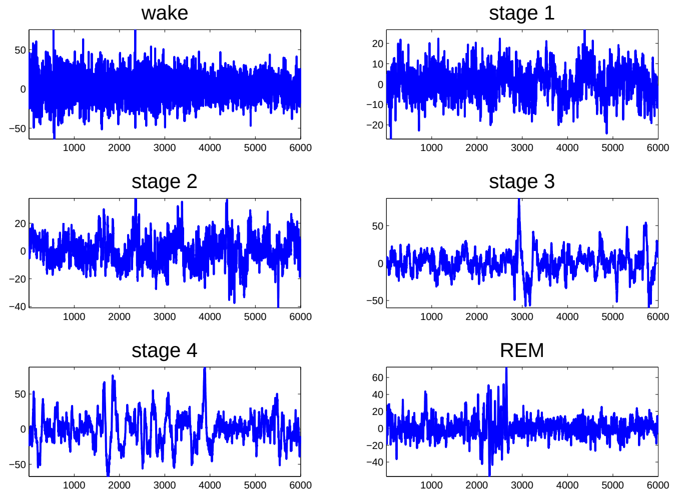
Amount and “Noise”

Manual Analysis Task

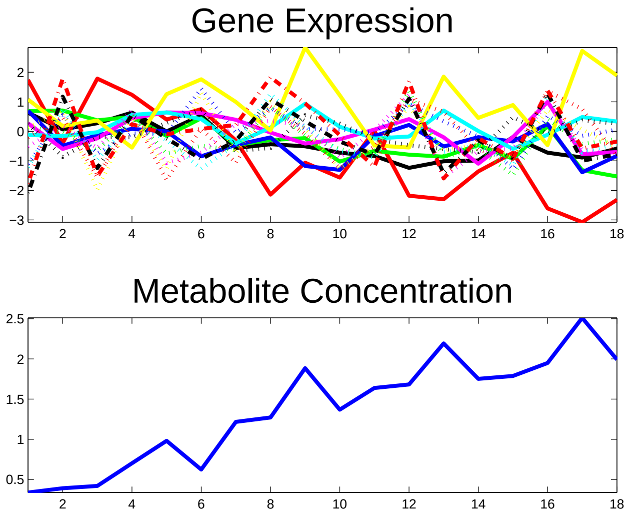
Which sine wave has the correct phase?



Example: Sleep EEG



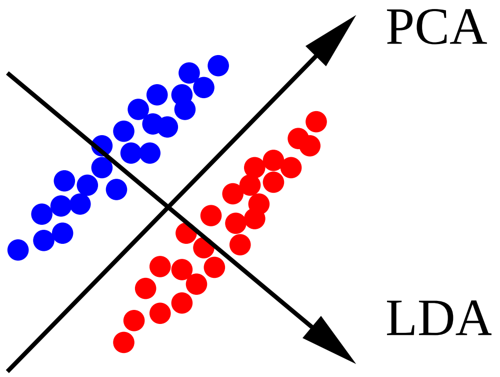
Example: Metabolomics



Why Understand Data Analysis?

Result = Data + Model!

Linear discriminant and principle component analysis can provide orthogonal projections of the same data.



Two Scenarios in Applied Life Sciences

1. Given measurements $\boldsymbol{x}_n$ and some corresponding dependent information y_n, we might ask: How are they related?

Two Scenarios in Applied Life Sciences

1. Given measurements $\boldsymbol{x}_n$ and some corresponding dependent information y_n, we might ask: How are they related?

2. Given two sets of measurements $\boldsymbol{x}_n$ and $\boldsymbol{z}_n$, we might ask: Which of those are closer related to some corresponding dependent information y_n?

Two Scenarios in Applied Life Sciences

1. Given measurements $\boldsymbol{x}_n$ and some corresponding dependent information y_n, we might ask: How are they related?
2. Given two sets of measurements $\boldsymbol{x}_n$ and $\boldsymbol{z}_n$, we might ask: Which of those are closer related to some corresponding dependent information y_n?

$->$ two instances of “inference” commonly found in applied life sciences.

We do for the moment ignore the problem where we have only some measurements $\boldsymbol{x}_n$ and ask how they are structured.

First Scenario

Suppose a life science experiment provided some noisy data $\mathcal{Z} = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_N, y_N)\}$. Note: $\boldsymbol{x}_n$ possibly multivariate i.e. vectors.

Based on $\mathcal{Z}$, we have an inference problem of finding an “optimal” relation between $\boldsymbol{x}$ and y:

$$p(y|\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{\theta}) + \epsilon(\lambda)$$

First Scenario

Suppose a life science experiment provided some noisy data $\mathcal{Z} = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_N, y_N)\}$. Note: $\boldsymbol{x}_n$ possibly multivariate i.e. vectors.

Based on $\mathcal{Z}$, we have an inference problem of finding an “optimal” relation between $\boldsymbol{x}$ and y:

$$p(y|\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{\theta}) + \epsilon(\lambda)$$

Noise requires a deterministic and a random component.

