# Machine Learning
# Maximum Likelihood Principle

Dmitrij Schlesinger

WS2013/2014, 8.11.2013

COMPUTER VISION LAB DRESDEN

TECHNISCHE UNIVERSITÄT DRESDEN

# Probabilistic Learning

Let a parameterized class (family) of probability distributions be given, i.e. $p(x;\theta) \in \mathcal{P}$

Example – the set of Gaussians in $\mathbb{R}^n$

$$p(x;\mu,\sigma) = \frac{1}{(\sqrt{2\pi}\sigma)^n} \exp\left[-\frac{\|x-\mu\|^2}{2\sigma^2}\right]$$

parameterized by the mean $\mu \in \mathbb{R}^n$ and standard deviation $\sigma \in \mathbb{R}$, i.e. $\theta = (\mu, \sigma)$.

Let the training data be given, e.g. $L = (x^1, x^2, \ldots, x^{|L|})$, e.g. $x^l \in \mathbb{R}^n$ for Gaussians

One have to decide for a particular probability distribution from the given family, i.e. for a particular (the "best") parameter, e.g. $\theta^* = (\mu^*, \sigma^*)$ for Gaussians.

# Maximum Likelihood Principle

**Assumption**: the training data is a realization of the unknown probability distribution – it is sampled according to it.

$\rightarrow$ what is observed should have a high probability

$\rightarrow$ maximize the probability of the training data with respect to the unknown parameter

$$p(L;\theta) \rightarrow \max_{\theta}$$

All further staff are just examples/special cases ...

# Discrete Probability Distributions

The free parameter is a "vector" of probability values

$$\theta = p(k) \in \mathbb{R}^{|K|}, \quad p(k) \geq 0, \quad \sum_k p(k) = 1$$

Training data: $L = (k^1, k^2, \ldots, k^{|L|})$, $k^l \in K$
Assumption (very often): independent examples

$$P(L;\theta) = \prod_l p(k^l) = \prod_k \prod_{l:k^l=k} p(k) = \prod_k p(k)^{n(k)}$$

with the frequencies $n(k)$ in the training data

$$\ln P(L;\theta) = \sum_k n(k) \ln p(k) \rightarrow \max_p$$

or (for infinite training data)

$$\ln P(L;\theta) = \sum_k p^*(k) \ln p(k) \rightarrow \max_p$$

# Shannon Lemma

$$\sum_i a_i \ln x_i \to \max_x, \quad \text{s.t.} \quad x_i \geq 0 \ \forall i, \quad \sum_i x_i = 1 \text{ with } \quad a_i \geq 0$$

Method of Lagrange coefficients:

$$F = \sum_i a_i \ln x_i + \lambda \Big( \sum_i x_i - 1 \Big) \to \min_\lambda \max_x$$

$$\frac{\partial F}{\partial x_i} = \frac{a_i}{x_i} + \lambda = 0 \quad //\text{Note: } \lambda \text{ is common for all } i$$

$$x_i = c \cdot a_i \quad \text{and} \quad \sum_i c \cdot a_i = 1$$

$$x_i = \frac{a_i}{\sum_{i'} a_{i'}}$$

---

Solution for general discrete probability distributions:
count the frequencies of $k$, normalize to sum to $1$.

# Probability Densities

Example – Gaussians

$$p(x;\mu,\sigma)=\frac{1}{(\sqrt{2\pi}\sigma)^n}\exp\Big[-\frac{\|x-\mu\|^2}{2\sigma^2}\Big],$$

i.e. $\theta=(\mu,\sigma)$, with $\mu\in\mathbb{R}^n$, $\sigma\in\mathbb{R}$.

