

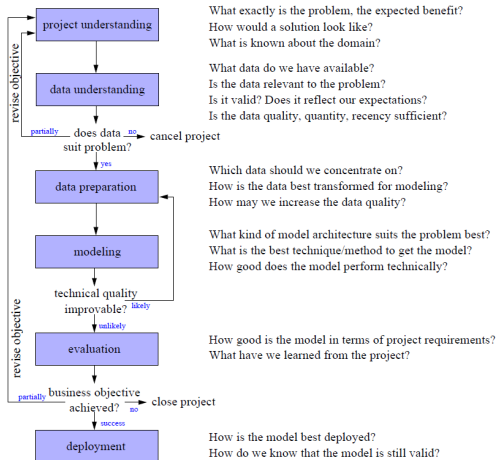
Data Analysis and Knowledge Discovery

Principles of Modelling

prof. Jukka Heikkonen

University of Turku
Department of Information Technology

Where are we now?



Outline

Introduction to machine learning

Algorithms for model fitting

Principles of modelling

- ▶ The general data mining task (classification, regression, clustering, associations, deviation analysis) for solving the problem should already be identified within project understanding.
- ▶ For each data mining task there are various models that are suitable to solve the given problem.
- ▶ Which one is the appropriate model?
- ▶ What are the underlying principles of all models?
- ▶ How can models be evaluated?

Machine Learning

Valiant, 1984

We say that a program for performing a task has been acquired by learning if it has been acquired by any means other than explicit programming.

Poggio & Smale, 2003

Learning from examples, refers to systems that are trained instead of programmed with a set of examples, that is, a set of input/output pairs.

Machine learning

- ▶ *machine learning*: field of artificial intelligence concerned with algorithms that can learn from data
- ▶ computational statistics, data mining, pattern recognition, statistical learning... closely related
- ▶ tools for extracting knowledge out of data in form of a model
- ▶ two main branches
 - ▶ unsupervised learning
 - ▶ supervised learning

Why Machine Learning?

Machine learning can provide new types of capabilities for computers:

- ▶ Data mining: extracting new information from medical records, maintenance records, etc.
- ▶ Self-customizing programs: Web browser that learns what you like and seeks it out
- ▶ Applications one can not program by hand: For example, speech recognition, autonomous driving

Why Machine Learning?

- ▶ Many old real-world applications of artificial intelligence were expert systems.
- ▶ Essentially a set of if-then rules to emulate a human expert.
- ▶ For example, if medical test A is positive and test B is negative and if patient is chronically thirsty, then diagnosis = diabetes with confidence 0.85.
- ▶ Rules were extracted via interviews of human experts.

Why Machine Learning?

- ▶ Expertise extraction for expert systems is tedious but for machine learning it is automatic.
- ▶ In expert systems, the extracted rules might not incorporate intuition, which may mask true reasons for answer: For example in medicine, the reasons given for diagnosis x might not be the objectively correct ones, and the expert might be unconsciously picking up on other info.

Why Machine Learning?

Expert systems: Expertise might not be comprehensive, for example, physician might not have seen some types of cases.

Machine learning: Automatic, objective, and data-driven, although it is only as good as the available data.

The Essence of Machine Learning

1. A pattern exists.
2. We can not pin it down mathematically.
3. We have data on it.

Unsupervised learning

- ▶ methods for exploratory data analysis: discover interesting patterns in data
- ▶ dimensionality reduction, clustering, associations, deviation analysis...
- ▶ applicable in several phases of data analysis process
 - ▶ data understanding: visualize data, check assumptions
 - ▶ data preparation: preprocessing before analysis (e.g. use PCA for dimensionality reduction)
 - ▶ modeling: sometimes the main analysis also carried out using unsupervised methods (e.g. cluster customers to different groups)

Supervised learning

- ▶ data: (input, correct output) -pairs (features, label)
- ▶ features: real-valued or categorical
- ▶ label: real-valued (regression) or categorical (classification)
- ▶ labels available for **training data**, unknown for future data
- ▶ goal: model dependency between the features and the label
- ▶ model predicts labels for new instances

Supervised learning

In supervised learning we are given a set of input-output pairs

$$\{(x_1, y_1), \dots, (x_n, y_n)\}$$

that we call a training set.

