Probabilistic Data Mining

Lehel Csató

Faculty of Mathematics and Informatics
Babeş–Bolyai University, Cluj-Napoca,

Matematika és Informatika Tanszék
Babeş–Bolyai Tudományegyetem, Kolozsvár

March 2008

Table of Contents

Outline

Historical background / Motivation:

- Huge amount of data, that should automatically be processed,
- Mathematics provides general solutions, solutions are i.e. not for a given problem,
- Need for "science", that uses mathematics machinery for solving practical problems.

Definitions for Machine Learning

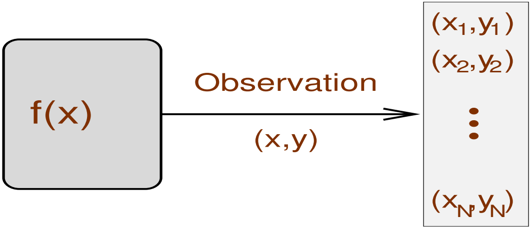
Machine learning

Collection of methods (from statistics, probability theory) to solve problems met in practice.

- noise filtering for
  - non-linear regression and/or
  - non-Gaussian noise
- Classification:
  - binary,
  - multiclass,
  - partially labelled
- Clustering,
- Inversion problems,
- density estimation, novelty detection.

Generally, we need to model the data,

Modelling Data

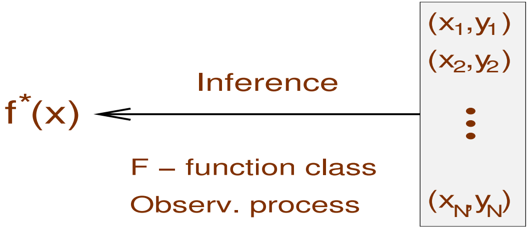


- Real world: there “is” a function $y = f(x)$
- Observation process: a corrupted datum is collected for a sample x_n:

$$\begin{aligned} t_n &= y_n + \epsilon && \text{additive noise} \\ t_n &= h(y_n, \epsilon) && h \text{ distortion function} \end{aligned}$$

- **Problem: find function $y = f(x)$**

Latent variable models



- **Data set** – collected.
- Assume a function class.
  - polynomial,
  - Fourier expansion,
  - Wavelet;
- Observation process – encodes the noise;
- Find the optimal function from the class.

Latent variable models II

- We have the **data set** $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$.
- Consider a function class:

$$(1) \quad \mathcal{F} = \left\{\mathbf{w}^T\mathbf{x} + b | \mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R}\right\}$$

$$(2) \quad \mathcal{F} = \left\{a_0 + \sum_{k=1}^{K} a_k \sin(2\pi k x) + \sum_{k=1}^{K} b_k \cos(2\pi k x) \middle| \mathbf{a}, \mathbf{b} \in \mathbb{R}^K,\ a_0 \in \mathbb{R}\right\}$$

- Assume an observation process:

$$y_n = f(\mathbf{x}_n) + \epsilon \quad \text{with } \epsilon \sim N(0, \sigma^2).$$

Latent variable models III

1. The **data set:** $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$.

2. Assume a function class:

$$\mathcal{F} = \{ f(\mathbf{x}, \boldsymbol{\theta}) | \boldsymbol{\theta} \in \mathbb{R}^p \}$$

   $\mathcal{F}$ – polynomial, etc.

3. Assume an observation process. Define a loss function:

$$L\left(y_n, f(\mathbf{x}_n, \boldsymbol{\theta})\right)$$

   For the Gaussian noise:

$$L(y_n, f(\mathbf{x}_n, \boldsymbol{\theta})) = (y_n - f(\mathbf{x}_n, \boldsymbol{\theta}))^2.$$

Outline

Parameter estimation

Estimating parameters:

Finding the **optimal value to $\boldsymbol{\theta}$**:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\Omega} L(\mathcal{D}, \boldsymbol{\theta})$$

where

- Ω is the domain of the parameters.
- $L(\mathcal{D}, \boldsymbol{\theta})$ is a "loss function" for the data set. Example:

$$L(\mathcal{D}, \boldsymbol{\theta}) = \sum_{n=1}^{N} L(y_n, f(\boldsymbol{x}_n, \boldsymbol{\theta}))$$

Maximum Likelihood Estimation

$L(\mathcal{D}, \boldsymbol{\theta})$ – (log)likelihood function.

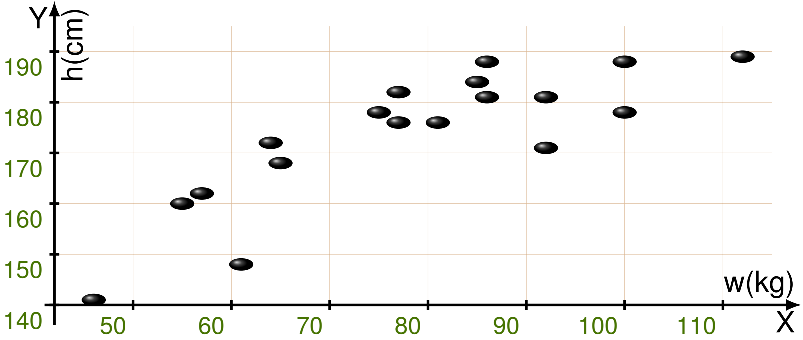
Maximum likelihood estimation of the model:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\Omega} L(\mathcal{D}, \boldsymbol{\theta})$$

Example – quadratic regression:

$$L(\mathcal{D}, \boldsymbol{\theta}) = \sum_{n=1}^{N} (y_n - f(\mathbf{x}_n, \boldsymbol{\theta}))^2 \quad \text{– factorisation}$$

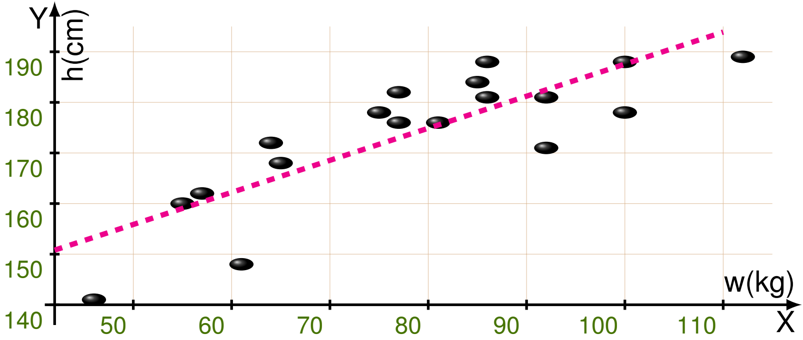
Drawback: can produce perfect fit to the data – **over-fitting**.



- We want to fit a model to the data.
- Use linear model: $h = \theta_0 + \theta_1 w$.
- Use log-linear model: $h = \theta_0 + \theta_1 \log(w)$.
- Use higher order polynomials, *e.g.* :

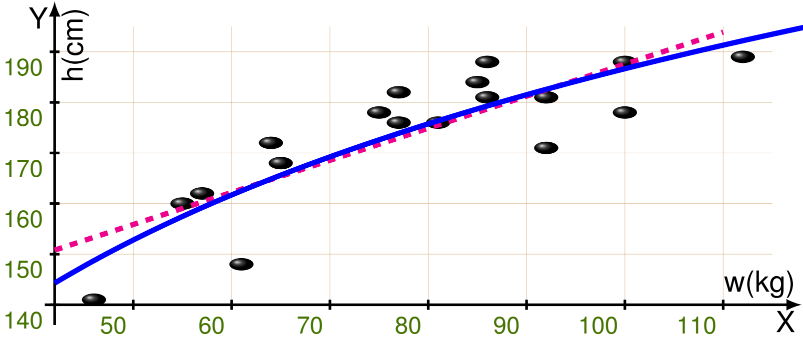
$$h = \theta_0 + \theta_1 w + \theta_2 w^2 + \theta_3 w^3 + \ldots .$$

Example of an ML estimate — Graphic



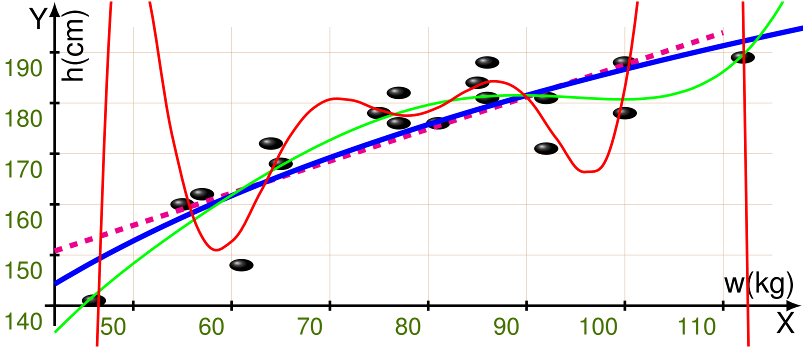
- We want to fit a model to the data.
- Use linear model: $h = \theta_0 + \theta_1 w$.
- Use log-linear model: $h = \theta_0 + \theta_1 \log(w)$.
- Use higher order polynomials, *e.g.* :

$$h = \theta_0 + \theta_1 w + \theta_2 w^2 + \theta_3 w^3 + \ldots.$$



- We want to fit a model to the data.
- Use linear model: $h = \theta_0 + \theta_1 w$.
- Use log-linear model: $h = \theta_0 + \theta_1 \log(w)$.
- Use higher order polynomials, *e.g.* :

$$h = \theta_0 + \theta_1 w + \theta_2 w^2 + \theta_3 w^3 + \ldots .$$



- We want to fit a model to the data.
- Use linear model: $h = \theta_0 + \theta_1 w$.
- Use log-linear model: $h = \theta_0 + \theta_1 \log(w)$.
- Use higher order polynomials, *e.g.* :

$$h = \theta_0 + \theta_1 w + \theta_2 w^2 + \theta_3 w^3 + \ldots .$$

M.L. for linear models I

Assume:

linear model for the $\boldsymbol{x} \rightarrow y$ relation

$$f(\boldsymbol{x}_n|\boldsymbol{\theta}) = \sum_{\ell=1}^{d} \theta_\ell x_\ell$$

with $\boldsymbol{x} = [1, x, x^2, \log(x), \ldots]^T$

quadratic loss for $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, h_N)\}$

$$E_2(\mathcal{D}|f) = \sum_{n=1}^{N} (y_n - f(\boldsymbol{x}_n|\boldsymbol{\theta}))^2$$

Probabilistic Data Mining

Lehel Csató

Modelling Data
Machine Learning
Latent variable models

Estimation
Maximum Likelihood
Maximum a-posteriori
Bayesian Estimation

Unsupervised
General concepts
Principal Components
Independent Components
Mixture Models

References

Minimisation:

$$\sum_{n=1}^{N}(y_n - f(\boldsymbol{x}_n|\boldsymbol{\theta}))^2 = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})$$
$$= \boldsymbol{\theta}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\theta} - 2\boldsymbol{\theta}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y}$$

Solution:

$$0 = 2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\theta} - 2\boldsymbol{X}^T\boldsymbol{y}$$
$$\boldsymbol{\theta} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

where $\boldsymbol{y} = [y_1, \ldots, y_N]^T$ and $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^T$ are the transformed data.

