Introduction to Bioinformatical Data Analysis with Machine Learning

Peter Sykacek[1]

Vienna Science Chair of Bioinformatics
Department of Biotechnology
BOKU University
peter.sykacek@boku.ac.at

Overview

- Fundamental Problems of Analysing Data
- Concepts in Data Analysis
- Supervised Learning
- Unsupervised Learning
- Model Fitting, Diagnosis and Evaluation
- William of Occam and Karl R. Popper
- You Want More?

The Nature of Data

- Discrete valued observations (e.g. class labels).
- Continuous valued observations (e.g. measurements).

Measurement processes involve errors which arise from noise (fluctuations) that are or can not be captured:

- Measurement noise.
- Wrong classifications (e.g. disease state).
- Simplified Models.

Individual data points do thus not reflect ground truth. Data analysis uses replicates to remove the noise and model the remaining aspects as good as possible.

Refresher: Scalar Product

Our data analysis task: We have N samples of K-dimensional "input" measurements collected in N row vectors $\boldsymbol{x}_n^T = [\boldsymbol{x}_n[1], ..., \boldsymbol{x}_n[K]]$ and one dimensional dependent variables y_n, which we intend representing by a function $f(\boldsymbol{x}_n, \boldsymbol{\theta})$ parameterised by $\boldsymbol{\theta}$. This task is called *regression analysis* (discussed later in more detail).
For simplicity we also assume that after having measured a $\boldsymbol{x}_n$, our best guesses for y_n (denoted $\hat{y}_n$) is obtained as linear combination of the $\boldsymbol{x}_n[k]$. This allows writing:

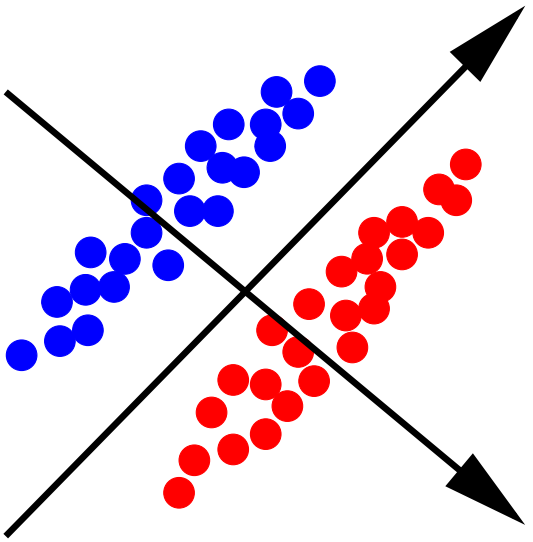
$$\hat{y}_n = \sum_k \boldsymbol{x}_n[k]\boldsymbol{\theta}[k], \text{ or } \hat{y}_n = \boldsymbol{x}_n^T\boldsymbol{\theta} \text{ and equivalently } \hat{y}_n = \boldsymbol{\theta}^T\boldsymbol{x}_n$$

The latter is called (vector) dot product or scalar product.

Why Understand Data Analysis?

Result = Data + Model!

Linear discriminant (LDA) and principle component analysis (PCA) give different projections of the same data.



Both use linear projections!

$$t_{\mathrm{PCA}} = \theta_{\mathrm{PCA}}^T x$$

$$t_{\mathrm{LDA}} = \theta_{\mathrm{LDA}}^T x$$

Technical Sufficiency

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

Example: Prove 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 processes. 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?)

Biological Relevance

Requires matching 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 greedy search to select some optimal gene set for cancer prediction and implying that this points to functionally important genes?

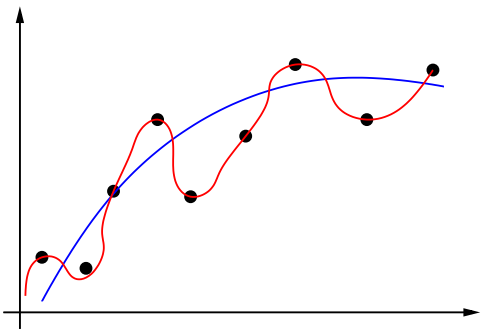
1) The powerful SVM does typically not outperform a linear classifier using a single gene.