- ▶ Classification: A learning problem with output values taken from a finite unordered set $C = \{c_1, \dots, c_k\}$. A special case is binary classification where $y_i \in \{-1, 1\}$.
- ▶ Regression: A learning problem whose output values are real $y_i \in \mathbb{R}$.

Components of learning

Metaphor: Credit approval

Applicant information

age	23 years
gender	male
annual salary	30,000 €
years in residence	1 year
years in job	1 year
current debt	15,000 €
...	...

Approve credit?

Components of learning

Formalization:

- ▶ Input: \mathbf{x} (customer application)
- ▶ Output: y (good/bad customer?)
- ▶ Target function: $f : \mathcal{X} \rightarrow \mathcal{Y}$ (ideal credit approval formula)
- ▶ Data: $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ (historical records)
↓
- ▶ Hypothesis: $g : \mathcal{X} \rightarrow \mathcal{Y}$

Hypothesis set

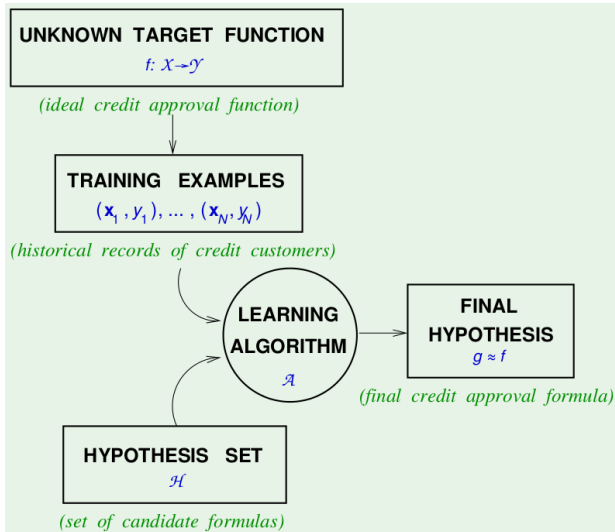
- ▶ The hypothesis set, \mathcal{H} , is the space of functions where we look for our solution.
- ▶ For many algorithms (such as optimization algorithms) it is the space the algorithm is allowed to search.
- ▶ Supervised learning uses the training data to learn a function

$$g : \mathcal{X} \rightarrow \mathcal{Y}$$

from \mathcal{H} , that can be applied to previously unseen data:

$$y_{pred} = g(x_{new})$$

Components of learning



Modelling steps

Building blocks for a machine learning algorithm:

Model class: general structure of the model (hypothesis set) e.g.:

- ▶ linear or quadratic function
- ▶ decision tree, neural network
- ▶ division of data to clusters
- ▶ ...

Score function: evaluates quality of different models

- ▶ does the model fit data?
 - ▶ squared error, cluster variance...
- ▶ how complex is the model?
 - ▶ e.g. minimum description length

Algorithm: find good model, as defined by score function

- ▶ mathematical optimization

Modelling steps

Final step: Validation (supervised learning)

- ▶ finding best fit to training data does not guarantee accurate predictions on new data
- ▶ danger of **overfitting**:
 - ▶ worst case: model just memorizes training data, does not generalize beyond it
 - ▶ e.g. a very complex function predicts whether you pay back your loan or not based on social security number
 - ▶ 100% accuracy on training data, fails to outperform random guessing on new instances
- ▶ alternatively **underfitting**:
 - ▶ model class not expressive enough, e.g. linear functions on non-linear problems

Model classes: Simple examples

- (a) Linear models. The simplest case of a linear model is a regression line

$$y = ax + b,$$

describing the idealized relationship between attributes x and y .