$->$ Inherent uncertainty, y is a random variable!

Inference

Parameter Inference:

Implies knowing $f(\boldsymbol{x}; \boldsymbol{\theta})$ and the noise model $\epsilon(\lambda)$ up to unknown parameters ($\boldsymbol{\theta}$ and λ) which we will be inferring from data.

Inference

Parameter Inference:

Implies knowing $f(\boldsymbol{x}; \boldsymbol{\theta})$ and the noise model $\epsilon(\lambda)$ up to unknown parameters ($\boldsymbol{\theta}$ and λ) which we will be inferring from data.

Model Inference:

A more realistic assumption is that the model class is unknown and we will be inferring model class and parameters.

Assessing Model Parameters

Idea: subtract the deterministic part from y_n:

$$\epsilon_n = y_n - f(\boldsymbol{x}_n; \boldsymbol{\theta})$$

For convenience introduce $\mathcal{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ and $\mathcal{D} = \{y_1, ..., y_N\}$. Assuming that ϵ_n are i.i.d samples, we get the likelihood function:

$$p(\mathcal{D}|\boldsymbol{\theta}, \lambda, \mathcal{X}) = \prod_n p(y_n|\boldsymbol{\theta}, \lambda, \boldsymbol{x}_n)$$

which is a suitable objective function to be maximized for $\boldsymbol{\theta}$ and λ.

Likelihood and Linear Regression

Assuming N samples, we have:

$$p(y_n|\boldsymbol{x}_n;\boldsymbol{\theta},\lambda) = (2\pi)^{-0.5}\lambda^{0.5}\exp(-0.5\lambda(y_n - \boldsymbol{x}_n^T\boldsymbol{\theta})^2) \text{ and}$$
$$p(\mathcal{D}|\mathcal{X};\boldsymbol{\theta},\lambda) = (2\pi)^{-\frac{N}{2}}\lambda^{\frac{N}{2}}\exp(-0.5\lambda\sum_n(y_n - \boldsymbol{x}_n^T\boldsymbol{\theta})^2)$$

Taking the $\log$, we get the log likelihood:

$$\text{llh} = \frac{N}{2}(\log(\lambda)-\log(2\pi))-0.5\lambda(\boldsymbol{y}-\boldsymbol{X\theta})^T(\boldsymbol{y}-\boldsymbol{X\theta})$$

which, if we consider maximising for $\boldsymbol{\theta}$ only, is a familiar expression.

$>$ minimising least squares assumes Gaussian noise!

Likelihood and Classification

“Classification” often used synonymously for regression with discrete outcomes. Likelihood of regression model:

$$P(\mathcal{D}|\mathcal{X};\boldsymbol{\theta}) = \prod_n P(y_n|\boldsymbol{x}_n,\boldsymbol{\theta})$$

To enforce $\sum_{y_n} P(y_n|\boldsymbol{x}_n,\boldsymbol{\theta})$ is 1, we apply a suitable output transformation, e.g. the cdf of the logistic distribution:

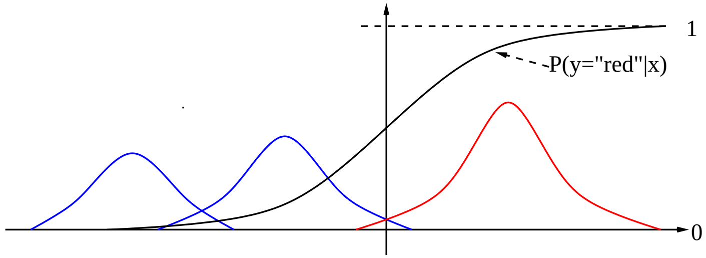
$$P(y_n|\boldsymbol{x}_n^T\boldsymbol{\theta}) = \frac{1}{1+\exp((2y_n-1)\boldsymbol{x}_n^T\boldsymbol{\theta})}$$

Probabilities are certainty measures about classes to avoid ignorant decisions:
Surgeon: Amputate or not?
Nurse: The SVM says +1.

Classification and Sampling Paradigm

$$P(y_n|\boldsymbol{x}_n) = \frac{P(y_n)p(\boldsymbol{x}_n|y_n)}{p(\boldsymbol{x}_n)}$$

$->$ Bayes theorem suggests that we can also model class priors $P(y_n)$ and class conditional densities $p(\boldsymbol{x}_n|y_n)$.



vantage: a useful density model, disadvantage: more complicated

A Major Problem

True model - linear regression, Gaussian noise:

$$p(y|\boldsymbol{x}) = f(\boldsymbol{x};\boldsymbol{\theta}) + \epsilon(\lambda)$$

$f(\boldsymbol{x};\boldsymbol{\theta}) = [1,\boldsymbol{x}^T]\boldsymbol{\theta}$ and $\epsilon(\lambda) = \mathcal{N}(\epsilon;0,\lambda)$, with λ denoting “precision” (i. e. inverse variance).