Generalised linear models:

- Use a *set of functions* $\Phi = [\phi_1(.), \ldots, \phi_M(.)]$.
- Project the inputs into the space spanned by $\mathrm{Im}(\Phi)$.
- Have a parameter vector of length M: $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_M]^T$.
- The model is $\left\{ \sum_m \theta_m \phi_m(\mathbf{x}) \,|\, \theta_m \in \mathbb{R} \right\}$.
- The optimal parameter vector is:

$$\theta^* = \left(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^T \mathbf{y}$$

- There are many candidate model **families**:
  - the degree of polynomials specifies a model family;
  - the rank of a Fourier expansion;
  - the mixture of $\{\log, \sin, \cos, \ldots\}$ also a *family*;
- Selecting the "best family" is a difficult *modelling* problem.
- In maximum likelihood there is no controll on how good a family is when processing a *given* data-set.

Smaller number of parameters than $\sqrt{\#\text{data}}$.

Maximum a–posteriori I

- Generalised linear model powerful – it can be extremely complex;
  - With no complexity control, overfitting problem.
- **Aim:** to include knowledge in the inference process.
- Our beliefs are reflected by the choice of the candidate functions.

Maximum a–posteriori I

- Generalised linear model powerful – it can be extremely complex;
  - With no complexity control, overfitting problem.
- **Aim:** to include knowledge in the inference process.
- Our beliefs are reflected by the choice of the candidate functions.

Goal:

- Prior knowledge specification using probabilities;
- Using probability theory for consistent estimation;
- Encode the observation noise in the model;

Probabilistic data description:

- **How likely** is that $\boldsymbol{\theta}$ generated the data:

$$
\begin{aligned}
y = f(\mathbf{x}) \quad &\Leftrightarrow \quad y - f(\mathbf{x}) \sim \delta_0 \\
y = f(\mathbf{x}) + \epsilon \quad &\Leftrightarrow \quad y - f(\mathbf{x}) \sim N_\epsilon
\end{aligned}
$$

- **Gaussian noise:** $y - f(\mathbf{x}) \sim N(0, \sigma^2)$

$$
P(y|f(\mathbf{x})) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{(y - f(\mathbf{x}))^2}{2\sigma^2}\right]
$$

William of Ockham (1285–1349) principle

Entities should not be multiplied beyond necessity.

Also known as (wiki...): "Principle of simplicity" – KISS, "When you hear hoofbeats, think horses, not zebras".

Simple models $\approx$ small number of parameters.

L_0 norm

L_2 norm $\Leftarrow$

Probabilistic representation:

$$p_0(\boldsymbol{\theta}) \propto \exp\left[-\frac{\|\boldsymbol{\theta}\|_2^2}{2\sigma_0^2}\right]$$

M.A.P. – probabilities assigned to

- $\mathcal{D}$ – via the log-likelihood function:

$$P(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}, \mathcal{F}) \propto \exp\left[-L(y_n, f(\boldsymbol{x}_n, \boldsymbol{\theta}))\right]$$

- $\boldsymbol{\theta}$ – prior probabilities:

$$p_0(\boldsymbol{\theta}) \propto \exp\left[-\frac{\|\boldsymbol{\theta}\|^2}{2\sigma_0^2}\right]$$

- A–posteriori probability:

$$p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{F}) = \frac{P(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})}{p(\mathcal{D}|\mathcal{F})}$$

$p(\mathcal{D}|\mathcal{F})$ – probability of the data for a given family.

M.A.P. – probabilities assigned to

- $\mathcal{D}$ – via the log-likelihood function:

$$P(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}, \mathcal{F}) \propto \exp\left[-L(y_n, f(\boldsymbol{x}_n, \boldsymbol{\theta}))\right]$$

- $\boldsymbol{\theta}$ – prior probabilities:

$$p_0(\boldsymbol{\theta}) \propto \exp\left[-\frac{\|\boldsymbol{\theta}\|^2}{2\sigma_0^2}\right]$$

- A–posteriori probability:

$$p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{F}) = \frac{P(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})}{p(\mathcal{D}|\mathcal{F})}$$

$p(\mathcal{D}|\mathcal{F})$ – probability of the data for a given family.

M.A.P. – probabilities assigned to

- $\mathcal{D}$ – via the log-likelihood function:

$$P(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}, \mathcal{F}) \propto \exp\left[-L(y_n, f(\boldsymbol{x}_n, \boldsymbol{\theta}))\right]$$

- $\boldsymbol{\theta}$ – prior probabilities:

$$p_0(\boldsymbol{\theta}) \propto \exp\left[-\frac{\|\boldsymbol{\theta}\|^2}{2\sigma_0^2}\right]$$

- A–posteriori probability:

$$p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{F}) = \frac{P(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})}{p(\mathcal{D}|\mathcal{F})}$$

$p(\mathcal{D}|\mathcal{F})$ – probability of the data for a given family.

M.A.P. estimation – finds $\boldsymbol{\theta}$ with largest probability:

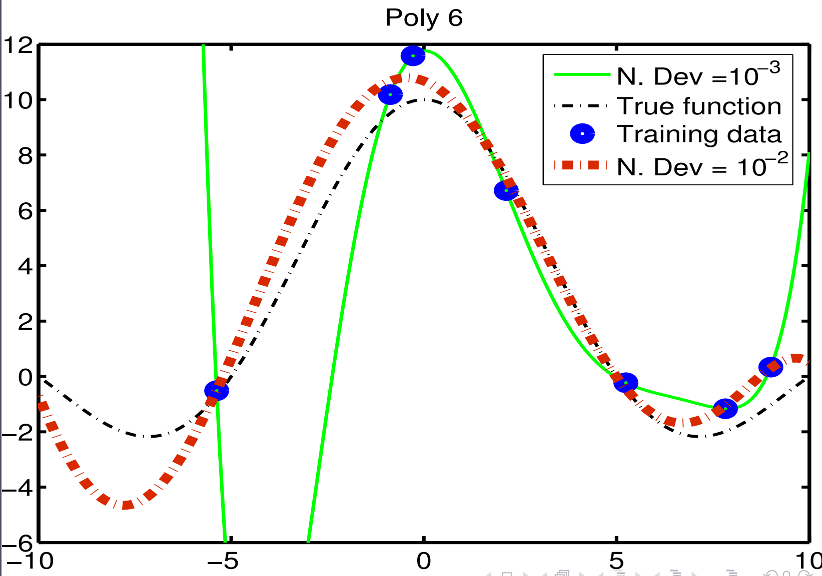
$$\boldsymbol{\theta}^*_{MAP} = \arg\max_{\boldsymbol{\theta}\in\Omega} p(\boldsymbol{\theta}|\mathcal{D},\mathcal{F})$$

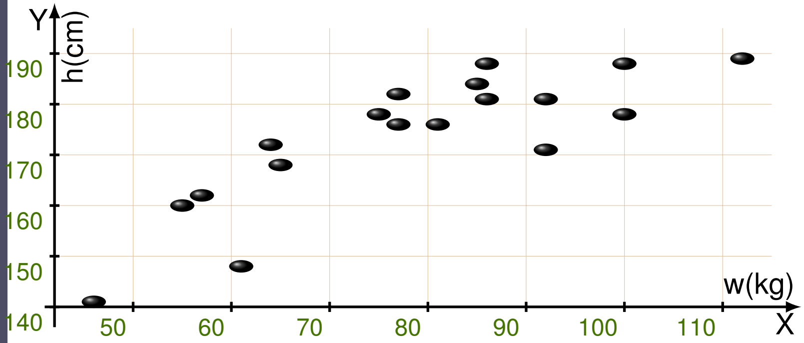
Example: with $L(y_n, f(\mathbf{x}_n, \boldsymbol{\theta}))$ and Gaussian prior:

$$\boldsymbol{\theta}^*_{MAP} = \operatorname*{argmax}_{\boldsymbol{\theta}\in\Omega} K - \frac{1}{2}\sum_n L(y_n, f(\mathbf{x}_n, \boldsymbol{\theta})) - \frac{\|\boldsymbol{\theta}\|^2}{2\sigma_0^2}$$

$\sigma_0^2 = \infty \quad \Longrightarrow \quad$ maximum likelihood.

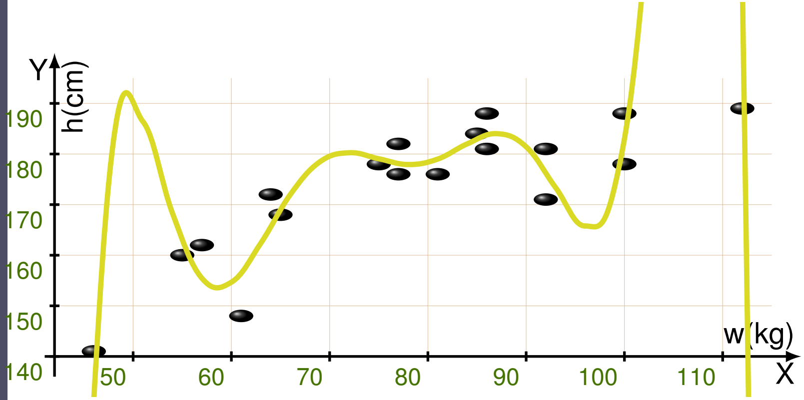
after a change of sign and $\max \rightarrow \min$


Poly 6
N. Dev =10^{-3}
True function
Training data
N. Dev = 10^{-2}
12
10
8
6
4
2
0
−2
−4
−6
−10
−5
0
5
10



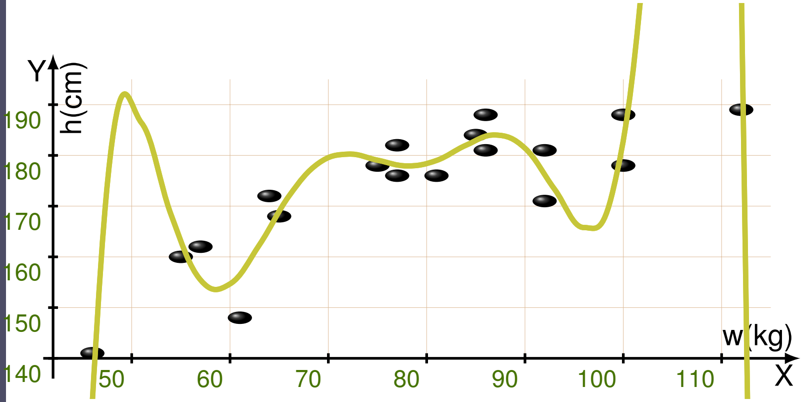
Aim: test different levels of flexibility. $\Rightarrow$ $p = 10$