- ▶ The parameters a and b still need to be determined.
- ▶ They should be chosen in such a way that the regression line fits best to the given data.
- ▶ A perfect fit, i.e. the data points lie exactly on the regression line, cannot be expected.
- ▶ To fit the regression line to the data, a criterion or error measures is required, defining how well a line with given parameters a and b fits to the data.

Model classes: Simple examples

- (a) Linear models. More generally, the prediction of a linear model is based on several features

$$y = w_1x_1 + w_2x_2 + \dots + w_dx_d + b,$$

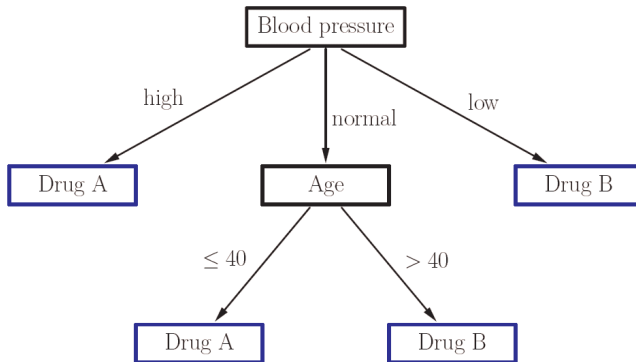
describing the idealized relationship between the feature vector $[x_1, \dots, x_d]$ and y .

Model classes: Simple examples

- ▶ Example models for nominal data: **association rules** like
 - ▶ *"If temperature=cold and precipitation=rain then action=read book"*.
 - ▶ *"If a customer buys product A, then he also buys product B with a probability of 30%"*.

Model classes: decision trees

Assignment of a drug to a patient:



Model classes: decision trees

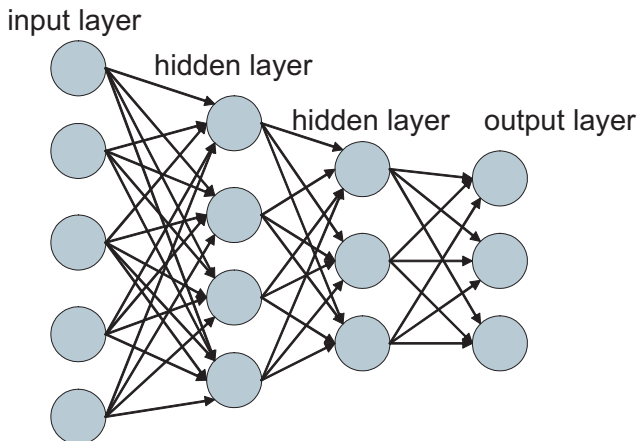
Recursive Descent:

- ▶ Start at the root node.
- ▶ If the current node is an **leaf node**:
 - Return the class assigned to the node.
- ▶ If the current node is an **inner node**:
 - Test the attribute associated with the node.
 - Follow the branch labeled with the outcome of the test.
 - Apply the algorithm recursively.

Intuitively: Follow the path corresponding to the case to be classified.

Model classes: neural networks

A feedforward neural network (advanced topic, covered in course TKO_3102)



Model classes

- ▶ **Parameterised models** are determined by specifying the values of a fixed number of parameters.
Example: A regression line $f(x) = a_1x + a_0$ is uniquely determined by the two parameters a_0 and a_1 .
- ▶ Examples for non-parameterised models:
 - ▶ Polynomials $f(x) = a_kx^k + \dots + a_1x + a_0$ where the degree k – the number of parameters – is not fixed.
 - ▶ Association rules.

How to find the best model?

- ▶ Project understanding will restrict the model class, but will usually not provide strict definition of the model class.
 - ▶ Example: If the data analysis task is identified as a regression problem, clustering models are not suitable.
 - ▶ But it is not specified which kind of regression function should be chosen.
- ▶ The model class must still be selected.
 - ▶ The choice of the wrong model class can spoil the whole data analysis process.
- ▶ It must be formalised what we mean by a “good” model.

Criteria for “good” models

Fitting criterion: How well does the model fit the data?