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.

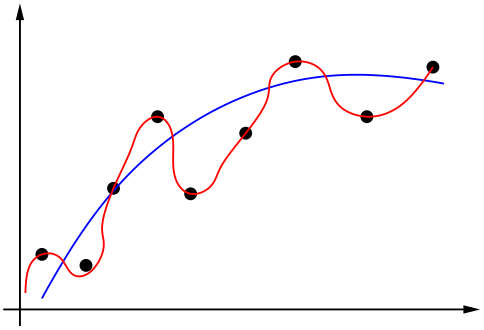
The otherwise useful approach (as a diagnostic tool) fails answering the biological question.

Fundamental Principle in Data Analysis

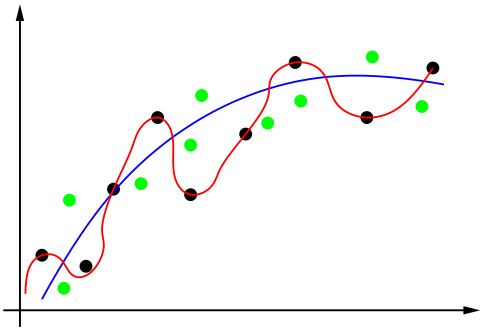


Which is the better model? Why is that the case?

Fundamental Principle in Data Analysis



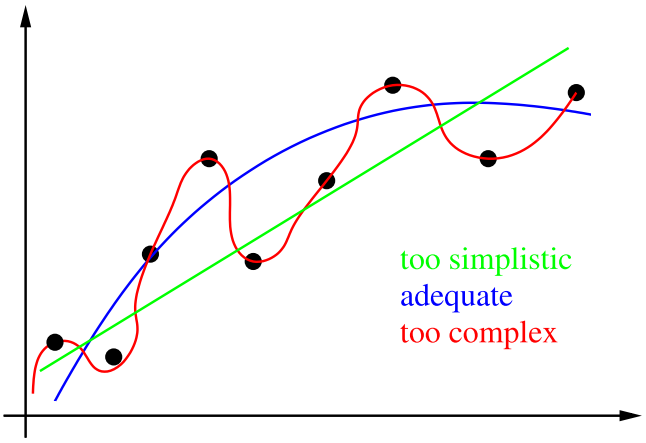
Which is the better model? Why is that the case?



Find (or abstract from) the underlying model that generated the data.

Adequate models

Capture underlying structure and avoid overfitting. “Fiddle parameters” affecting model complexity can have adverse effects.



Idea: overfitting is a result of tuning the model towards the training data. Over or under-complex models that do not capture the underlying data generating mechanism will perform worse on novel data obtained from the generating model than an appropriate model.

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 ?

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

PC 2010 8 TB Hard Drive (4*2 TB) $\approx$ 440 Euro

Stack height in 2010?

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.