Prior width:



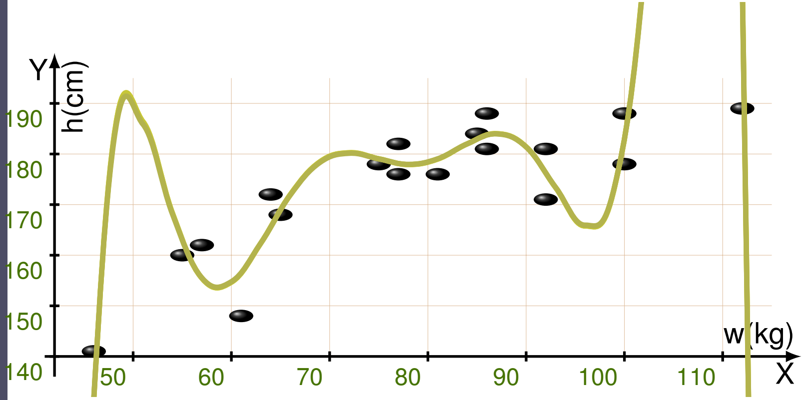
Aim: test different levels of flexibility. $\Rightarrow$ $p = 10$

Prior width: $\sigma_0^2 = 10^6$



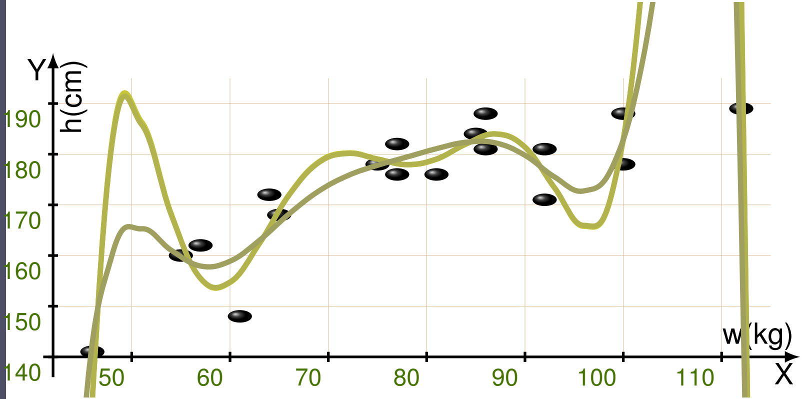
Aim: test different levels of flexibility. $\Rightarrow$ $p = 10$

Prior width: $\sigma_0^2 = 10^6$ $\sigma_0^2 = 10^5$



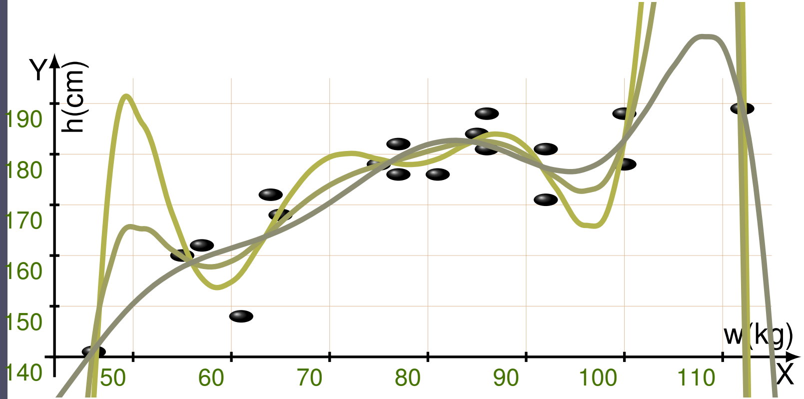
Aim: test different levels of flexibility. $\Rightarrow \quad p = 10$

Prior width: $\sigma_0^2 = 10^6 \quad \sigma_0^2 = 10^5 \quad \sigma_0^2 = 10^4$



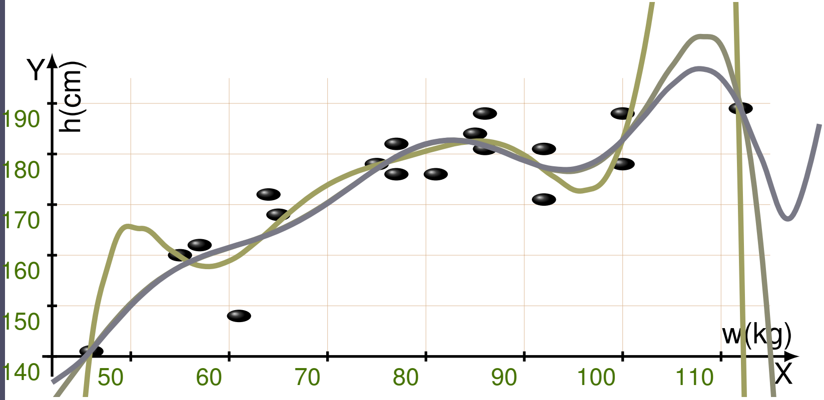
Aim: test different levels of flexibility. $\Rightarrow \quad p = 10$

Prior width: $\sigma_0^2 = 10^6 \quad \sigma_0^2 = 10^5 \quad \sigma_0^2 = 10^4 \quad \sigma_0^2 = 10^3$



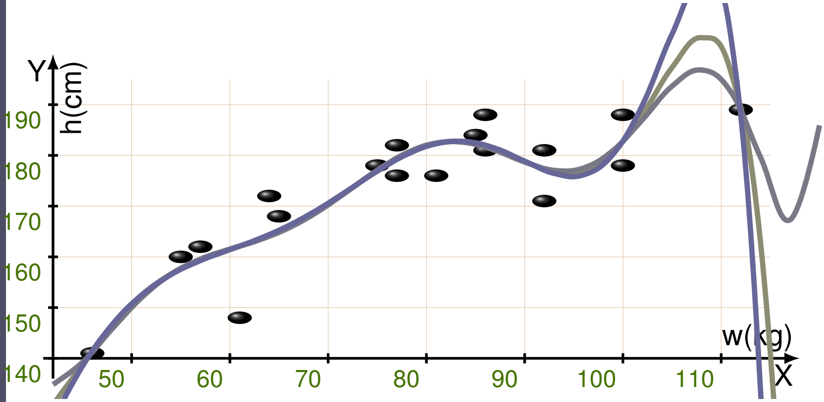
Aim: test different levels of flexibility. $\Rightarrow \quad p = 10$

Prior width: $\sigma_0^2 = 10^6 \quad \sigma_0^2 = 10^5 \quad \sigma_0^2 = 10^4 \quad \sigma_0^2 = 10^3$
$\sigma_0^2 = 10^2$



Aim: test different levels of flexibility. $\Rightarrow \quad p = 10$

Prior width: $\sigma_0^2 = 10^6 \quad \sigma_0^2 = 10^5 \quad \sigma_0^2 = 10^4 \quad \sigma_0^2 = 10^3$
$\sigma_0^2 = 10^2 \quad \sigma_0^2 = 10^1$



Aim: test different levels of flexibility. $\Rightarrow$ $p = 10$

Prior width: $\sigma_0^2 = 10^6$ $\sigma_0^2 = 10^5$ $\sigma_0^2 = 10^4$ $\sigma_0^2 = 10^3$ $\sigma_0^2 = 10^2$ $\sigma_0^2 = 10^1$ $\sigma_0^2 = 10^0$

$$\boldsymbol{\theta}^*_{MAP} = \underset{\boldsymbol{\theta}\in\Omega}{\operatorname{argmax}}\, K - \frac{1}{2}\sum_n E_2(y_n, f(\boldsymbol{x}_n, \boldsymbol{\theta})) - \frac{\|\boldsymbol{\theta}\|^2}{2\sigma_0^2}$$

Transform into vector notation:

$$\boldsymbol{\theta}^*_{MAP} = \underset{\boldsymbol{\theta}\in\Omega}{\operatorname{argmax}}\, K - \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) - \frac{\boldsymbol{\theta}^T\boldsymbol{\theta}}{2\sigma_0^2}$$

solve for $\boldsymbol{\theta}$ by differentiation:

$$\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) - \frac{1}{\sigma_0^2}\boldsymbol{I}_d\boldsymbol{\theta} = 0$$

$$\boldsymbol{\theta}^*_{MAP} = \left(\boldsymbol{X}^T\boldsymbol{X} + \frac{1}{\sigma_0^2}\boldsymbol{I}_d\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

again M.L. for $\sigma_0^2 = \infty$

Mximum a–posteriori models:

- Allow for the inclusion of prior knowledge;
- May protect against overfitting;
- Can measure the fitness of the family to the data;
  Procedure called **M.L. type II**.

M.A.P. application

Idea: instead of computing the most probable value of $\boldsymbol{\theta}$, we can measure the fit of the model $\mathcal{F}$ to the data $\mathcal{D}$.

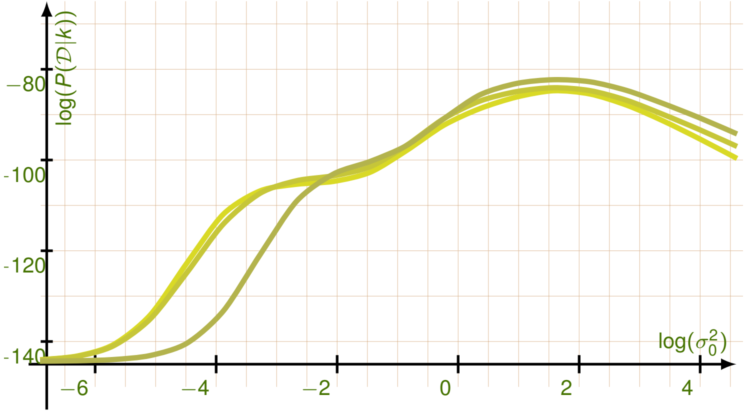
$$
\begin{aligned}
P(\mathcal{D}|\mathcal{F}) &= \sum_{\boldsymbol{\theta}_\ell \in \Omega} p(\mathcal{D}, \boldsymbol{\theta}_\ell | \mathcal{F}) \\
&= \sum_{\boldsymbol{\theta}_\ell \in \Omega} p(\mathcal{D} \,|\, \boldsymbol{\theta}_\ell, \mathcal{F}) p_0(\boldsymbol{\theta}_\ell | \mathcal{F})
\end{aligned}
$$

Gaussian noise and polynomial of order K:

$$
\begin{aligned}
\log(P(\mathcal{D}|\mathcal{F})) &= \log\left(\int_{\Omega_{\boldsymbol{\theta}}} d\theta \; \frac{p(\mathcal{D}\,|\,\boldsymbol{\theta},\mathcal{F}) p_0(\boldsymbol{\theta}\,|\,\mathcal{F})}{p(\mathcal{D}\,|\,\mathcal{F})}\right) = \log\left(\mathrm{N}(\boldsymbol{y}|0, \boldsymbol{\Sigma}_X)\right) \\
&= -\frac{1}{2}\left(N \log(2\pi) + \log|\boldsymbol{\Sigma}_X| + \boldsymbol{y}^T \boldsymbol{\Sigma}_X^{-1} \boldsymbol{y}\right)
\end{aligned}
$$