Model complexity: Usually, simpler models are preferred as they

- ▶ are easier to understand,
- ▶ easier to fit,
- ▶ avoid **overfitting**.

Overfitting refers to the problem that complex models might be flexible enough to fit the given data almost perfectly – learning them by heart – without representing the general structure in the data.

Criteria for “good” models

Interpretability is often desired.

- ▶ In some cases, a **black box model** suffices.
A black box model is function – a black box – that computes suitable outputs for inputs, but the structure of the function or its coefficients do not have any specific meaning.
- ▶ Interpretability is difficult to measure since it is context dependent: **Example**. A quadratic function (in the parameter time) might be perfectly interpretable in the context of a mechanical process where acceleration, velocity and distance are involved, whereas in a different setting the coefficients of a quadratic function might have no meaning.

Criteria for “good” models

Computational aspects: Finding the best, or at least a good, model w.r.t. the criteria data fitting, model complexity and interpretability is also a matter of computational complexity.

- In some cases, it might be very easy to find the best model w.r.t. to a given criterion.

Example. If the sum of the quadratic errors should be minimised by a regression line, a closed form solution can be provided, although the search space \mathbb{R}^2 is infinite.

Example. Although the search space for association rules is finite, at least when all involved attributes are categorical, the search space (all possible association rules that can be formulated) can be extremely large, leading to a

Practical advice

- ▶ figuring out the right features often much more important than the choice of learning method
- ▶ start with simple methods (linear regression, k-nearest neighbour, Naive Bayes...) rather than complex ones (support vector machines with kernels, deep neural networks, Bayesian networks...)
 - ▶ easier to understand
 - ▶ easy to find or program yourself good implementations
 - ▶ need less tuning, less risk of overfitting
 - ▶ in many practical problems often give just as good results as more advanced methods
- ▶ once you have done your best using a simple approach, you can check if you can do better with more advanced ones
- ▶ much more useful to know a few basic methods well, than 20 methods badly

Algorithms for model fitting

The objective function (scoring function) for models

- ▶ does not tell us how to find the best or a good model,
- ▶ it only provides a means for comparing models.

Optimisation algorithms to find the best or at least a good model are needed.

Algorithms for model fitting

Closed form solutions. In the best case, an explicit solution can be provided.

Example. Find a regression line $y = ax + b$ that minimises the mean square error for the data set $(x_1, y_1), \dots, (x_n, y_n)$.
Computing partial derivatives of the objective (error) function

$$E(a, b) = \frac{1}{n} \sum_{i=1}^n (ax_i + b - y_i)^2$$

w.r.t. the parameters a and b yields

$$\begin{aligned} \frac{\partial E}{\partial a} &= \frac{2}{n} \sum_{i=1}^n (ax_i + b - y_i)x_i = 0, \\ \frac{\partial E}{\partial b} &= \frac{2}{n} \sum_{i=1}^n (ax_i + b - y_i) = 0. \end{aligned}$$

Algorithms for model fitting

The solution of this system of equations is

$$a = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i) (\sum_{i=1}^n y_i)}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2},$$
$$b = \bar{y} - a\bar{x}$$

where $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$.

Algorithms for model fitting

Find a regression line $y = ax + b$ for the data $(x_1, y_1), \dots, (x_n, y_n)$
we can easily provide matrix equation: $Y = X\theta$, where

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, X = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix}, \theta = \begin{pmatrix} a \\ b \end{pmatrix}.$$

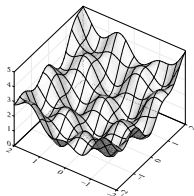
Least square solution for the equation is given by
 $\theta = (X^T X)^{-1} X^T Y$

Algorithms for model fitting

- ▶ Can be generalised to regression functions that are linear in the parameters to be optimised.
- ▶ Constraints can be incorporated e.g. with Lagrange functions.
- ▶ For arbitrary non-linear regression functions, no closed form solution exists.

