Pattern Recognition

Maximum Likelihood Principle

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. $L = (x^1, x^2, \dots, x^{|L|})$, $x^l \in \mathbb{R}^n$

One have to decide for a particular probability distribution from the given family, i.e. for a particular 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.

→ What is observed should have a high probability

→ Maximize the probability of the training data with respect to the unknown parameter

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

General discrete probability distributions

The free parameter is a "vector"

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

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 relative 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

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

Method of Lagrange coefficients:

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

$$\frac{dF}{dx_i} = \frac{a_i}{x_i} + \lambda = 0$$

$$x_i = c \cdot a_i$$

$$\sum_i c \cdot a_i - 1 = 0 \qquad\qquad x_i = \frac{a_i}{\sum_{i'} a_{i'}}$$

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

Probability densities

Example – Gaussians

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

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

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

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

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

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

“Mixed” models for recognition

$$p(x, k; \Theta) = p(k; \Theta_a) \cdot p(x|k; \Theta_k)$$

with $k \in K$ (classes, discrete) and $x \in X$ (observations, general)

Unknown parameters are $\Theta_a = p(k)$ and class-specific Θ_k

Training data consists of pairs $L = \big((x^1, k^1), (x^2, k^2), \ldots, (x^{|L|}, k^{|L|})\big)$

$$\begin{aligned} \ln p(L; \Theta) &= \sum_l \Big[\ln p(k^l) + \ln p(x^l|k^l; \Theta_{k^l})\Big] = \\ &= \sum_k n(k) \ln p(k) + \sum_k \sum_{l:k^l=k} \ln p(x^l|k; \Theta_k) \to \max_{p(k), \Theta_k} \end{aligned}$$

→ can be optimized independently with respect to Θ_a, Θ_1, … , $\Theta_{|K|}$

This was a **supervised** learning.

Unsupervised learning

Expectation-Maximization Algorithm:

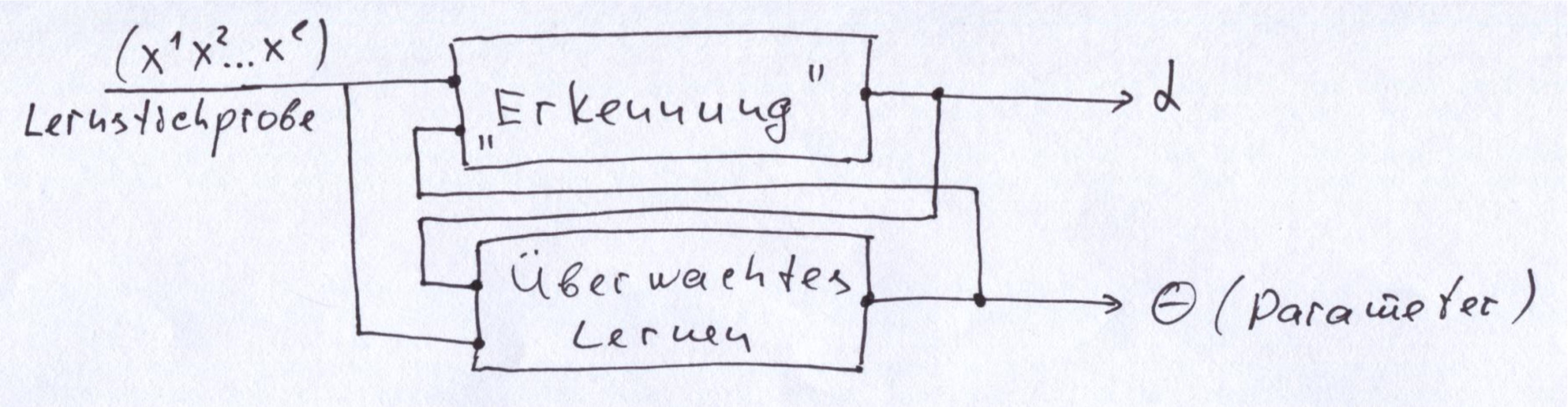
The task:

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 never observed.