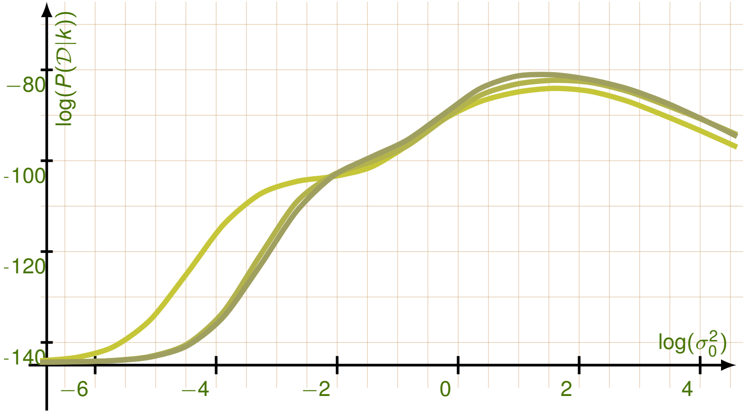
where

$$
\boldsymbol{\Sigma}_X = \boldsymbol{I}_N \sigma_n^2 + \boldsymbol{X}\boldsymbol{\Sigma}_0 \boldsymbol{X}^T \quad \text{with} \quad
\begin{aligned}
\boldsymbol{X} &= \left[\boldsymbol{x}^0, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^K\right] \\
\boldsymbol{\Sigma}_0 &= \mathrm{diag}(\sigma_0^2, \sigma_1^2, \ldots, \sigma_K^2)
\end{aligned}
$$

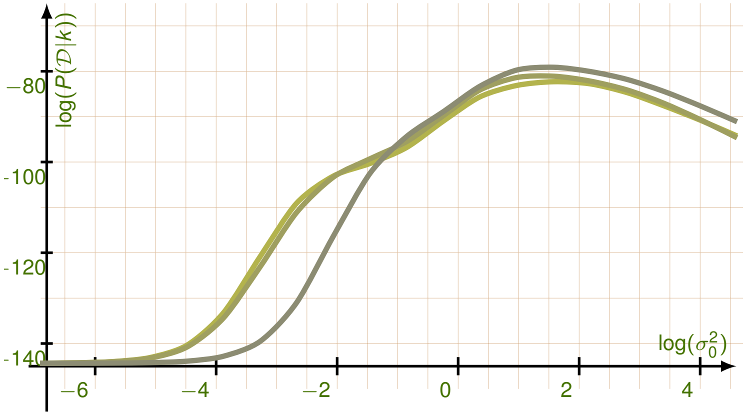


Aim: test different models.

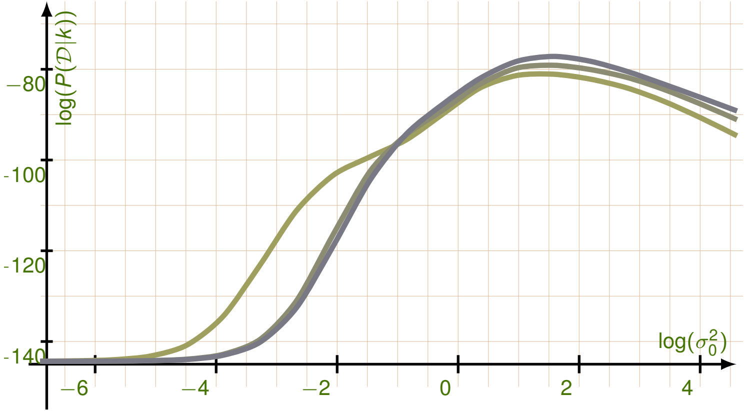
Polynomial families: $k = 10$ $k = 9$ $k = 8$.



Aim: test different models.
Polynomial families: $k = 9$ $k = 8$ $k = 7$.

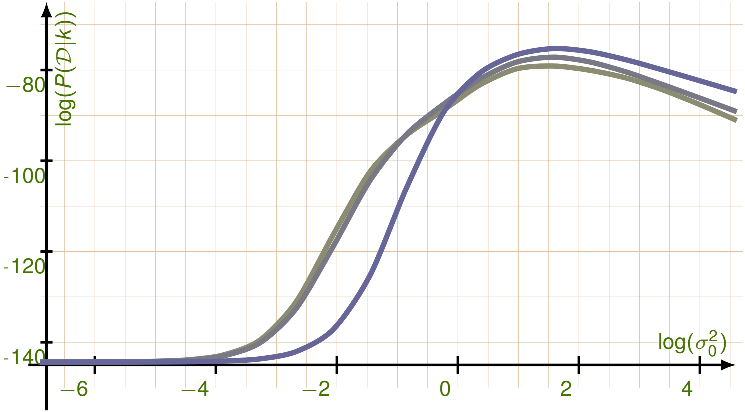


Aim: test different models.
Polynomial families: $k = 8$ $k = 7$ $k = 6$.



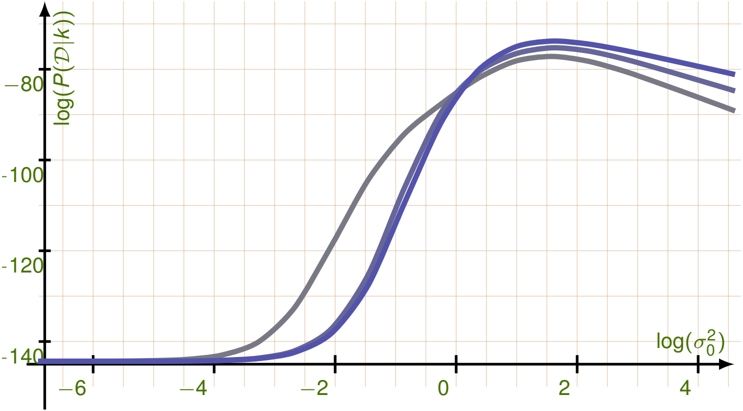
Aim: test different models.

Polynomial families: $k = 7$ $k = 6$ $k = 5$.

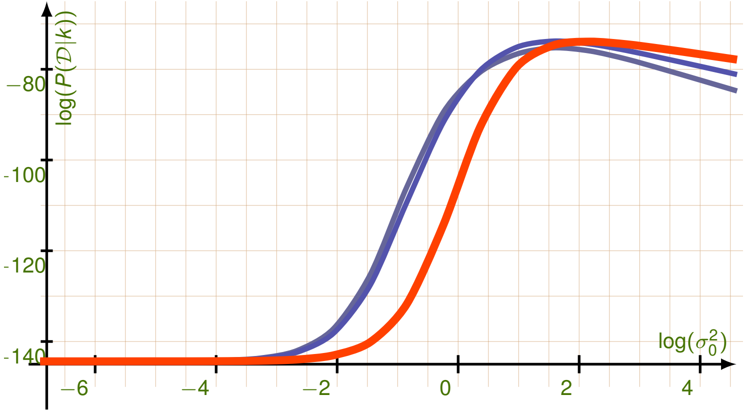


Aim: test different models.

Polynomial families: $k = 6$ $k = 5$ $k = 4$.



Aim: test different models.
Polynomial families: $k = 5$ $k = 4$ $k = 3$.



Aim: test different models.

Polynomial families: $k = 4$ $k = 3$ $k = 2$.

Probabilistic Data Mining

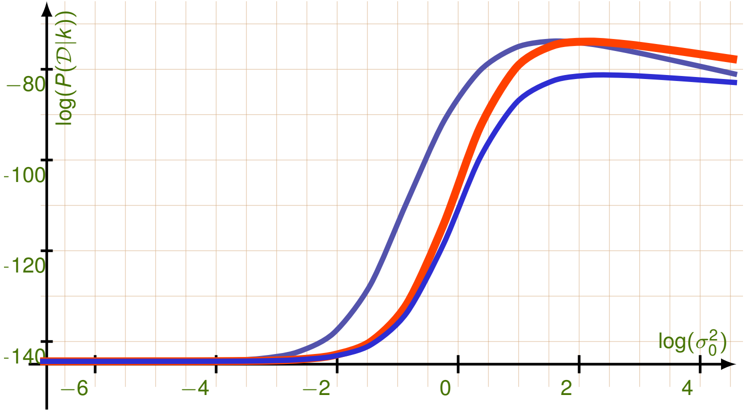
Lehel Csató

Modelling Data
Machine Learning
Latent variable models

Estimation
Maximum Likelihood
Maximum a-posteriori
Bayesian Estimation

Unsupervised
General concepts
Principal Components
Independent Components
Mixture Models

References



Aim: test different models.

Polynomial families: $k = 3$ $k = 2$ $k = 1$.

- M.L. and M.A.P. estimates provide single solutions.
- Point estimates lack the assessment of un/certainty.
- Better solution:
  for a query $\mathbf{x}_*$, the system output is probabilistic:

$$x_* \quad \Rightarrow \quad p(y_* | \mathbf{x}_*, \mathcal{F})$$

- Tool:
  go beyond the M.A.P. solution and use the **a–posteriori distribution** of the parameters.

We again use Bayes' rule:

$$p(\boldsymbol{\theta}|\mathcal{D},\boldsymbol{\mathcal{F}}) = \frac{P(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})}{p(\mathcal{D}|\boldsymbol{\mathcal{F}})} \quad \text{with } p(\mathcal{D}|\boldsymbol{\mathcal{F}}) = \int_\Omega d\boldsymbol{\theta}\; P(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta}).$$

and exploit the whole posterior distribution of the parameters.

A-posteriori parameter estimates

We operate with $p_{\text{post}}(\boldsymbol{\theta}) \stackrel{\text{def}}{=} p(\boldsymbol{\theta}|\mathcal{D},\boldsymbol{\mathcal{F}})$ and use the total probability rule:

$$p(y_*|\mathcal{D},\boldsymbol{\mathcal{F}}) = \sum_{\boldsymbol{\theta}_\ell \in \Omega_{\boldsymbol{\theta}}} p(y_*|\boldsymbol{\theta}_\ell,\boldsymbol{\mathcal{F}})\, p_{\text{post}}(\boldsymbol{\theta}_\ell)$$

in assessing system output.