Algorithms for model fitting

For differentiable score functions, a **gradient methods** can be applied (compare MDS).



Problems

- ▶ Will only find local optima.
- ▶ Parameters (step width) must be adjusted or computed in each iteration step.

Maximum Likelihood method

- ▶ Maximum Likelihood (ML) method is one of the most popular approaches for model fitting.
- ▶ ML is based on the assumption that random variable X follows distribution $p(x|\theta)$ where vector θ is defined by k parameters $\theta = (\theta_1, \dots, \theta_k)$.
- ▶ ML method determines optimal value for θ given the data x .
- ▶ The model $p(x|\theta_1, \dots, \theta_k)$ for X is the best one that makes the sample (x_1, \dots, x_n) most probable.

Maximum Likelihood method

Probability for an observation (x_1, \dots, x_n)

- ▶ $p(x_i|\theta_1, \dots, \theta_k) = P(X = x_i), i = 1, \dots, n.$
- ▶ Probability to get an observation (x_1, \dots, x_n) when assuming that samples x_i are independent:

$$L(\theta) = \prod_{i=1}^n p(x_i|\theta_1, \dots, \theta_k), \quad (1)$$



where $L(\theta)$ is so called likelihood-function consisting of $x_i, i = 1, \dots, n$ and parameters $\theta_j, j = 1, \dots, k$.

Maximum Likelihood method search those parameter values $\hat{\theta}$ that maximize the likelihood-function $L(\theta)$.

Maximum Likelihood method

- ▶ Maximum for $L(\theta)$ for many cases can be found by setting $\frac{\partial L}{\partial \theta_i} = 0$, $i = 1, \dots, k$.
- ▶ Often a logarithm over L is taken, e.g.

$$l(\theta) = \log L(\theta) = \sum_{i=1}^n \log p(x_i, \theta_1, \dots, \theta_k),$$

where $l(\theta)$ is called logarithmical likelihood-function (log likelihood).

Notice that if likelihood-function is not continuous the derivative zeros do not necessary directly give the maximum for L .

Maximum Likelihood method: example

Let us have $X \sim \text{bin}(n, p)$ with possible outcomes $\omega = \{A, B\}$. Having observations (x_1, \dots, x_n) the counts for A and B can be determined: n_A ja n_B ($n_A + n_B = n$). Estimate p by ML method.

- Likelihood-function $L(p)$ for $\text{bin}(n, p)$ with $X = n_A$ is

$$L(p) = \binom{n}{n_A} p^{n_A} (1 - p)^{n_B} ,$$

- Logarithm gives

$$l(p) = \log L(p) = \log \binom{n}{n_A} + n_A \log p + n_B \log(1 - p) .$$


- Derivate and set it to zero

$$\frac{dl}{dp} = \frac{n_A}{p} - \frac{n_B}{1 - p} = 0 .$$

- So $n_A(1 - p) = n_B p$ and hence ML estimate \hat{p} for p is $\hat{p} = \frac{n_A}{n}$.

Maximum Likelihood method: regression line

Find a regression line $y = ax + b$ for the data set $(x_1, y_1), \dots, (x_n, y_n)$ by ML method.

- ▶ The model can be described as $y = ax + b + \epsilon$, where ϵ is residual of the model.
- ▶ Let us assume that $\epsilon \sim N(0, \sigma^2)$. 
- ▶ Because $\epsilon = y - b - ax$ then $(Y - b - aX) \sim N(0, \sigma^2)$ and hence probability

$$p(y|a, b, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y - b - ax)^2\right\}.$$

- ▶ Having data set $(x_1, y_1), \dots, (x_n, y_n)$ the likelihood function is

$$L(y_1, \dots, y_n|a, b, \sigma^2) = \prod_{i=1}^n p(y_i|a, b, \sigma^2).$$

Maximum Likelihood method: regression line

- ▶ $L(y_1, \dots, y_n | a, b, \sigma^2) = \prod_{i=1}^n p(y_i | a, b, \sigma^2)$ means that

$$L(y_1, \dots, y_n | a, b, \sigma^2) =$$

$$\left(\frac{1}{2\pi\sigma^2} \right)^{\frac{n}{2}} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - b - ax_i)^2 \right\}.$$