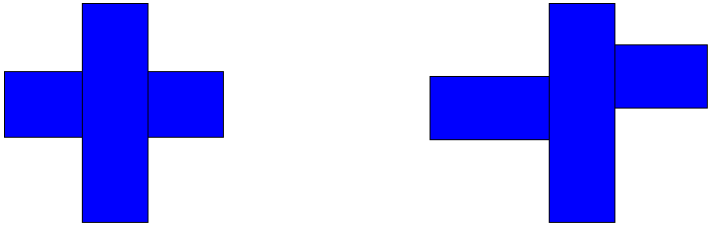
Finite sample size and different model classes: What is the maximum of the likelihood?

Think “phone book”: Perfect memorizing of all y_n, modelling error 0, $\lambda->\infty$, $p(\mathcal{D}|\boldsymbol{\theta},\lambda,\mathcal{X})->\infty$.

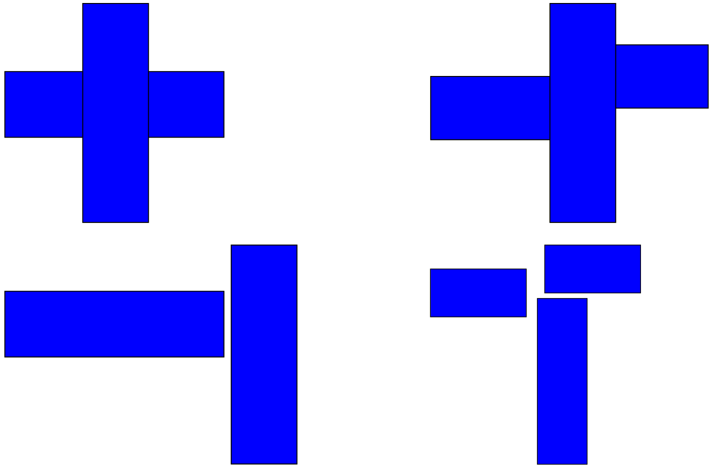
$->$ likelihood unsuitable objective for model inference!

Why is memorizing useless?

Guess the Correct “Model”



Guess the Correct “Model”



Model comparison requires penalties on top of likelihood! (AIC, BIC, etc.)

Occam’s Razor

We implicitly apply Occam’s Razor

William of Occam (or Ockham) (1288 - 1348)

Entia non sunt multiplicanda sine necessitate: Entities are not to be multiplied without necessity.

Interpretation: One should always opt for an explanation in terms of the fewest possible number of causes, factors, or variables.

Material from `http://en.wikipedia.org/wiki/William_of_Ockham`.

Bayesian Inference

Thomas Bayes (1701 - 1763)

Occam’s Razor built in!
Two important consequences for “learning from data”. Inference based on a decision theoretic framework

Bayesian Inference

Thomas Bayes (1701 - 1763)

Occam's Razor built in!
Two important consequences for "learning from data". Inference based on a decision theoretic framework

$p(I|\mathcal{D}) = \frac{p(\mathcal{D}|I)p(I)}{p(\mathcal{D})}$

1) Revise beliefs by Bayes theorem

Bayesian Inference

Thomas Bayes (1701 - 1763)

Occam's Razor built in!
Two important consequences for "learning from data". Inference based on a decision theoretic framework

$p(I|\mathcal{D}) = \frac{p(\mathcal{D}|I)p(I)}{p(\mathcal{D})}$

1) Revise beliefs by Bayes theorem

$\alpha_{opt} = \text{argmax}_\alpha < u(\alpha) >$, where
$< \ u(\alpha) \ >= \ \int_G u(\alpha, I)p(I|\mathcal{D})dI.$

2) Decisions by maximising expected utility

A Bayesian Dice Model - the Likelihood

Goal: inferring probabilities observing sides of a dice, i.e. $\pi = \{\pi_1, .., \pi_5, 1 - \sum_{k=1}^{5} \pi_k\}$
Data: N observations from rolling the dice.

A Bayesian Dice Model - the Likelihood

Goal: inferring probabilities observing sides of a dice, i.e. $\pi = \{\pi_1, .., \pi_5, 1 - \sum_{k=1}^{5} \pi_k\}$
Data: N observations from rolling the dice.

We need a likelihood function:
Throwing the dice once results in a multinomial one distribution over sides, i.e.
$P(I_n|\pi) = \prod_{k=1}^{6} \pi_k^{\delta(I_n=k)}$, where $I_n \in \{1, .., 6\}$.

Independence assumption $->$ likelihood:
$p(\mathcal{D}|\pi) = \prod_n P(I_n|\pi)$, where $\mathcal{D} = \{I_1, ..., I_N\}$ denotes the N outcomes.
What is the final expression of the likelihood?

Bayesian Dice Model - the Prior

We typically use a conjugate prior: a convenient choice to remain within a functional family which is a known distribution. The Multinomial suggests a Dirichlet prior over π:

$$p(\pi) = \frac{\Gamma(\sum_{k=1}^{6} \alpha_k)}{\prod_{k=1}^{6} \Gamma(\alpha_k)} \prod_{k=1}^{6} \pi_k^{\alpha_k - 1}$$

$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} \exp(-x) dx$ is known as gamma function.