Given the data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ estimate the linear fit:

$$y = \theta_0 + \sum_{i=1}^{d} \theta_i x_i = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}^T \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \stackrel{\text{def}}{=} \boldsymbol{\theta}^T \mathbf{x}$$

Gaussian distributions noise and prior:

$$\epsilon = y_n - \boldsymbol{\theta}^T \mathbf{x}_n \sim \mathrm{N}(0, \sigma_n^2)$$
$$\mathbf{w} \sim \mathrm{N}(0, \boldsymbol{\Sigma}_0)$$

Goal: compute the posterior distribution $p_{\text{post}}(\boldsymbol{\theta})$.

$$p_{\text{post}}(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta})\, p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{F}) = p_0(\boldsymbol{\theta}\,|\boldsymbol{\Sigma}_0) \prod_{n=1}^{N} P(y_n|\boldsymbol{\theta}^T \boldsymbol{x}_n)$$

$$\begin{aligned}
-2\log\left(p_{\text{post}}(\boldsymbol{\theta})\right) &= K_{\text{post}} + \frac{1}{\sigma_n^2}\left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\right)^T \left(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\right) + \boldsymbol{\theta}^T \boldsymbol{\Sigma}_0^{-1} \boldsymbol{\theta} \\
&= \boldsymbol{\theta}^T \left(\frac{1}{\sigma_n^2}\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{\Sigma}_0^{-1}\right)\boldsymbol{\theta} - \frac{2}{\sigma_n^2}\boldsymbol{\theta}^T\boldsymbol{X}^T\boldsymbol{y} + K'_{\text{post}} \\
&= \left(\boldsymbol{\theta} - \boldsymbol{\mu}_{\text{post}}\right)^T \boldsymbol{\Sigma}_{\text{post}}^{-1} \left(\boldsymbol{\theta} - \boldsymbol{\mu}_{\text{post}}\right) + K''_{\text{post}}
\end{aligned}$$

and by identification

$$\boldsymbol{\Sigma}_{\text{post}} = \left(\frac{1}{\sigma_n^2}\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{\Sigma}_0^{-1}\right)^{-1} \quad \text{and} \quad \boldsymbol{\mu}_{\text{post}} = \boldsymbol{\Sigma}_{\text{post}} \frac{\boldsymbol{X}^T\boldsymbol{y}}{\sigma_n^2}$$

Bayesian linear model

The posterior distribution for the parameters is a Gaussian with parameters

$$\Sigma_{\text{post}} = \left(\frac{1}{\sigma_n^2} X^T X + \Sigma_0^{-1} \right)^{-1} \quad \text{and} \quad \mu_{\text{post}} = \Sigma_{\text{post}} \frac{X^T y}{\sigma_n^2}$$

Point estimates from keeping :

- M.L. if we take $\Sigma_0 \to \infty$ and considering only μ_{post}.
- M.A.P if we approximate the distribution with a single value at the maximum, *i.e.* μ_{post}.

Prediction for new values $\boldsymbol{x}_*$:

- use the likelihood $P(y_*|\boldsymbol{x}_*, \boldsymbol{\theta}, \mathcal{F})$,
- and the posterior for $\boldsymbol{\theta}$
- and Bayes' rule.

The steps:

$$p(y_*|\boldsymbol{x}_*, \mathcal{D}, \mathcal{F}) = \int_{\Omega_{\boldsymbol{\theta}}} d\theta \; p(y_*|\boldsymbol{x}_*, \boldsymbol{\theta}, \mathcal{F}) p_{\text{post}}(\boldsymbol{\theta} \,|\, \mathcal{D}, \mathcal{F})$$

$$= \int_{\Omega_{\boldsymbol{\theta}}} d\theta \; \exp\left[-\frac{1}{2}\left(K_* + \frac{(y_* - \boldsymbol{\theta}^T \boldsymbol{x}_*)^2}{\sigma_n^2} + (\boldsymbol{\theta} - \boldsymbol{\mu}_{\text{post}})^T \boldsymbol{\Sigma}_{\text{post}}^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_{\text{post}})\right)\right]$$

$$= \int_{\Omega_{\boldsymbol{\theta}}} d\theta \; \exp\left[-\frac{1}{2}\left(K_* + \frac{y_*^2}{\sigma_n^2} - \boldsymbol{a}^T \boldsymbol{C}^{-1} \boldsymbol{a} + Q(\boldsymbol{\theta})\right)\right]$$

where

$$\boldsymbol{a} = \frac{\boldsymbol{x}_* y_*}{\sigma_n^2} + \boldsymbol{\Sigma}_{\text{post}}^{-1} \boldsymbol{\mu}_{\text{post}} \qquad \boldsymbol{C} = \frac{\boldsymbol{x}_* \boldsymbol{x}_*^T}{\sigma_n^2} + \boldsymbol{\Sigma}_{\text{post}}$$

Integrating out the quadratic in $\boldsymbol{\theta}$:

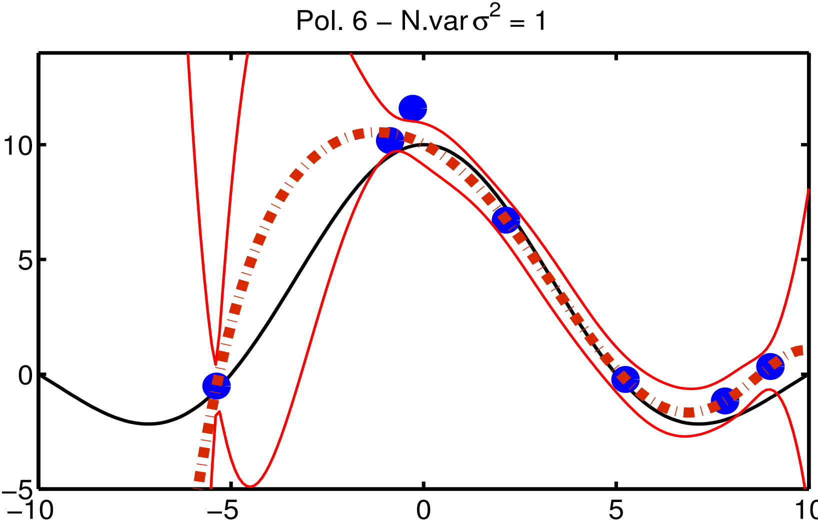
Predictive distribution at $\boldsymbol{x}_*$

$$p(y_*|\boldsymbol{x}_*, \mathcal{D}, \mathcal{F}) = \exp\left[-\frac{1}{2}\left(K_* + \frac{(y_* - \boldsymbol{x}_*\boldsymbol{\mu}_{\text{post}})^2}{\sigma_n^2 + \boldsymbol{x}_*^T\boldsymbol{\Sigma}_{\text{post}}^{-1}\boldsymbol{x}_*}\right)\right]$$

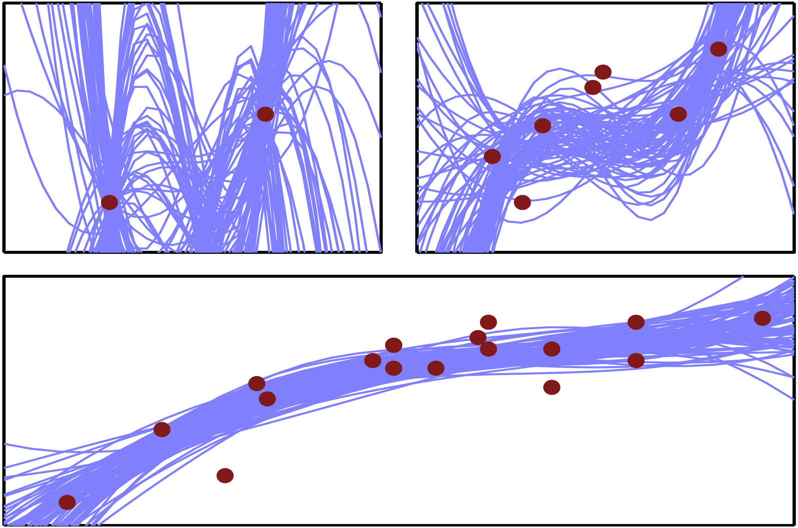
$$= \mathrm{N}\left(y_* \mid \boldsymbol{x}_*^T\boldsymbol{\mu}_{\text{post}}\ ,\ \sigma_n^2 + \boldsymbol{x}_*^T\boldsymbol{\Sigma}_{\text{post}}\boldsymbol{x}_*\right)$$

With the predictive distribution we:

- measure the variance of the prediction for each point: $\sigma_*^2 = \sigma_n^2 + \boldsymbol{x}_*^T\boldsymbol{\Sigma}_{\text{post}}\boldsymbol{x}_*$;
- sample from the parameters and plot the candidate predictors.



The errors are the symmetric thin lines.



Third order polynomials are used to approximate the data.

When computing $p_{\text{post}}(\boldsymbol{\theta}|\mathcal{D},\mathcal{F})$ we assumed that the posterior can be represented analytically.

This is not the case.

Approximations are needed for the

- posterior distribution
- predictive distribution

In Bayesian modelling an important issue is how we approximate the posterior distribution.

Complete specification of the model

Can include prior beliefs about the model.

Accurate predictions

Can include prior beliefs about the model.

Computational cost

Using models for prediction can be difficult and expensive in time and memory.

Complete specification of the model

Can include prior beliefs about the model.

Accurate predictions

Can include prior beliefs about the model.

Computational cost

Using models for prediction can be difficult and expensive in time and memory.

Bayesian models

Flexible and accurate – **if** priors about the model are used.
It can be expensive.

Outline

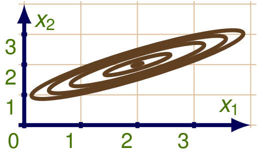
Unsupervised setting

- Data can be unlabeled, *i.e.* no values y are associated to an input $\mathbf{x}$.
- We want to "extract" information from $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.
- We assume that the data – although high-dimensional – spans a **manifold** of a much smaller dimension.
- Task is to find the subspace corresponding to the *data span*.

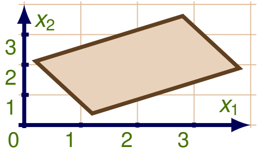
Models in unsupervised learning

It is again important the **model of the data**:

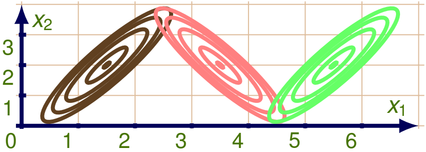
- Principal Components;



- Independent Components;

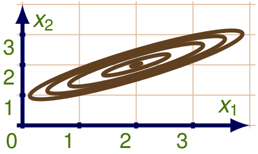


- Mixture models;



The PCA model I

- Simple data structure.
- Spherical cluster that is:
  - translated;
  - scaled;
  - rotated.



We aim to find the principal directions of the data spread.

Principal direction:

the direction $\mathbf{u}$ along which the data preserves most of its *variance*.