- ▶ Now taking logarithm we obtain

$$l(a, b, \sigma^2) = -\frac{n}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - b - ax_i)^2.$$

Maximum Likelihood method: regression line

- ▶ We notice that

$$l(a, b, \sigma^2) = -\frac{n}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - b - ax_i)^2 .$$

is maximized when

$$\sum_{i=1}^n (y_i - b - ax_i)^2$$

is minimized.

- ▶ So ML estimates \hat{b} , \hat{a} corresponds exactly to the mean square error solution (also known as least square (LS) solution) that we have defined before.

Maximum Likelihood method: regression line

- ▶ ML estimate for σ^2 is obtained from derivate constraint

$$\frac{\partial l}{\partial \sigma^2} \Big|_{\sigma^2 = \hat{\sigma}^2} = -\frac{n}{2\hat{\sigma}^2} + \frac{1}{2(\hat{\sigma}^2)^2} \sum_{i=1}^n (y_i - \hat{b} - \hat{a}x_i)^2 = 0 ,$$

where it follows


$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{b} - \hat{a}x_i)^2 .$$

- ▶ Now we could also calculate maximum log likelihood

$$\begin{aligned} l(\hat{a}, \hat{b}, \hat{\sigma}^2) &= -\frac{n}{2} \log 2\pi \hat{\sigma}^2 - \frac{1}{2\hat{\sigma}^2} \sum_{i=1}^n (y_i - \hat{b} - \hat{a}x_i)^2 = \\ &= -\frac{n}{2} \log(2\pi) - \frac{n}{2} \log \hat{\sigma}^2 - \frac{n}{2} . \end{aligned}$$

Bayes: MAP estimator

Posterior distribution $p(\theta|x)$ when data x is given is according to Bayesian formula

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} . \quad (2)$$


- ▶ $p(\theta)$ is a prior probability for parameters θ
- ▶ $p(x|\theta)$ is likelihood function
- ▶ $p(x) = \int p(x|\theta)p(\theta)d\theta$ is normalization factor.

Maximum A posterior (MAP) method maximizes $p(\theta|x)$

$$p(\hat{\theta}|x) = \max_{\theta} p(\theta|x) . \quad (3)$$

MAP estimator

MAP estimate versus ML estimate

MAP (maximum a posterior) maximizes $p(\theta|x)$

ML (maximum likelihood) maximizes $p(x|\theta)$



When maximizing $p(\theta|x)$ we have the derivate constraint

$$\frac{\partial}{\partial \theta} p(\theta|x) = 0 ,$$

which is identical to

$$\frac{\partial}{\partial \theta} \log p(\theta|x) = 0 .$$

Using Bayesian formula we finally have to be solved:

$$\frac{\partial}{\partial \theta} \log p(x|\theta) + \frac{\partial}{\partial \theta} \log p(\theta) = 0 .$$

MAP estimator

Prior $p(\theta)$

- ▶ If $p(\theta)$ is a “flat” distribution compared to $p(x|\theta)$, then the mode of a *posteriori* distribution $p(\theta|x)$ is close to the mode of $p(\theta)$.
- ▶ If $p(\theta)$ is uniform distribution then $\frac{\partial}{\partial \theta} \log p(\theta) = 0$ and hence

$$\frac{\partial}{\partial \theta} \log p(\theta|x) = \frac{\partial}{\partial \theta} \log p(x|\theta) ,$$

which means that a *posteriori* distribution has is maximum when

$$\frac{\partial}{\partial \theta} \log p(x|\theta) = 0 .$$

Notice that is corresponds to ML estimate.