Write the definition of $\Gamma(\alpha)$ down! You will need it later during the lecture!

The α_k are hyper parameters of our model.

What is their logical meaning?

Bayesian Dice Model: the Posterior

Multiplying prior and likelihood and renormalising gives the posterior distribution over π as the result of Bayesian inference of the dice model:

$$p(\pi|\mathcal{D}) = \frac{1}{p(\mathcal{D})} \frac{\Gamma(\sum_{k=1}^{6} \alpha_k)}{\prod_{k=1}^{6} \Gamma(\alpha_k)} \prod_{k=1}^{6} \pi_k^{\alpha_k + n_k - 1}$$

where $p(\mathcal{D}) = \int_{\pi_1,..,\pi_6} p(\pi, \mathcal{D}) d\pi$ denotes the marginal likelihood, which is useful for model selection.

What is the functional form of the marginal likelihood ?

Iterative Inference

Given prior counts $\{\alpha_1, ..\alpha_k\}$ and data sets $\mathcal{D}_1 = \{I_1, ..., I_N\}$ and $\mathcal{D}_2 = \{I_{N+1}, ..., I_{N+M}\}$, using $p(\pi|\mathcal{D}_1)$ as prior for $\mathcal{D}_2$ will result in the same posterior $p(\pi|\mathcal{D}_1, \mathcal{D}_2)$ we get from the original prior and the pooled data $\mathcal{D} = \{I_1, .., I_{N+M}\}$:

$$p(\pi|\mathcal{D}_1) = \frac{\Gamma(\sum_k (\alpha_k + n_k))}{\prod_k \Gamma(\alpha_k + n_k)} \prod_k \pi_k^{\alpha_k + n_k - 1}$$

$$p(\pi|\mathcal{D}_1, \mathcal{D}_2) = \frac{\Gamma(\sum_k (\alpha_k + n_k + m_k))}{\prod_k \Gamma(\alpha_k + n_k + m_k)} \prod_k \pi_k^{\alpha_k + n_k + m_k - 1}$$

Since $n_k + m_k$ is the overall number of observations of side k this is equivalent to $p(\pi|\mathcal{D})$.

Applied Bayesian Decision Theory

Horse betting: bet x; choice α; uncertain outcome of race I. Bookmakers “odds” r_A and r_B (one + odds ratio) imply utility function $u(\alpha, I)$:

$\alpha \backslash I$	“A” wins	“B” wins
bet “A”	xr_A	0
bet “B”	0	xr_B
no bet	x	x

Need probability of $I = [A, B]$ i.e. respective horse wins. From previous observations (races) $\mathcal{D}$: $P(I = A|\mathcal{D}) = 0.7$ and $P(I = B|\mathcal{D}) = 0.3$.

Horse Betting ctd.

Calculate expected utility $u(\alpha) = \sum_I u(\alpha, I) P(I|\mathcal{D})$:

bet “A”	bet “B”	no bet
$0.7 x r_A$	$0.3 x r_B$	x

Maximise expected utility!

case	I	II	III
r_A	1.4	1.9	1.3
r_B	3.2	2.5	4.5

What are your decisions?

Horse Betting ctd.

Calculate expected utility $u(\alpha) = \sum_I u(\alpha, I) P(I|\mathcal{D})$:

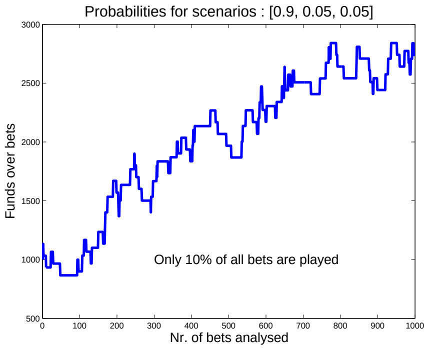
bet “A”	bet “B”	no bet
$0.7 x r_A$	$0.3 x r_B$	x

Maximise expected utility!

case	I	II	III
r_A	1.4	1.9	1.3
r_B	3.2	2.5	4.5

What are your decisions?

Can we earn money?



Inferring a Univariate Gaussian

Data $\mathcal{D} = \{x_1, .., x_N\}$: drawn from a univariate Gaussian with mean μ and precision λ.
Goal: inferring μ and λ, i.e. apply Bayes theorem:

$$p(\mu, \lambda|\mathcal{D}, g, h, l_0) = \frac{p(\mathcal{D}|\mu, \lambda) p(\mu|l_0) p(\lambda|g, h)}{p(\mathcal{D}|g, h, l_0)}$$

What is the precision?

Inferring a Univariate Gaussian

Data $\mathcal{D} = \{x_1, .., x_N\}$: drawn from a univariate Gaussian with mean μ and precision λ.
Goal: inferring μ and λ, i.e. apply Bayes theorem:

$$p(\mu, \lambda|\mathcal{D}, g, h, l_0) = \frac{p(\mathcal{D}|\mu, \lambda) p(\mu|l_0) p(\lambda|g, h)}{p(\mathcal{D}|g, h, l_0)}$$

What is the precision?