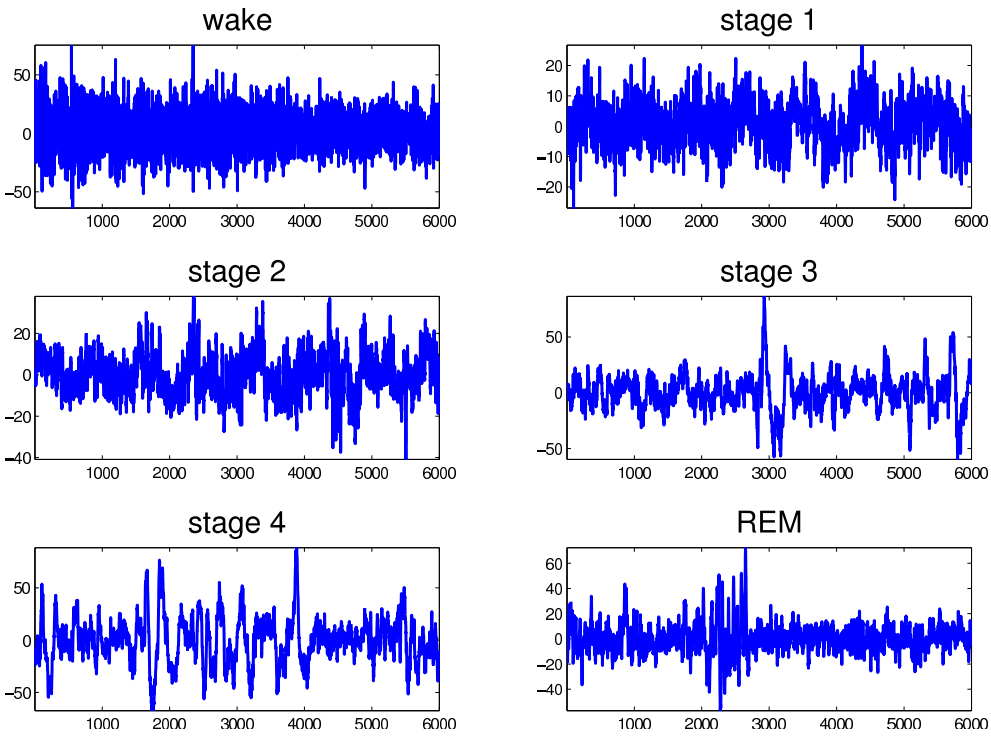
Medical monitoring 2:
An FMRI scanner, 1dm^3 volume, 10s temporal and 1mm^3 spatial resolution, 16 bit.
One scanner generates $10^6 * 360 * 2$ byte ≈ 720 MB per hour which fills 1 TB in about 58 days.

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.

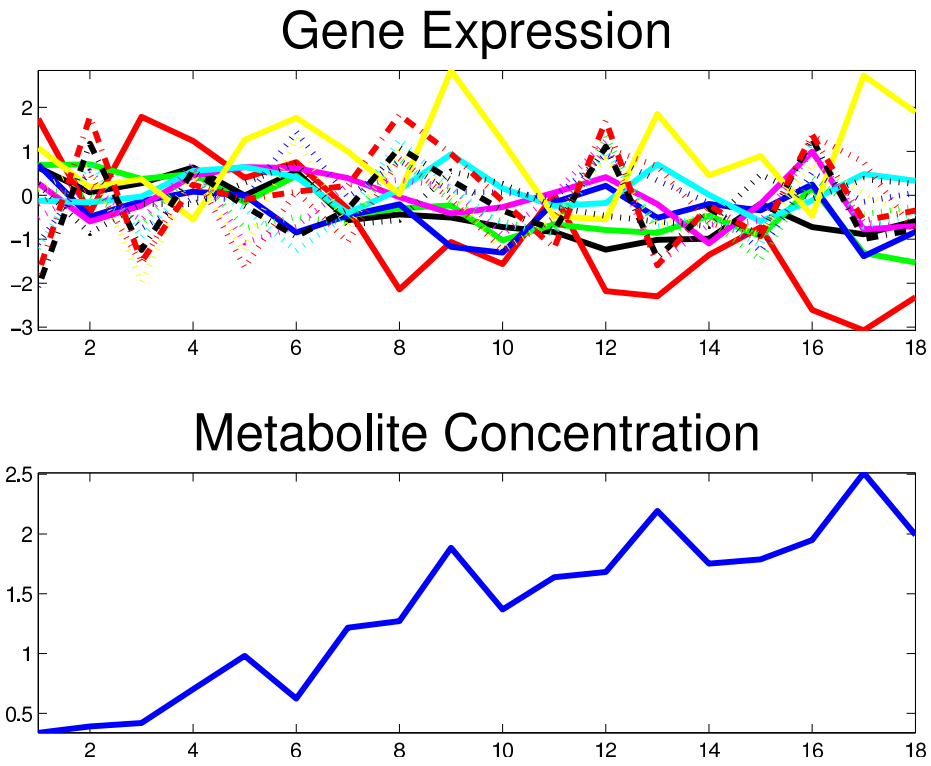
Such data can for two reasons not be analysed manually:

Amount and “Noise”

Example: Sleep EEG



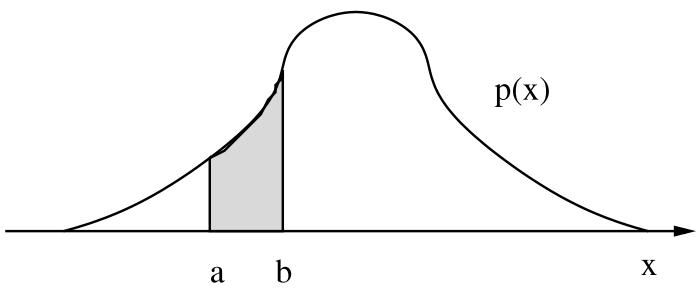
Example: Metabolomics



Random Variable and PDF

Data analysis is inherently connected with the concept of random variables. A random variable is a non deterministic quantity where repeated observations differ and are generated according to some overall property. Properties of random variables are for example captured by an associated probability density function (pdf), $p(x)$.

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



pdf's are also defined in multi dimensional spaces!

Analysis Strategies

All data analysis problems can be grouped into two categories:

1. Supervised Learning methods are used for regression problems.
2. Unsupervised Learning methods are used for exploratory data analysis.

Regression Problems

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

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

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

Regression Problems

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

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

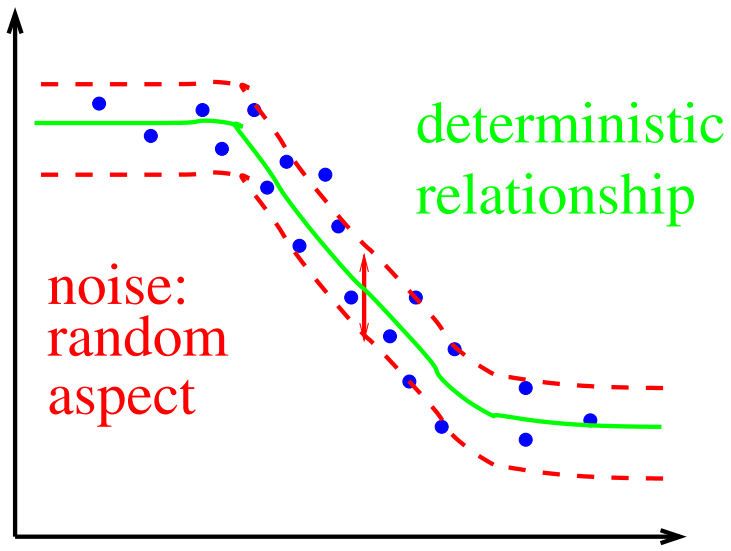
$$y = f(\boldsymbol{x}; \boldsymbol{\theta}) + \epsilon(\lambda)$$

Noise requires a deterministic and a random component.

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

Implication of Randomness

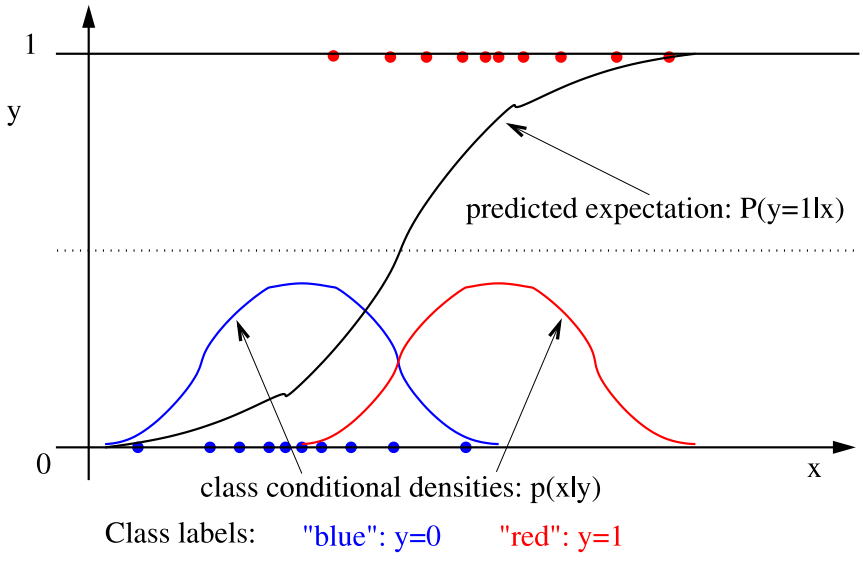
Noise implies that we do best by predicting the expected y values from x. The complete description of the relationship between y and x is provided by the conditional probability density function $p(y|x)$. The red error bars in the above figure capture only the second order statistics of the noise. This is only sufficient for Gaussian distributed noise.