Principal direction:

$$\boldsymbol{u} = \underset{\|\boldsymbol{u}\|=1}{\operatorname{argmax}} \frac{1}{2N} \sum_{n=1}^{N} (\boldsymbol{u}^T \boldsymbol{x}_n - \boldsymbol{u}\bar{\boldsymbol{x}})^2$$

we pre-process: $\bar{\boldsymbol{x}} = \mathbf{0}$. Replacing the empirical covariance with $\boldsymbol{\Sigma}_{\boldsymbol{x}}$:

$$\begin{aligned} \boldsymbol{u} &= \underset{\|\boldsymbol{u}\|=1}{\operatorname{argmax}} \frac{1}{2N} \sum_{n=1}^{N} (\boldsymbol{u}^T \boldsymbol{x}_n - \boldsymbol{u}\bar{\boldsymbol{x}})^2 \\ &= \underset{\boldsymbol{u},\lambda}{\operatorname{argmax}} \frac{1}{2} \boldsymbol{u}^T \boldsymbol{\Sigma}_{\boldsymbol{x}} \boldsymbol{u} - \lambda(\|\boldsymbol{u}\|^2 - \mathbf{1}) \end{aligned}$$

with λ the Lagrange multiplier. Differentiating w.r.t $\boldsymbol{u}$:

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} \boldsymbol{u} - \lambda \boldsymbol{u} = \mathbf{0}$$

The optimum solution **must obey**:

$$\boldsymbol{\Sigma}_x \boldsymbol{u} = \lambda \boldsymbol{u}$$

The **eigendecomposition** of the covariance matrix.

$(\lambda_*, \boldsymbol{u}_*)$ is an eigenvalue, eigenvector of the system.

If we replace back, the value of the expression is λ_*.
$\Rightarrow$
Optimal solution when $\lambda_* = \lambda_{max}$.

Principal direction:

The eigenvector $\boldsymbol{u}_{max}$ corresponding to the largest eigenvalue of the system.

Probabilistic Data Mining
Lehel Csató
Modelling Data
Machine Learning
Latent variable models
Estimation
Maximum Likelihood
Maximum a-posteriori
Bayesian Estimation
Unsupervised
General concepts
Principal Components
Independent Components
Mixture Models
References

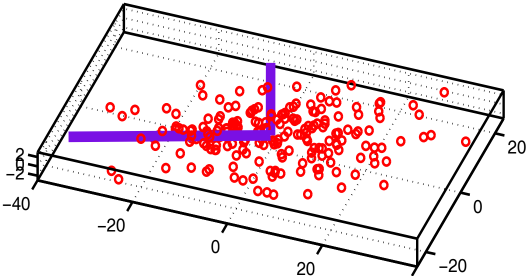
How is this used in data mining?

Assume that data is:

- jointly Gaussian:

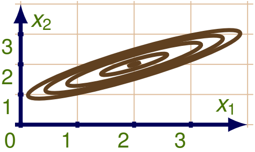
$$\boldsymbol{x} = \mathrm{N}(\boldsymbol{m}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}}),$$

- high-dimensional;
- only few (2) directions are relevant.



How is this used in data mining?

- Subtracting mean.
- Eigendecomposition.
- Selecting the K eigenvectors corresponding to the K largest values.
- Computing the K projections: $z_{n\ell} = \boldsymbol{x}_n^T \boldsymbol{u}_\ell$.



The projection using matrix $\boldsymbol{P} \stackrel{\text{def}}{=} [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K]^T$:

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{P}$$

and $\boldsymbol{z}_n$ can is used as a compact representation of $\boldsymbol{x}_n$.

Reconstruction:

$$\boldsymbol{x}'_n = \sum_{\ell=1}^{K} z_{n\ell}\boldsymbol{u}_\ell \quad \text{or, with matrix notation:} \quad \boldsymbol{X}' = \boldsymbol{Z}\boldsymbol{P}^T$$

PCA projection analysis:

$$\begin{aligned}
E_{PCA} &= \frac{1}{N^2}\sum_{n=1}^{N}\left(\boldsymbol{x}_n - \boldsymbol{x}'_n\right)^2 = \frac{1}{N^2}\operatorname{tr}\left[\left(\boldsymbol{X}-\boldsymbol{X}'\right)^T\left(\boldsymbol{X}-\boldsymbol{X}'\right)\right] \\
&= \operatorname{tr}\left[\boldsymbol{\Sigma_x} - \boldsymbol{P}^T\boldsymbol{\Sigma_z}\boldsymbol{P}\right] \\
&= \operatorname{tr}\left[\boldsymbol{U}\left(\operatorname{diag}(\lambda_1,\ldots,\lambda_d) - \operatorname{diag}(\lambda_1,\ldots,\lambda_K,0,\ldots)\right)\boldsymbol{U}^T\right] \\
&= \operatorname{tr}\left[\boldsymbol{U}^T\boldsymbol{U}\operatorname{diag}(0,\ldots,0,\lambda_{K+1},\ldots,\lambda_d)\right] \\
&= \sum_{\ell=1}^{d-K}\lambda_{K+\ell}
\end{aligned}$$

Probabilistic Data Mining

Lehel Csató

Modelling Data
Machine Learning
Latent variable models

Estimation
Maximum Likelihood
Maximum a-posteriori
Bayesian Estimation

Unsupervised
General concepts
Principal Components
Independent Components
Mixture Models

References

PCA reconstruction error:

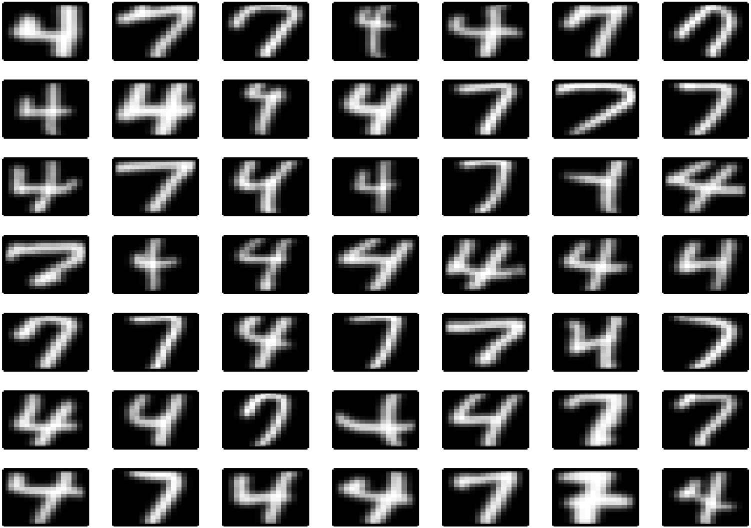
The error made using the PCA directions:

$$E_{PCA} = \sum_{\ell=1}^{d-K} \lambda_{K+\ell}$$

PCA properties:

- PCA system orthonormal: $\boldsymbol{u}_\ell^T \boldsymbol{u}_r = \delta_{\ell-r}$
- Reconstruction fast.
- Spherical assumption critical.

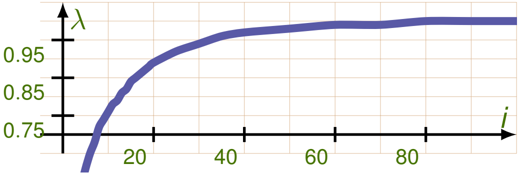
USPS digits – testbed for several models.



USPS characteristics:

- handwritten data centered and scaled;
- $\approx$ 10.000 items of 16×16 grayscale images;

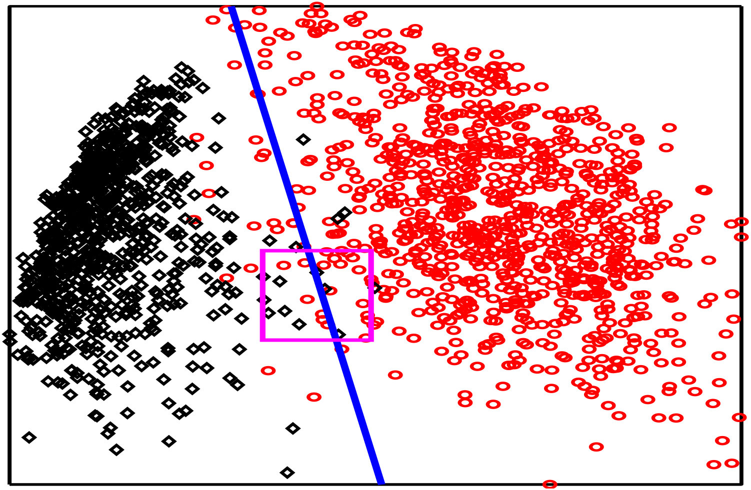
We plot $k_r = \sum_{\ell=1}^{r} \lambda_\ell$ - normalised



Conclusion for the USPS set:

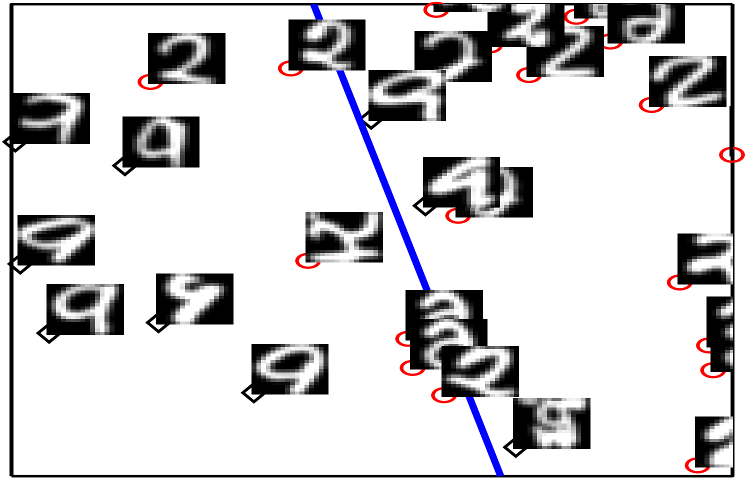
- The normalised $\lambda_1 = 0.24 \quad \Rightarrow \quad \boldsymbol{u}_1$ accounts for 24% of the data.
- at $\approx$ 10 more than 70% of variance is explained.
- at $\approx$ 50 more than 98% $\Rightarrow$ 50 numbers instead of 256.

Visualisation application:



Visualisation along the **first two** eigendirections.

Visualisation application:



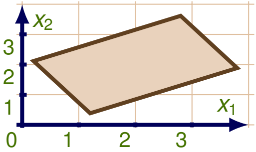
Detail.

The ICA model I

Start from the PCA:

$$x = Pz$$

is a **generative model** for the data.



We assumed that

- z i.i.d. Gaussian random variables $z \sim \mathcal{N}(\mathbf{0}, \text{diag}(\lambda_\ell))$;
- $\Rightarrow$ x are not independent;
- $\Rightarrow$ z are **Gaussian** sources;

In most of real data:

- Sources are **not Gaussian**.
- But sources are **independent**.
- We exploit that!.