$$\ln p(L;\mu,\sigma)=\sum_l\Big[-n\ln\sigma-\frac{\|x^l-\mu\|^2}{2\sigma^2}\Big]=$$

$$=-|L|\cdot n\cdot\ln\sigma-\frac{1}{2\sigma^2}\sum_l\|x^l-\mu\|^2\rightarrow\max_{\mu,\sigma}$$

$$\frac{d\ln p(L;\mu,\sigma)}{d\mu}=0\qquad\Rightarrow\qquad\mu=\frac{1}{|L|}\sum_l x^l$$

$$\frac{d\ln p(L;\mu,\sigma)}{d\sigma}=0\qquad\Rightarrow\qquad\sigma=\frac{1}{n\cdot|L|}\sum_l\|x^l-\mu\|^2$$

# “Mixed” models for recognition

$p(x,k;\theta) = p(k;\theta_a)\cdot p(x|k;\theta_k)$, with $k \in K$ (classes, usually discrete) and $x \in X$ (observations, general)

Unknown parameters are $\theta_a = p(k)$ and class-specific $\theta_k$

Training data consists of pairs $L = \Big((x^1,k^1),\ldots,(x^{|L|},k^{|L|})\Big)$

$$\begin{aligned}
\ln p(L;\theta) &= \sum_l \Big[\ln p(k^l) + \ln p(x^l|k^l;\theta_{k^l})\Big] = \\
&= \sum_k n(k)\ln p(k) + \sum_k \sum_{l:k^l=k} \ln p(x^l|k;\theta_k) \rightarrow \max_{p(k),\theta_k}
\end{aligned}$$

can be optimized **independently** with respect to $\theta_a$, $\theta_1$ ... $\theta_{|K|}$

---

This was a **supervised** learning

The task:

The probability model is $p(x, k; \theta)$ as before,

training data are **incomplete**, i.e. $L = (x^1, x^2, \ldots, x^{|L|})$
– classes are not observed.

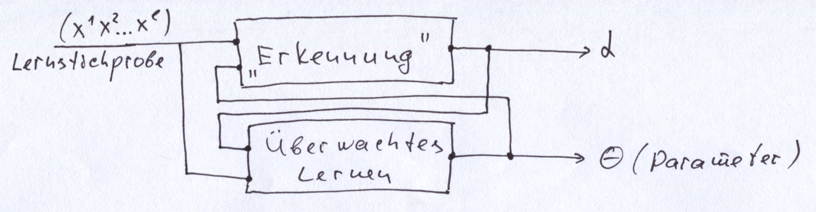
Maximum Likelihood reads:

$$\ln p(L; \theta) = \sum_l \ln p(x^l; \theta) = \sum_l \ln \sum_k p(x^l, k; \theta) \rightarrow \max_\theta$$

Problem – "$\sum \ln \sum$"

# Expectation Maximization Algorithm (idea)

An iterative approach:




1. "Recognition" (complete the data):
   $(x^1, x^2 \ldots)$, $\theta$ $\quad \Rightarrow$ "classes"
2. Supervised learning:
   "classes", $(x^1, x^2 \ldots) \Rightarrow \quad \theta$

Note: Bayesian recognition is not possible, since there is no loss-function !!!

# Expectation Maximization Algorithm (derivation)

The task:

$$\ln p(L;\theta) = \sum_l \ln p(x^l;\theta) = \sum_l \ln \sum_k p(x,k^l;\theta) \rightarrow \max_\theta$$

We introduce a “redundant $1$” and re-write it as

$$\sum_l \left[ \sum_k \alpha_l(k) \ln p(k,x^l;\theta) - \sum_k \alpha_l(k) \ln \frac{p(k,x^l;\theta)}{\sum_{k'} p(k',x^l;\theta)} \right]$$

with $\alpha_l(k) \geq 0$ and $\sum_k \alpha_l(k) = 1$ for all $l$.

With such $\alpha$-s the two above expressions are equivalent !!!