Classification

Classification is a special case of regression, with predicted values (i.e. the y) being discrete.

Simplest case: Binary regression with $y = \{0, 1\}$. Predicting expectations is here equivalent to predicting class probabilities $P(y|x)$.



Exploratory Data Analysis

Exploratory data analysis searches for unknown structure in a data set of size N of the type $\mathcal{X} = \{x_1, x_2, ..., x_N\}$, with the x_n drawn from an unknown pdf $p(x)$. The learning task is modelling x as a function of an unobserved (latent) variable t, which provides a summary of the data.

Typical models for exploratory data analysis:

Mixture density models:

$p(x) = \sum_k P(t = k)p(x|t = k)$, and $t \in \{1, .., K\}$.

Continuous latent variable models:

$p(x) = \int_t p(t)p(x|t)dt$, $x \in \Re^k$, $t \in \Re^d$ and $k > d$.

Mixture Density Models

Gaussian mixture model: $p(x|t=k) = \mathcal{N}(x; \mu_k, \lambda_k)$, i.e. a (possibly multivariate) Gaussian density function.
K-means clustering: can be regarded as a mixture density model with $P(t=k) = 1/K$ and $p(x|t=k)$ being K uniform densities with domains emerging from the Voronoi tessellation defined by the K cluster centers.
Hidden Markov Model: assumes a one dimensional ordering (e.g. time) among the latent variables t_n. We have thus a more complicated prior: $P(t_n|t_{n-1})$.

These models infer as summary information which mixture component generated the data point x_n $->$ Clustering.

Continuous latent variable models

Common aspect: $x = [m+]Wt[+\epsilon]$, $W : [d \times k]$ dimensional coefficients matrix, m, ϵ : optional mean and noise term.
PCA (principle component analysis): $t \sim \mathcal{N}(t; 0, \Lambda)$, $\Lambda : [d \times d]$ diagonal cov. matrix, $\epsilon : 0$, m: data mean.
Factor analysis: $t \sim \mathcal{N}(t; 0, \Lambda)$, $\Lambda : [d \times d]$ general cov. matrix, $\epsilon_k \sim N(\epsilon_k; 0, \lambda_k)$ and m: data mean.
ICA (independent component analysis): $t \sim \prod_d p(t_d|\theta_d)$ and $p(t_d|\theta_d)$: univariate density functions, at maximum one Gaussian, m, ϵ : implementation dependent.

These models provide as summary a lower dimensional representation of the data $->$ dimensionality reduction.

Analysis Tasks and Methods

Task	$->$	Method
predict continuous y from input data	$->$	Regression
predict discrete y from input data	$->$	Classification
find unknown groups in input data	$->$	Clustering (e.g. k-means, mixture models)
find low dimensional representation for input data	$->$	Dimensionality reduction (ICA, PCA)

Once we chose the appropriate analysis methodology, we can start model fitting. This requires tuning of parameters and, as the most challenging aspect in data analysis, selecting a suitable model class.

Assessing Model Parameters

An obvious idea for representing all $(y_n \boldsymbol{x}_n)$ pairs well is to select $\boldsymbol{\theta}$ such that we get for all n samples an $f(\boldsymbol{x}_n; \boldsymbol{\theta})$ which is close to the corresponding y_n.
What is an appropriate expression we can minimise to achieve this?