The ICA model II

The following model assumption:

$$x = As$$

where

- z independent **sources**;
- A linear **mixing matrix**;

Looking for matrix B that recovers the sources:

$$s' \stackrel{\text{def}}{=} Bx = B(As) = (BA)s$$

i.e. (BA) is a permutation and scaling but retains **independence.**

The ICA model III

In practice:

$$\mathbf{s}' \stackrel{\text{def}}{=} \mathbf{B}\mathbf{x}$$

with $\mathbf{s} = [s_1, \ldots, s_K]$ all independent sources.
Independence test: the KL-divergence between the **joint distribution** and the **marginals**

$$\mathbf{B} = \operatorname*{argmin}_{\mathbf{B} \in \mathrm{SO}_d} \mathrm{KL}\left(p(s_1, s_2) \| p(s_1)p(s_2)\right)$$

where SO_d is the group of matrices with $|\mathbf{B}| = 1$.

In ICA we are looking for matrix $\mathbf{B}$ that minimises:

$$\sum_{\ell} \int_{\Omega_\ell} ds_\ell \log p(s_\ell) - \int_{\Omega_\ell} d\mathbf{s} \log(p(s_1, \ldots, s_d))$$

Kullback-Leibler divergence

$$\mathrm{KL}(p\|q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

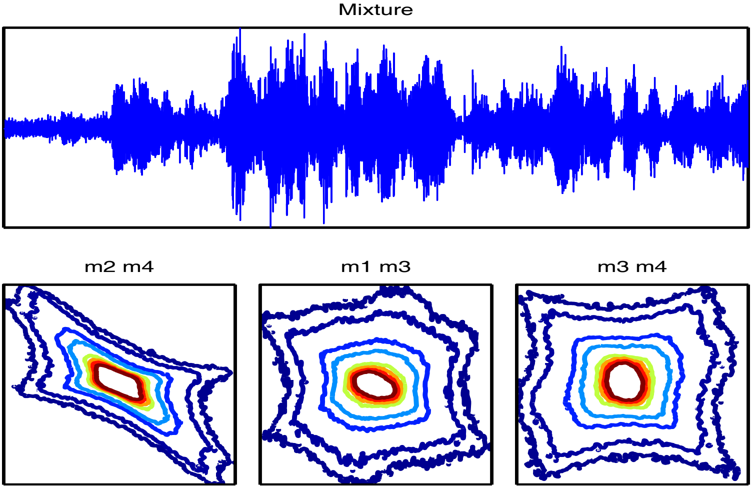
- is zero only and only if $p = q$,
- is *not* a measure of distance (but cloooose to it!),
- Efficient when exponential families are used.

Short proof:

$$0 = \log 1 = \log \left(\sum_x q(x) \right) = \log \left(\sum_x p(x) \frac{q(x)}{p(x)} \right)$$
$$\geq \sum_x p(x) \log \left(\frac{q(x)}{p(x)} \right) = -\mathrm{KL}(p\|q)$$

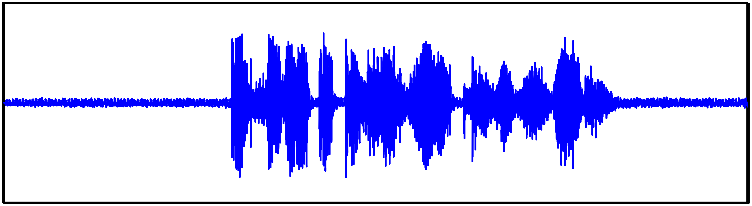
$$\Rightarrow \qquad \mathrm{KL}(p\|q) \geq 0$$

Separation of source signals:

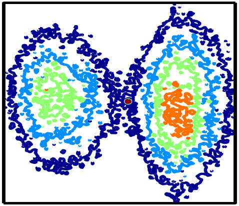


Results of separation:

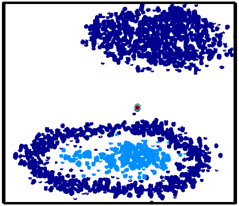
Source



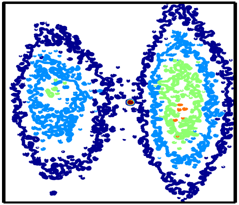
m2 m4



m1 m3



m3 m4



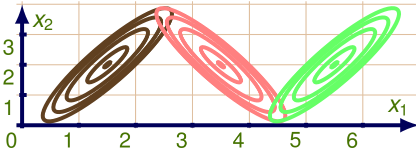
FastICA package

Applications of ICA

Applications:

- Coctail party problem;
  Separates noisy and multiple sources from multiple observations.
- Fetus ECG;
  Separation of the ECG signal of a fetus from its mother's ECG.
- MEG recordings;
  Separation of MEG "sources".
- Financial data;
  Finding hidden factors in financial data.
- Noise reduction;
  Noise reduction in natural images.
- Interference removal;
  Interference removal from CDMA – Code-division multiple access – communication systems.

- The data structure is more complex.
- More than a single source for data.



The mixture model:

$$P(\mathbf{x}|\Sigma) = \sum_{k=1}^{K} \pi_k \, p_k(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \tag{1}$$

where:

$\pi_1, \ldots, \pi_K$ – mixing components.

$p_k(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)$ – density of a component.

The components are usually called clusters.

The generation process reflects the assumptions about the model.

The data generation:

- first we select from which component,
- then we sample from the component's density function.

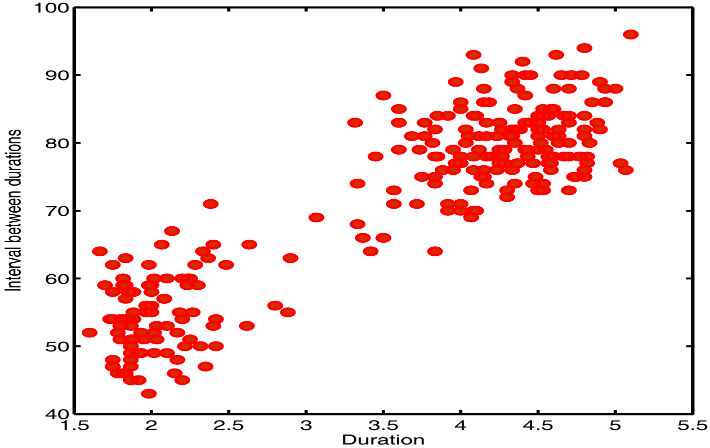
When modelling data we **do not know:**

- Which point belongs to which cluster.
- What are the parameters for each density function.

Old Faithful geyser in
the Yellowstone National park.
Characterised by:

- intense eruptions;
- differing times between them.

Rule:
Duration is 1.5 to 5 minutes.
The length of eruption helps determine the interval.
If an eruption lasts less than 2 minutes the interval will be around 55 minutes. If the eruption last 4.5 minutes the interval may be around 88 minutes.



- The longer the duration, the longer the interval.
- The linear relation $I = \theta_0 + \theta_1 d$ is not the best.
- There are only a very few eruptions lasting ≈ 3 minutes.

The mixture model I

Assumptions:

- We know the family of individual density functions:
  These density functions are parametrised with a few parameters.

- The densities are easily identifiable:
  If we knew which data belongs to which cluster, the density function is easily identifiable.

Gaussian densities are often used – fulfill both "conditions".

The mixture model II

The Gaussian mixture model:

$$p(\mathbf{x}) = \pi_1 \mathrm{N}_1(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \pi_2 \mathrm{N}_2(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

for **known** densities
(centres and ellipses):

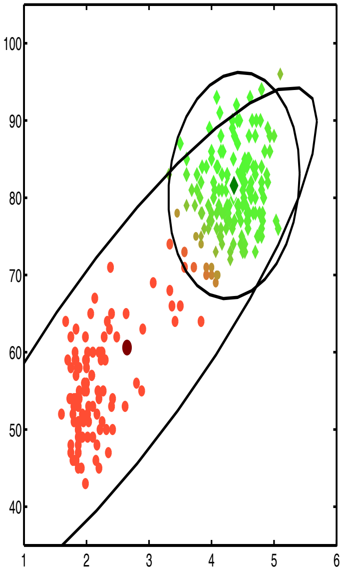
$$p(\mathbf{x}_n|k) = \frac{\mathrm{N}_k(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\, p(k)}{\sum_\ell \mathrm{N}_\ell(\mathbf{x}_n|\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)\, p(\ell)}$$

i.e. we know the **probability** that data comes from cluster k
(shades from red to green).

For $\mathcal{D}$:

$\mathbf{x}$	$p(\mathbf{x}\|1)$	$p(\mathbf{x}\|2)$
$\mathbf{x}_1$	γ_{11}	γ_{12}
$\vdots$	$\vdots$	$\vdots$
$\mathbf{x}_N$	γ_{N1}	γ_{N2}

$\gamma_{n\ell}$ – responsibility of $\mathbf{x}_n$ in cluster ℓ.



The mixture model III

When $\gamma_{n\ell}$ known, the parameters are computed using the data weighted by their responsibilities:

$$(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \operatorname*{argmax}_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \prod_{n=1}^{N} \left(\mathrm{N}_k(\boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma})\right)^{\gamma_{nk}}$$

for all k.

This means:

$$(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \Leftarrow \sum_n \gamma_{nk} \log \mathrm{N}(\boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

When making inference

Have to find the responsibility vector **and** the parameters of the mixture.

The mixture model III

When $\gamma_{n\ell}$ known, the parameters are computed using the data weighted by their responsibilities:

$$(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \operatorname*{argmax}_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \prod_{n=1}^{N} \left(\mathrm{N}_k(\boldsymbol{x}_n|\boldsymbol{\mu},\boldsymbol{\Sigma})\right)^{\gamma_{nk}}$$

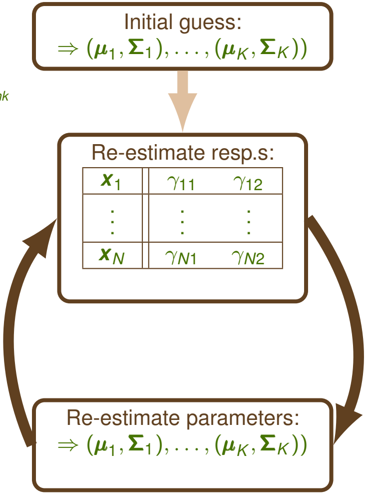
for all k.
This means:

$$(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \;\Leftarrow\; \sum_n \gamma_{nk} \log \mathrm{N}(\boldsymbol{x}_n|\boldsymbol{\mu},\boldsymbol{\Sigma})$$

When making inference

Have to find the responsibility vector **and** the parameters of the mixture.

Given data $\mathcal{D}$:



Responsibilities γ

The additional **latent** variables needed to help computation.

In the mixture model:

- goal is to *fit* model to data;
- which submodel gets a particular data;

Achieved by the maximisation of the log-likelihood function.