Univariate Gaussian distribution:

$$p(x_n|\mu, \lambda) = (2\pi)^{-\frac{1}{2}} |\lambda|^{\frac{1}{2}} \exp\left(-0.5\lambda(x_n - \mu)^2\right)$$

and Likelihood: $p(\mathcal{D}|\mu, \lambda) = \prod_n p(x_n|\mu, \lambda)$

Functional form of the likelihood?

Priors over μ and λ

Likelihood:

$$p(\mathcal{D}|\mu,\lambda) = (2\pi)^{-\frac{N}{2}}|\lambda|^{\frac{N}{2}} \exp\left(-0.5\lambda(N\mu^2 - 2\mu\sum_n x_n + \sum_n x_n^2)\right)$$

Conjugate prior for μ ?

Priors over μ and λ

Likelihood:

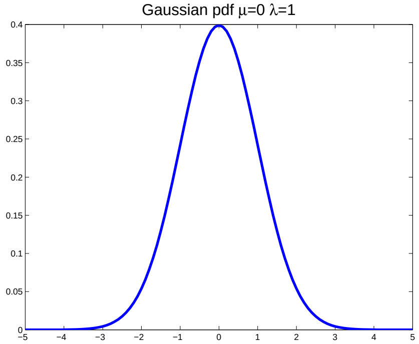
$$p(\mathcal{D}|\mu,\lambda) = (2\pi)^{-\frac{N}{2}}|\lambda|^{\frac{N}{2}} \exp\left(-0.5\lambda(N\mu^2 - 2\mu\sum_n x_n + \sum_n x_n^2)\right)$$

Conjugate prior for μ ?

Priors:
$p(\mu|l_0) = (2\pi)^{-0.5}|l_0|^{0.5}\exp(-0.5 l_0 \mu^2)$, zero mean Gaussian with precision $l_0 = \gamma\lambda$ “g-prior”
$p(\lambda|g,h) = \frac{h^g}{\Gamma(g)}|\lambda|^{(g-1)}\exp(-h\lambda)$, Gamma distribution with shape g and inverse scale h.

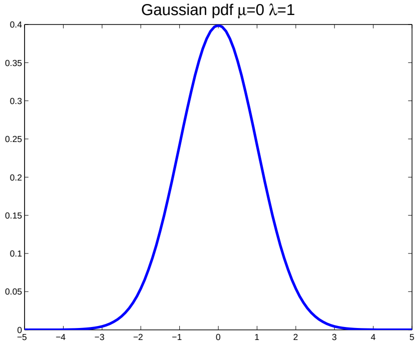
Priors ctd.

Gaussian defined for $x \in \Re$

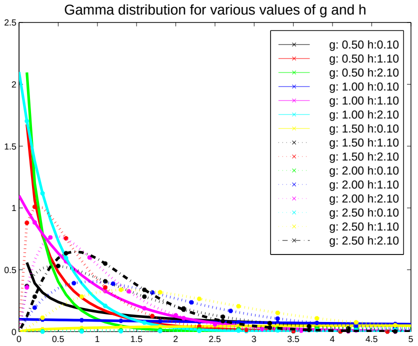


Priors ctd.

Gaussian defined for $x \in \Re$



Gamma defined for $x \in \Re | x > 0$



Prior Times Likelihood

$$p(\mathcal{D},\mu,\lambda|g,h,\gamma) = p(\mathcal{D}|\mu,\lambda)p(\mu|\lambda\gamma)p(\lambda|g,h)$$

$$= (2\pi)^{-\frac{N+1}{2}}\frac{h^g}{\Gamma(g)}|\gamma|^{\frac{1}{2}}|\lambda|^{(\frac{N+1}{2}+g-1)}$$

$$\times \exp\Big(-\lambda\Big(h+0.5\Big((\gamma+N)\mu^2 - 2\mu\sum_n x_n + \sum_n x_n^2\Big)\Big)\Big)$$

$$= (2\pi)^{-\frac{N+1}{2}}\frac{h^g}{\Gamma(g)}|\gamma|^{\frac{1}{2}}|\lambda|^{(\frac{N+1}{2}+g-1)}$$

$$\times \exp\Big(-\lambda\Big(h+0.5\Big(\Big(\sum_n x_n^2 - \frac{(\sum_n x_n)^2}{\gamma+N}\Big)\Big)\Big)\Big)$$

$$\times \exp\Big(-\lambda 0.5(\gamma+N)\Big(\mu - \frac{\sum_n x_n}{\gamma+N}\Big)^2\Big)$$

For normalisation, integrate over λ and μ.

Integrating out λ

We need to solve:

$$\int_{\lambda=0}^{\infty}|\lambda|^{(\frac{N+1}{2}+g-1)}\exp(-\lambda\beta_0)d\lambda$$

Any ideas?

Integrating out λ

We need to solve:

$$\int_{\lambda=0}^{\infty}|\lambda|^{(\frac{N+1}{2}+g-1)}\exp(-\lambda\beta_0)d\lambda$$

Any ideas?