# Expectation Maximization Algorithm (derivation)

Proof of the equivalence for one example:

$$\sum_k \alpha_l(k) \ln p(k, x^l; \theta) - \sum_k \alpha_l(k) \ln \frac{p(k, x^l; \theta)}{\sum_{k'} p(k', x^l; \theta)} =$$

$$= \sum_k \Big[ \alpha_l(k) \ln p(k, x^l; \theta) -$$

$$- \Big[ \alpha_l(k) \ln p(k, x^l; \theta) - \alpha_l(k) \ln \sum_{k'} p(k', x^l; \theta) \Big] \Big] =$$

$$\sum_k \alpha_l(k) \ln \sum_{k'} p(k', x^l; \theta) = \ln \sum_{k'} p(k', x^l; \theta) \cdot \sum_k \alpha_l(k) =$$

$$= \ln \sum_{k'} p(k', x^l; \theta)$$

(for many $x^l$ just sum up)

# Expectation Maximization Algorithm

To summarize (shorthand) we have:

$$\ln p(L;\theta) = F(\theta,\alpha) - G(\theta,\alpha) \to \max_{\theta}$$

with

$$\begin{aligned} F(\theta,\alpha) &= \sum_l \sum_k \alpha_l(k) \ln p(k,x^l;\theta) \\ G(\theta,\alpha) &= \sum_l \sum_k \alpha_l(k) \ln \frac{p(k,x^l;\theta)}{\sum_{k'} p(k',x^l;\theta)} = \\ &= \sum_l \sum_k \alpha_l(k) \ln p(k|x^l;\theta) \end{aligned}$$

Note:
both $F$ and $G$ are usually concave but not their difference.

# Expectation Maximization Algorithm

$$\ln p(L;\theta) = F(\theta,\alpha) - G(\theta,\alpha) \to \max_{\theta}$$

Start with an arbitrary $\theta^{(0)}$, repeat:

1. **Expectation** step: "complete the data".
   Choose $\alpha^{(t)}$ so that $G(\theta,\alpha)$ reaches its maximum with respect to $\theta$ at the actual value $\theta^{(t)}$. Note: this is **not an optimization**, this is the estimation of an **upper bound** of $G$!!! According to the Shannon Lemma:

$$\alpha_l^{(t)}(k) = p(k|x^l;\theta^{(t)})$$

2. **Maximization** step: "supervised learning".

$$\theta^{(t+1)} = \arg\max_{\theta} F(\theta,\alpha^{(t)})$$

   Note: as $G(\theta,\alpha)$ reaches its maximum at $\theta^{(t)}$, the second addend may only decrease (the likelihood is maximized)!!!

# Some comments to the Maximum Likelihood

Maximum Likelihood estimator is not the only estimator – there are many others as well.

Maximum Likelihood is **consistent**, i.e. it gives the true parameters for infinite training sets.

---

Consider the following experiment for an estimator:

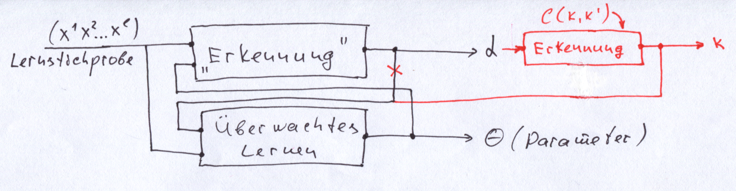
1. We generate **infinite** numbers of training sets each one being **finite**;
2. For each training set we estimate the parameter;
3. We average all estimated values.

If the average is the true parameter, the estimator is called **unbiased**. Maximum Likelihood is not always unbiased – it depends on the parameter to be estimated. Examples – the mean for a Gaussian is unbiased, the standard deviation – not.

# Some comments to the EM-Algorithm

EM always converges, but not always to the global optimum :-(

A "commonly used" technique:




The expectation step is replaced by a "real" recognition. It becomes similar to the K-Means algorithm and is often called "EM-like schema". It is **wrong**!!! It is no EM. It is an approximation of the Maximum Likelihood – the so called Saddle-Point approximation. However, it is very popular because in the practice it is often much simpler to do recognition as to compute posterior probabilities $\alpha$.