Probabilistic Data Mining

Lehel Csató

Modelling Data
Machine Learning
Latent variable models

Estimation
Maximum Likelihood
Maximum a-posteriori
Bayesian Estimation

Unsupervised
General concepts
Principal Components
Independent Components
Mixture Models

References

$$(\boldsymbol{\pi}, \boldsymbol{\Theta}) = \text{argmax} \sum_n \log \left[\sum_\ell \pi_\ell \mathrm{N}_\ell(\boldsymbol{x}_n | \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell) \right]$$

$\boldsymbol{\Theta} = [\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K]$ is the vector of parameters;
$\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]$ the shares of the factors;

Problem with optimisation:
The parameters are not separable due to the sum within the logarithm.

Solution:
Use an approximation.

The EM algorithm II

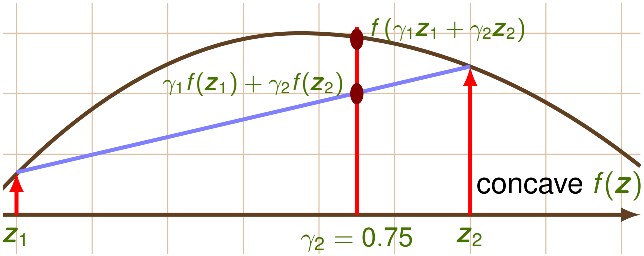
$$\log P(\mathcal{D}|\boldsymbol{\pi}, \boldsymbol{\Theta}) = \sum_n \log \left[\sum_\ell \pi_\ell \mathrm{N}_\ell(\mathbf{x}_n|\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell) \right] = \sum_n \log \left[\sum_\ell p_\ell(\mathbf{x}_n, \ell) \right]$$

Use Jensen's inequality:

$$\log \left(\sum_\ell p_\ell(\mathbf{x}_n, \ell|\theta_\ell) \right) = \log \left(\sum_\ell q_n(\ell) \frac{p_\ell(\mathbf{x}_n, \ell|\theta_\ell)}{q_n(\ell)} \right) \geq \sum_\ell q_n(\ell) \log \left(\frac{p_\ell(\mathbf{x}_n, \ell)}{q_n(\ell)} \right)$$

for **any** $[q_n(1), \ldots, q_n(\ell)]$.

skip Jensen



Jensen's Inequality

For any concave $f(z)$, any z_1 and z_2, and any $\gamma_1, \gamma_2 > 0$ such that $\gamma_1 + \gamma_2 = 1$:

$$f(\gamma_1 z_1 + \gamma_2 z_2) \geq \gamma_1 f(z_1) + \gamma_2 f(z_2)$$

The EM algorithm III

$$\log\left(\sum_{\ell} p_\ell(\mathbf{x}_n,\ell|\theta_\ell)\right) \geq \sum_{\ell} q_n(\ell) \log\left(\frac{p_\ell(\mathbf{x}_n,\ell)}{q_n(\ell)}\right)$$

for any distribution $q_n(\cdot)$.
Replacing with the right-hand side, we have:

$$\log P(\mathcal{D}|\boldsymbol{\pi},\boldsymbol{\Theta}) \geq \sum_n \sum_\ell q_n(\ell)\log\frac{p_\ell(\mathbf{x}_n|\boldsymbol{\theta}_\ell)}{q_n(\ell)}$$

$$\geq \sum_\ell \left[\sum_n q_n(\ell)\log\frac{p_\ell(\mathbf{x}_n|\boldsymbol{\theta}_\ell)}{q_n(\ell)}\right] = \mathcal{L}$$

and therefore the optimisation w.r.to cluster parameters separate.

$$\partial_\ell \quad \Rightarrow \quad 0 = \sum_n q_n(\ell)\frac{\partial \log p_\ell(\mathbf{x}_n|\boldsymbol{\theta}_\ell)}{\partial \boldsymbol{\theta}_\ell}$$

For distributions from exponential family optimisation is easy.

The EM algorithm IV

- **any** set of distributions $q_1(\ell), \ldots, q_N(\ell)$ provides a lower bound to the log-likelihood.
- We should choose the distributions so that they are the closest to the **current** parameter set.

We assume the **parameters** have the value $\boldsymbol{\theta}_0$.

Want to minimise the difference:

$$\log P(\mathbf{x}_n, \ell|\boldsymbol{\theta}_\ell^0) - \mathcal{L} = \sum_\ell q_n(\ell) \log P(\mathbf{x}_n, \ell|\boldsymbol{\theta}_\ell^0) - \sum_\ell q_n(\ell) \log \frac{p_\ell(\mathbf{x}_n, \ell|\boldsymbol{\theta}_\ell^0)}{q_n(\ell)}$$

$$\sum_\ell q_n(\ell) \log \frac{P(\mathbf{x}_n, \ell|\boldsymbol{\theta}_\ell^0) q_n(\ell)}{p_\ell(\mathbf{x}_n|\boldsymbol{\theta}_\ell^0)}$$

and observe that by **setting**

$$q_n(\ell) = \frac{p_\ell(\mathbf{x}_n|\boldsymbol{\theta}_\ell^0)}{P(\mathbf{x}_n, \ell|\boldsymbol{\theta}_\ell^0)}$$

we have $\sum_\ell q_n(\ell)\ 0 = 0$.

The EM algorithm:

Init – initialise model parameters;

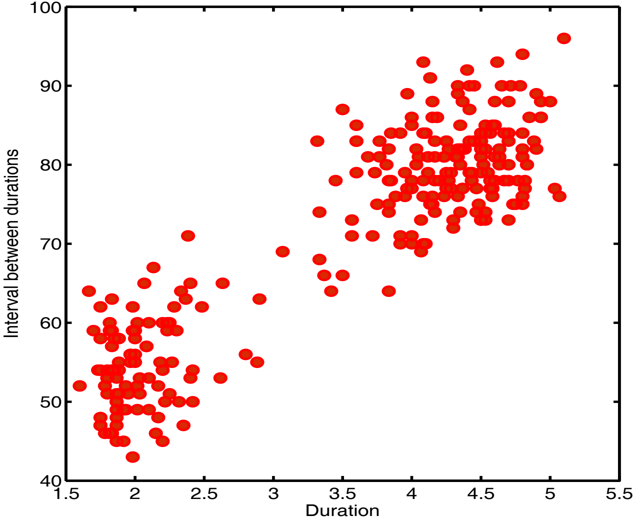
E step – compute the responsibilities $\gamma_{n\ell} = q_n(\ell)$;

M step – for each k optimize

$$0 = \sum_n q_n(\ell) \frac{\partial \log p_\ell(\mathbf{x}_n|\boldsymbol{\theta}_\ell)}{\partial \boldsymbol{\theta}_\ell}$$

repeat – goto the **E step**.

Old faithful:



Probabilistic Data Mining

Lehel Csató

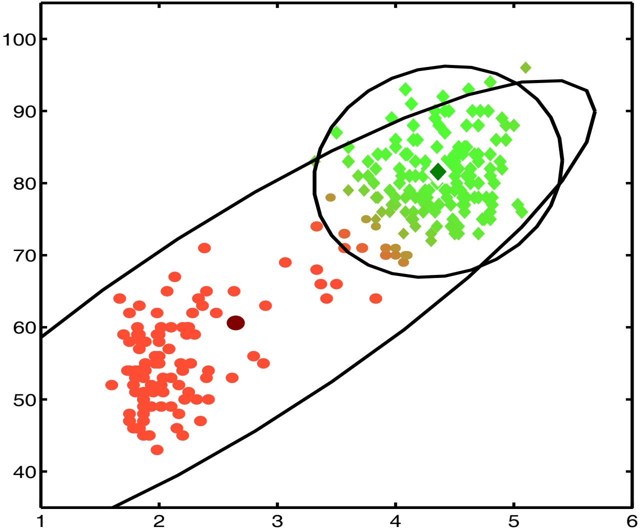
Modelling Data
Machine Learning
Latent variable models

Estimation
Maximum Likelihood
Maximum a-posteriori
Bayesian Estimation

Unsupervised
General concepts
Principal Components
Independent Components
Mixture Models

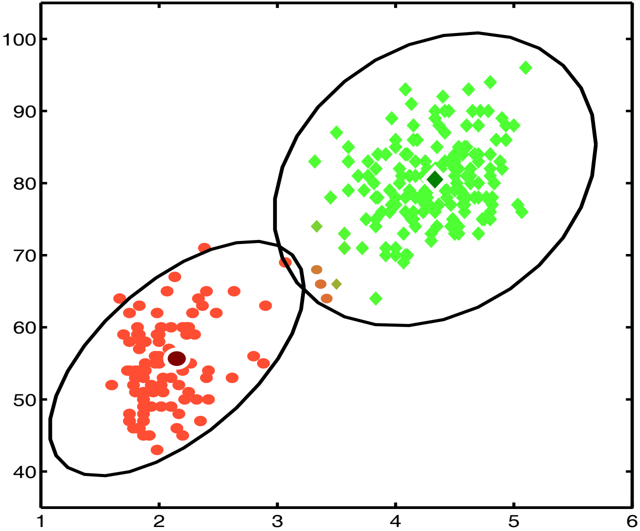
References

Old faithful:

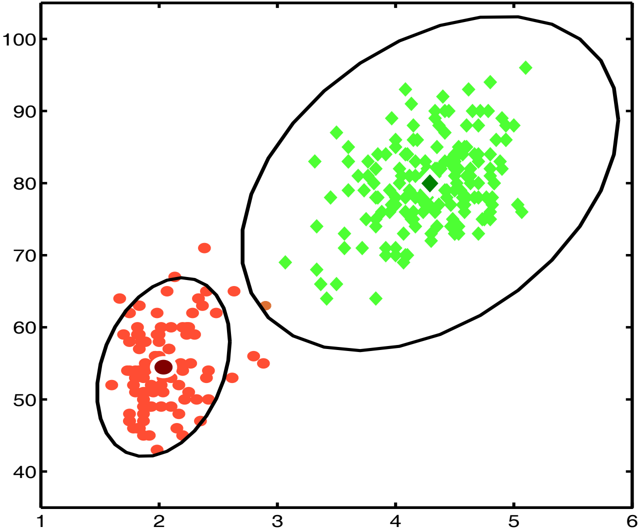


EM application III

Old faithful:



Old faithful:



References

- J. M. Bernardo and A. F. Smith.
  Bayesian Theory.
  John Wiley & Sons, 1994.

- C. M. Bishop.
  Pattern Recognition and Machine Learning.
  Springer Verlag, New York, N.Y., 2006.

- T. M. Cover and J. A. Thomas.
  Elements of Information Theory.
  John Wiley & Sons, 1991.

- A. P. Dempster, N. M. Laird, and D. B. Rubin.
  Maximum likelihood from incomplete data via the EM algorithm.
  Journal of the Royal Statistical Society series B, 39:1–38, 1977.

- T. Hastie, R. Tibshirani, és J. Friedman.
  The Elements of Statistical Learning: Data Mining, Inference, and Prediction.
  Springer Verlag, 2001.