Setting $x = \lambda\beta_0$, and $d\lambda = \frac{dx}{\beta_0}$ we convert to a Gamma type integral $\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}\exp(-x)dx$ and get:

$$p(\mathcal{D},\mu|g,h,\gamma) = (2\pi)^{-\frac{N+1}{2}}\frac{h^g}{\Gamma(g)}|\gamma|^{\frac{1}{2}}\Gamma\left(\frac{N+1}{2}+g\right)$$

$$\times\left(h+0.5\Big(\Big(\sum_n x_n^2 - \frac{(\sum_n x_n)^2}{\gamma+N}\Big)\Big) + 0.5(\gamma+N)\Big(\mu - \frac{\sum_n x_n}{\gamma+N}\Big)^2\right)^{-\left(\frac{N+1}{2}+g\right)}$$

Further Analysis of $p(\mathcal{D},\mu|g,h,\gamma)$

$$p(\mathcal{D},\mu|g,h,\gamma) = (2\pi)^{-\frac{N+1}{2}}\frac{h^g}{\Gamma(g)}|\gamma|^{\frac{1}{2}}\Gamma\left(\frac{N+1}{2}+g\right)$$

$$\times\left(h+0.5\Big(\Big(\sum_n x_n^2 - \frac{(\sum_n x_n)^2}{\gamma+N}\Big)\Big)\right)^{-\left(\frac{N+1}{2}+g\right)}$$

$$\times\left(1+\frac{0.5(\gamma+N)\Big(\mu-\frac{\sum_n x_n}{\gamma+N}\Big)^2}{h+0.5\Big(\Big(\sum_n x_n^2 - \frac{(\sum_n x_n)^2}{\gamma+N}\Big)\Big)}\right)^{-\left(\frac{N+1}{2}+g\right)}$$

Compare with student-t distribution:

$$p(\mu|\theta,\kappa,\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}|\kappa|^{0.5}(\nu\pi)^{-0.5}\left(1+\frac{(\mu-\theta)^2\kappa}{\nu}\right)^{-\frac{\nu+1}{2}}$$

$->$ last factor proportional to student-t distrbution over μ

Analysis of $p(\mathcal{D}, \mu|g, h, \gamma)$ ctd.

Comparing coefficients:

$$\theta = \frac{\sum_n x_n}{N+\gamma} \;,\; \nu = N + 2g$$

$$\kappa = \frac{(N+2g)(N+\gamma)}{2h + \sum_n x_n^2 - \left(\sum_n x_n\right)^2/(N+\gamma)}$$

$$p(\mathcal{D}, \mu|g, h, \gamma) = (2\pi)^{-\frac{N+1}{2}} \frac{h^g}{\Gamma(g)} |\gamma|^{\frac{1}{2}} \Gamma\left(\frac{N+1}{2} + g\right)$$
$$\times \left(h + 0.5\Big(\Big(\sum_n x_n^2 - \frac{(\sum_n x_n)^2}{\gamma + N}\Big)\Big)\right)^{-\left(\frac{N+1}{2}+g\right)}$$
$$\times \frac{\Gamma\left(\frac{N+2g}{2}\right)}{\Gamma\left(\frac{N+2g+1}{2}\right)} \left|\frac{(N+2g)(N+\gamma)}{2h + \sum_n x_n^2 - \left(\sum_n x_n\right)^2/(N+\gamma)}\right|^{-0.5} ((N+2g)\pi)^{0.5}$$
$$\times \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} |\kappa|^{0.5} (\nu\pi)^{-0.5} \left(1 + \frac{(\mu-\theta)^2\kappa}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Any ideas how to get the marginal likelihood $p(\mathcal{D}|g, h, \gamma)$?

Marginal Likelihood and Posterior

$$p(\mathcal{D}|g, h, \gamma) = (2\pi)^{-\frac{N+1}{2}} \frac{h^g}{\Gamma(g)} |\gamma|^{\frac{1}{2}} \left(h + 0.5\Big(\Big(\sum_n x_n^2 - \frac{(\sum_n x_n)^2}{\gamma + N}\Big)\Big)\right)^{-\left(\frac{N+1}{2}+g\right)}$$
$$\times \Gamma\left(\frac{N+2g}{2}\right) \left|\frac{(N+2g)(N+\gamma)}{2h + \sum_n x_n^2 - \left(\sum_n x_n\right)^2/(N+\gamma)}\right|^{-0.5} ((N+2g)\pi)^{0.5}$$

$$p(\mu, \lambda|\mathcal{D}, g, h, \gamma) = \left(h + 0.5\Big(\Big(\sum_n x_n^2 - \frac{(\sum_n x_n)^2}{\gamma + N}\Big)\Big)\right)^{\left(\frac{N+1}{2}+g\right)}$$
$$\times \frac{1}{\Gamma\left(\frac{N+2g}{2}\right)\sqrt{((N+2g)\pi)}} \left|\frac{(N+2g)(N+\gamma)}{2h + \sum_n x_n^2 - \left(\sum_n x_n\right)^2/(N+\gamma)}\right|^{0.5}$$
$$\times |\lambda|^{(\frac{N+1}{2}+g-1)} \exp\Big(-\lambda\Big(h + 0.5\Big((\gamma+N)\mu^2 - 2\mu\sum_n x_n + \sum_n x_n^2\Big)\Big)\Big)$$