Assessing Model Parameters

An obvious idea for representing all $(y_n \boldsymbol{x}_n)$ pairs well is to select $\boldsymbol{\theta}$ such that we get for all n samples an $f(\boldsymbol{x}_n; \boldsymbol{\theta})$ which is close to the corresponding y_n.

What is an appropriate expression we can minimise to achieve this?

One possible choice is the sum of squared errors (SSE). Idea: subtract the deterministic part from y_n: $\epsilon_n = y_n - f(\boldsymbol{x}_n; \boldsymbol{\theta})$

$$\mathrm{SSE} = \sum_n \epsilon_n^2 = \sum_n (y_n - f(\boldsymbol{x}_n; \boldsymbol{\theta}))^2$$

Analysing the SSE I

After minimising the SSE w.r.t. $\boldsymbol{\theta}$, the average of all ϵ_n should be zero. Why?

Because otherwise we can get an even smaller SSE by subtracting this average from all $f(\boldsymbol{x}_n; \boldsymbol{\theta})$.

We may thus assume that the inexplicable errors, which are random variables, are distributed according to a zero mean Gaussian distribution: $p(\epsilon_n|\lambda) = 1/\sqrt{2\pi}\sqrt{\lambda}\exp(-0.5\lambda\epsilon_n^2)$.

Rules of probability calculus: P(A,B)=P(A)*P(B) if A and B independent.

Analysing the SSE II

For independent ϵ_n, their joint pdf is thus:

$$p(\epsilon_1, ..., \epsilon_N|\lambda) = \left(\frac{\lambda}{2\pi}\right)^{\frac{N}{2}} \prod_n \exp(-0.5\lambda\epsilon_n^2)$$

Taking the logarithm we get:

$$\log(p(\epsilon_1, ..., \epsilon_N|\lambda)) = \frac{N}{2}\log\left(\frac{\lambda}{2\pi}\right)\sum_n -0.5\lambda\epsilon_n^2$$

$$= \frac{N}{2}\log\left(\frac{\lambda}{2\pi}\right) - 0.5\lambda\sum_n (y_n - f(\boldsymbol{x}_n;\boldsymbol{\theta}))^2$$

Likelihood and SSE

Looking at $\frac{N}{2}\log\left(\frac{\lambda}{2\pi}\right) - 0.5\lambda\sum_n (y_n - f(\boldsymbol{x}_n;\boldsymbol{\theta}))^2$, we can see that maximising it w.r.t. $\boldsymbol{\theta}$ is the same as minimising the SSE w.r.t. $\boldsymbol{\theta}$.

Expression $\log(p(\epsilon_1, ..., \epsilon_N|\lambda))$ is called a log likelihood. Maximising the log likelihood for a regression model with Gaussian noise distribution is identical to minimising the sum of squares error. Minimising the SSE makes thus very specific assumptions about the noise.

The concept of likelihoods (the above $p(\epsilon_1, ..., \epsilon_N|\lambda)$) or of log likelihoods is generic. Such objective functions can be derived for many data analysis strategies.

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?

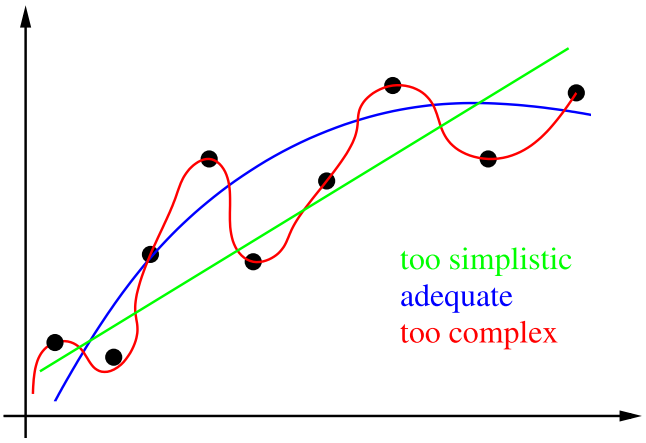
Think "phone book": Perfect memorising of all y_n, modelling error 0, $\lambda -> \infty$, $p(\mathcal{D}|\boldsymbol{\theta}, \lambda, \mathcal{X}) -> \infty$.