MAP estimator example

Let us have process measurements x_i , $i = 1, \dots, n$. The measures follows the model $x_i = as_i + \epsilon_i$, where s_i is known and the model error $\epsilon \sim N(0, \sigma_\epsilon)$. Use ML and MAP estimators to estimate a with an assumption $p(a) = N(0, \sigma_a^2)$, when both σ_ϵ and σ_a are known.

$$\epsilon_i = x_i - as_i \text{ and } \epsilon \sim N(0, \sigma_\epsilon^2) \Rightarrow$$

$$p(x_1, \dots, x_n | a) \equiv p(x | a) = \left(\frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} \right)^n \exp \left\{ -\frac{1}{2\sigma_\epsilon^2} \sum_{i=1}^n (x_i - as_i)^2 \right\}.$$

MAP estimator example

The condition

$$\frac{\partial}{\partial a} \log p(x|a) = \frac{1}{\sigma_\epsilon^2} \sum_{i=1}^n (x_i - as_i)s_i = 0 ,$$

gives ML estimate

$$a_{\text{ML}} = \frac{\sum_{i=1}^n x_i s_i}{\sum_{i=1}^n s_i^2} .$$

MAP estimator example

MAP estimate for a is obtained from

$$\frac{\partial}{\partial a} \log p(x|a) + \frac{\partial}{\partial a} \log p(a) = 0 ,$$

where it is assumed that

$$p(a) = \frac{1}{\sqrt{2\pi\sigma_a^2}} \exp \left\{ -\frac{a^2}{2\sigma_a^2} \right\} .$$

MAP estimator example

We obtain the form

$$\frac{1}{\sigma_{\epsilon}^2} \sum_{i=1}^n (x_i - as_i)s_i - \frac{a}{\sigma_a^2} = 0$$

giving MAP estimate

$$a_{\text{MAP}} = \frac{\frac{\sigma_a^2}{\sigma_{\epsilon}^2} \sum_{i=1}^n x_i s_i}{1 + \frac{\sigma_a^2}{\sigma_{\epsilon}^2} \sum_{i=1}^n s_i^2}.$$

Notice that if $\sigma_a \gg \sigma_{\epsilon}$ then $a_{\text{MAP}} \rightarrow a_{\text{ML}}$.

Algorithms for model fitting: other cases

- ▶ For discrete problems with a finite search space (like finding association rules), **combinatorial optimisation** strategies are needed.
- ▶ In principle, an **exhaustive search** of the finite domain \mathcal{M} is possible, however, in most cases it is not feasible, since \mathcal{M} is much too large.
- ▶ **Example.** Finding the best possible association rules with an underlying set of 1000 items (products).
Every combination of items, i.e. every nonempty subset is a possible candidate set from which several rules may be constructed.
The number of nonempty subsets alone contains $2^{1000} - 1 > 10^{300}$ elements.

Heuristic strategies are therefore needed.

Algorithms for model fitting: Heuristic strategies

Random search. Create random solutions and choose the best one among them.

Very inefficient

Greedy strategies. Formulate an algorithm that tries to improve the solution in each step.

- ▶ **Example.** Gradient method.

- ▶ **Example.** Hillclimbing.

Start with a random solution,
generate new solutions in the “neighbourhood”
of the solution.

If a new solution is better than the old one,
generate new solutions in its “neighbourhood”.

Can find a solution quickly, but get stuck in local optima.

Algorithms for model fitting: Heuristic strategies

Simulated annealing is a mixture between random search and a greedy strategy. Simulated annealing is a modified version of hillclimbing, sometimes replacing better solutions by worse ones with a (low) probability. This probability is decreased in each iteration step.

Evolutionary algorithms like **evolution strategies** or **genetic algorithms** combine random with greedy components, using a population of solutions in order to explore the search space in parallel and efficiently.

Algorithms for model fitting: Heuristic strategies

Alternating optimisation can be applied when the set of parameters can be split into disjoint subsets in such a way that for each subset an analytical solution for the optimum can be provided, given the parameters in the other subsets are fixed. Alternating optimization computes the analytical solution for the parameter subsets alternatingly and iterates this schema until convergence.