A MatLab Implementation

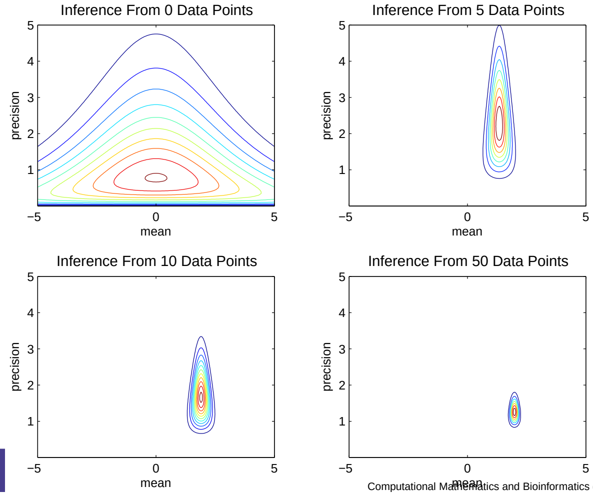
Note the implementation on the log scale!

```
function [mrgllh]=prcmn_gauss_mrglh(data, g, h, gam)
% function [mrgllh]=prcmn_gauss_mrglh(data, g, h, gam)
% calculates the log marginal likelihood of inferring a
% univariate Gaussian under a g-prior like scenario.
%
% (C) P. Sykacek 2007 <peter@sykacek.net>

data=data(:);
ndat=length(data);
sum_x_sqr=sum(data.ˆ2);
sqr_sum_x=sum(data).ˆ2;
mrgllh=-(ndat+1)/2 * log(2*pi) + g*log(h) - gammaln(g) + 0.5*log(gam);
mrgllh=mrgllh-((ndat+1)/2+g)*log(h+0.5*(sum_x_sqr-sqr_sum_x/(ndat+gam)));
mrgllh=mrgllh+gammaln(ndat/2+g)-0.5*(log(ndat+2*g)+log(ndat+gam)-...
    log(2*h+sum_x_sqr-sqr_sum_x/(ndat+gam)));
mrgllh=mrgllh+0.5*(log(ndat+2*g)+log(pi));
```

Posterior Dependency on Data Size

Prior settings: $g = 1.2$, $h = 0.9$ and $\gamma = 0.1$



Bayesian Model Selection I

All aspects of Bayesian inference:

Parameter inference:

$$p(\boldsymbol{\theta}|\mathcal{D}, I) = \frac{p(\mathcal{D}|\boldsymbol{\theta}, I)p(\boldsymbol{\theta}|I)}{p(\mathcal{D}|I)}$$

Note: $p(\mathcal{D}|I) = \int_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta}, I)p(\boldsymbol{\theta}|I)d\boldsymbol{\theta}$

Novel part: By including an indicator I, we made the model class explicit.

Bayesian Model Selection II

Reasoning about different model classes I:

$$P(I|\mathcal{D}) = \frac{P(I)p(\mathcal{D}|I)}{p(\mathcal{D})}$$

Note: $p(\mathcal{D}|I)$ is just the normalisation constant from parameter inference.

The above denominator is the normalisation constant $p(\mathcal{D}) = \sum_I P(I)p(\mathcal{D}|I)$.

Renormalising the maginal likelihood of model class I multiplied by its prior probability gives thus the posterior probability of model class I under the data $\mathcal{D}$.

Bayesian Model Selection III

If we have K models, we may chose $P(I) = \frac{1}{K}$ to reflect “ignorance”.

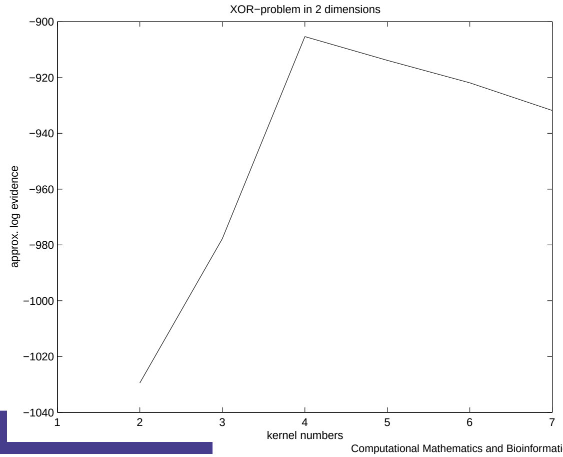
Model selection will choose model I with the largest posterior probability.

For equal priors, we select the model with the largest marginal likelihood. Unlike maximising the likelihood this quantity does not necessary lead to the most complex model winning!

If several model classes are equally probable, we should use $P(I|\mathcal{D})$ for model averaging.