Maximum-Likelihood:

$$\ln p(L;\Theta) = \sum_l \ln p(x^l;\Theta) = \sum_l \ln \sum_k p(x^l,k;\Theta) \to \max_{\Theta}$$

An iterative approach:



1. "Recognition" (complete the data): $(x^1, x^2 \dots x^l),\ \Theta \Rightarrow$ "classes"
2. Supervised learning: "classes"+ $(x^1, x^2 \dots x^l) \Rightarrow \Theta$

Note:

Bayesian recognition is impossible, since there is no loss-function !!!

EM-Algorithm (derivation)

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

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

The two above expressions are equivalent !!!

EM-Algorithm (derivation)

Proof the equivalence for just one example:

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

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

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

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

EM-Algorithm

To summarize (shorthand): we have

$$\ln p(L;\Theta) = F(\Theta,\alpha) - G(\Theta,\alpha)$$

with

$$F(\Theta,\alpha) = \sum_l \sum_k \alpha_l(k) \ln p(k, x^l; \Theta)$$

$$G(\Theta,\alpha) = \sum_l \sum_k \alpha_l(k) \ln \frac{p(k, x^l;\Theta)}{\sum_{k'} p(k', x^l; \Theta)} = \sum_l \sum_k \alpha_l(k) \ln p(k|x^l;\Theta)$$

EM-Algorithm

$$\ln p(L;\Theta) = F(\Theta,\alpha) - G(\Theta,\alpha)$$

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 Θ 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: since $G(\Theta,\alpha)$ reaches its maximum at $\Theta^{(t)}$, the second addend may only decrease → the likelihood is maximized!!!

Some comments to 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.

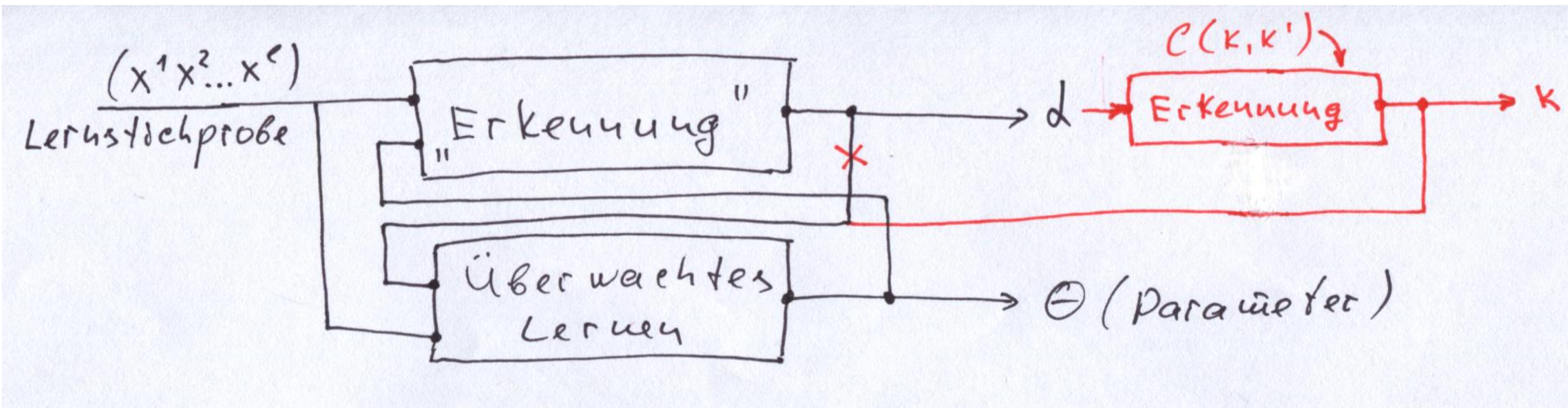
If the average is the true parameter, the estimator is called **unbiased**.

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

Some comments to Expectation-Maximization

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 simpler to do recognition as to compute posterior probabilities α .