$->$ likelihood unsuitable objective for model inference!

Why is memorising useless?

Adequacy of models

Optimise model structure and avoid overfitting. Fiddle parameters affecting model complexity can have adverse effects.



Over or under-complex models that do not capture the underlying data generating mechanism will perform worse on novel data obtained from the generating model than an appropriate model. Model classes (just examples):

$y = kx + d + \epsilon$

$y = lx^2 + kx + d + \epsilon$

$y = \sum_{j=0}^{J}(x^j k_j) + \epsilon$

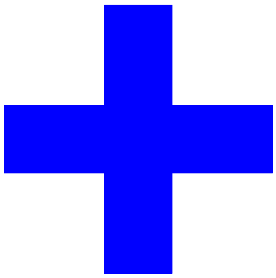
all cases: $\epsilon \sim \mathcal{N}(\epsilon; 0, \lambda)$.

How getting complexity right ?

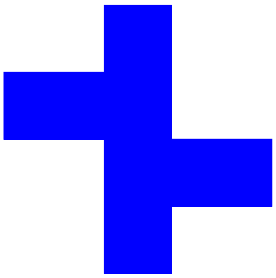
See the ideas by Karl R. Popper!

Human Intuition and Complexity

How many components?



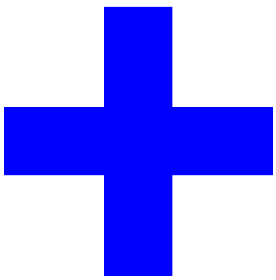
Object A



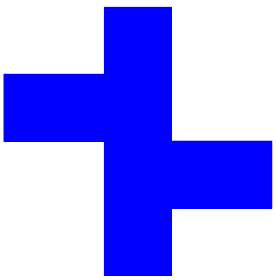
Object B

Human Intuition and Complexity

How many components?

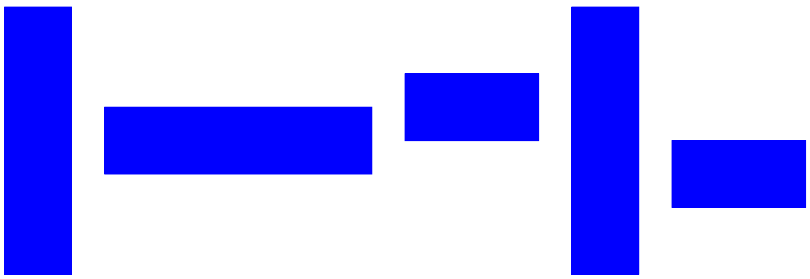


Object A



Object B

Most likely answer:



Object A

Object B

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`.

Occams Razor in ML & PRN

Model selection can be done by the following approaches:

- Adding a complexity penalty to the objective function. Many choices and active research field: AIC (Akaikes information criterion), BIC (Bayesian information criterion), MDL (minimum description length), etc.
- Using learning methods like Bayesian inference with Occam's razor built in.
- Using empirical approaches comparing model classes etc. by validation testing (computer simulation using independent data).

Instances of model selection

Problems in context of model selection:

- Variable selection: search for input subsets which improve predictive performance or identify important variables (e.g. differentially expressed genes).
- Change point detection: Separating data into groups which show similar statistical properties.
- Clustering: (see above)
- Determining optimal model orders (applies to most ML& PRN methods!)
- Determining suitable noise characteristics.
- ...