Typical Behaviour

Plot of (approximate) log marginal likelihood in a binary regression problem (XOR-structure).



The Bayesian Version of a Paired T-Test

The classical paired t-test infers, whether some data are unlikely under the null hypothesis of being a zero mean Gaussian with unknown variance.

The Bayesian alternative is inferring the posterior probabilities, whether a zero mean Gaussian ($I = 0$), or a generic Gaussian ($I = 1$) are more probable under the dataset.

We choose uninformative priors $P(I = 0) = P(I = 1) = 0.5$ and need in addition the marginal likelihoods. As we know the marginal likelihood of the generic Gaussian already, we need only consider the zero mean Gaussian model.

jump 2 TOC

Zero Mean Gaussian Model

Likelihood:

$$p(\mathcal{D}|\lambda) = (2\pi)^{-\frac{n}{2}}|\lambda|^{\frac{N}{2}} \exp\left(-0.5\lambda \sum_n x_n^2\right)$$

and Gamma prior over λ:

$$p(\lambda|g,h) = \frac{h^g}{\Gamma(g)}\lambda^{g-1}\exp(-\lambda h)$$

Derive the marginal likelihood!

jump 2 TOC

Marginal Likelihoods and $P(I|\mathcal{D})$

Zero mean Gaussian:

$$p(\mathcal{D}|g,h,I=0) = \frac{h^g}{\Gamma(g)}(2\pi)^{-\frac{n}{2}}\left(h + 0.5\sum_n x_n^2\right)^{-(\frac{N}{2}+g)}\Gamma\left(\frac{N}{2}+g\right)$$

Full Gaussian (from previous calculations):

$$p(\mathcal{D}|g,h,\gamma,I=1) = (2\pi)^{-\frac{N+1}{2}}\frac{h^g}{\Gamma(g)}|\gamma|^{\frac{1}{2}}\left(h + 0.5\Big(\Big(\sum_n x_n^2 - \frac{(\sum_n x_n)^2}{\gamma+N}\Big)\Big)\right)^{-(\frac{N+1}{2}+g)}$$
$$\times\Gamma\left(\frac{N+2g}{2}\right)\left|\frac{(N+2g)(N+\gamma)}{2h+\sum_n x_n^2 - \left(\sum_n x_n\right)^2/(N+\gamma)}\right|^{-0.5}((N+2g)\pi)^{0.5}$$

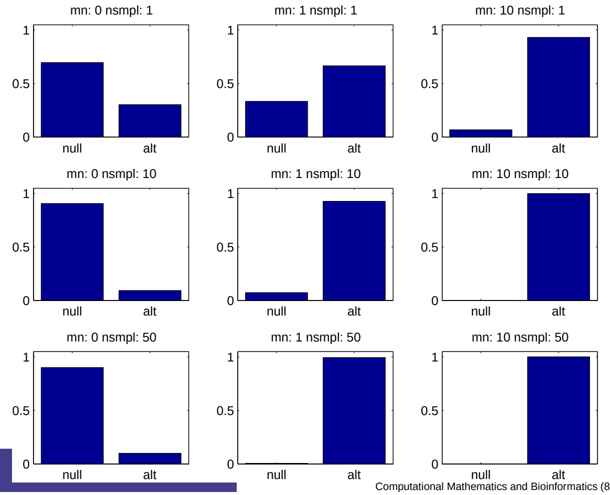
$P(I|\mathcal{D})$ from log marginal likelihoods, where $\log(p(\mathcal{D},I)) = \log(p(\mathcal{D}|I)) + \log(p(I))$:

$$P(I=i|\mathcal{D}) = \frac{1}{1+\sum_{j\neq i}\exp\left(\log(p(\mathcal{D},I=j)) - \log(p(\mathcal{D},I=j))\right)}$$

jump 2 TOC

Bayesian "T-Test" Applied

Priors: $g = 1.2$, $h = 0.9$, $\gamma = 0.1$ and $P(I) = 0.5$



jump 2 TOC

Summary

Model inference is based on Bayes theorem:

$$P(\theta|\mathcal{D}) = \frac{p(\theta)p(\mathcal{D}|\theta)}{p(\mathcal{D})}$$

and marginalisation:

$$P(I|\mathcal{D}) = \frac{\int_\theta p(\mathcal{D},\theta|I)d\theta P(I)}{\sum_I \int_\theta p(\mathcal{D},\theta|I)p(I)d\theta}$$

Inference results are either decisions after maximising expected utilities or posteriors summarising all uncertainty.

An important advantage of Bayesian statistics is to provide a consistent framework for all inference tasks.

Outlook

This lecture captured only very simple models that gave rise to analytically tractable calculations.

For models which include nonlinearities the integrals can not be solved analytically and explicit (exact) solutions do not exist.

If you are interested in advanced Bayesian methods that allow solving more complex problems you are warmly invited to attend 793.402 “Bayesian Data Analysis in the Life Scienes”. It will cover advanced aspects and include practical analysis sessions (3*6 hrs theory and 3 days blocked in the PC lab).