Validation: Diagnostic Measures

The mean square generalisation error (MSE_{test}) allows assessing regression models.

$$\mathrm{MSE}_{test} = \frac{1}{N} \sum_n \left(y_n^{test} - f(\boldsymbol{x}_n^{test}, \boldsymbol{\theta}^{train}) \right)^2$$

Classification aims at labeling novel samples correctly. This is tested by the generalisation accuracy acc_{test}.

$$\forall n \; \hat{y}_n^{test} = \mathrm{argmax}_k (P(y = k | \boldsymbol{x}_n^{test}, \boldsymbol{\theta}^{train}))$$

$$\mathrm{acc}_{test} = \frac{1}{N} \sum_n \delta(y_n^{test}, \hat{y}_n^{test})$$

We classify such that the most probable class wins and estimate the fraction of correctly classified test cases.

Validation: Estimation Procedures

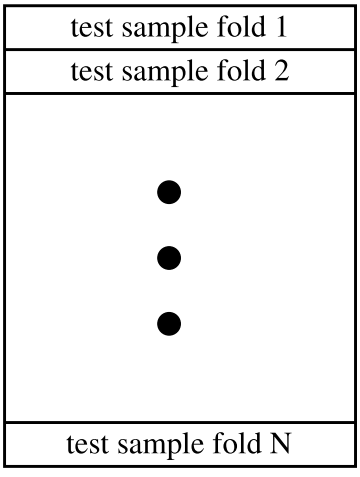
Trade-off: reliable inference requires large "training sets" (i.e. many samples for model fitting); unbiased diagnostics require large "test sets" (i.e. many novel samples for assessing the model).

Diagnostic quantities are only unbiased if we leave test samples untouched! Test samples must not be used for any modelling decisions.

$->$ Solution: reuse samples by iterating over model fitting and testing.

N-Fold Cross Testing

Sketch and MatLab like pseudo code




```
allres=[]; allpred=[];
for n=1:n_folds
    % split into training and test data
    [train, test]=foldsplit(orig_data, n_folds, n);
    % model inference
    [model]=trainfunc(train, fiddleparams);
    % store this folds true targets and predictions
    [res]=truetarg(test);
    [pred]=predtarg(test, model);
    allres=[allres; res];
    allpred=[allpred; pred];
end
```

Leave one out has as many folds as samples. An alternative by resampling with replacement is called Bootstrapping.

More on Assessing Classifiers

What is the implication of predicting the label which got largest posterior probability?

Consider an individual prediction and that we know the correct posterior $P(y|\boldsymbol{x})$: deciding for label $y = 1$ implies being correct with probability $P(y = 1|\boldsymbol{x})$ and being wrong with probability $1 - P(y = 1|\boldsymbol{x})$.

If we decide for the label $k = \text{argmax}_k(P(y = k|\boldsymbol{x}))$, we will thus minimise the overall number of missclassifications.

What if the missclassifications cost is class dependant?

Missclassification Cost

Classifying correctly induces zero cost. A false negative for class k induces cost α_k.
Predicting $y = t$ results in an expected cost
(deciding that the true y is probably t, this is the cost we expect to suffer):

$$C = \sum_{k \neq t} P(y = k|\boldsymbol{x})\alpha_k$$

Missclassification cost C is thus minimised by classifying $y = \text{argmax}_k(P(y = k|\boldsymbol{x})\alpha_k)$.

This structure is commonly found in life science applications. In cancer screening a false negative is obviously much worse than a false positive.

More on Data Analysis

Data Analysis is a very important topic in modern life sciences. I offer thus three elective courses on data analysis (all given in English, providing theoretical concepts and practical hands on experience in the computer lab).

- *Efficient Microarray Data Analysis with R and FSPMA (793.403)* 1.0 HRS WS 2010 1.5 ETCS, This is a two day blocked lecture which will be held in the Muthgasse Computer Lab in the 6th floor.
- *Neural Networks and Pattern Recognition in Bioinformatics (793.404)* 3.0 HRS as of WS 2010/11 a catalogued elective course with 4.5 ETCS - theoretical part and MatLab based practical in the computer lab.
- *Bayesian Data Analysis in the Life Sciences (793.402)* 3.0 HRS SS 2010 and again in SS 2011 (then with 4.5 ETCS). This lecture consists of a theoretical part and a 3 days blocked MatLab based practical in the computer lab.

Further details at
`http://www.sykacek.net/teaching.html`