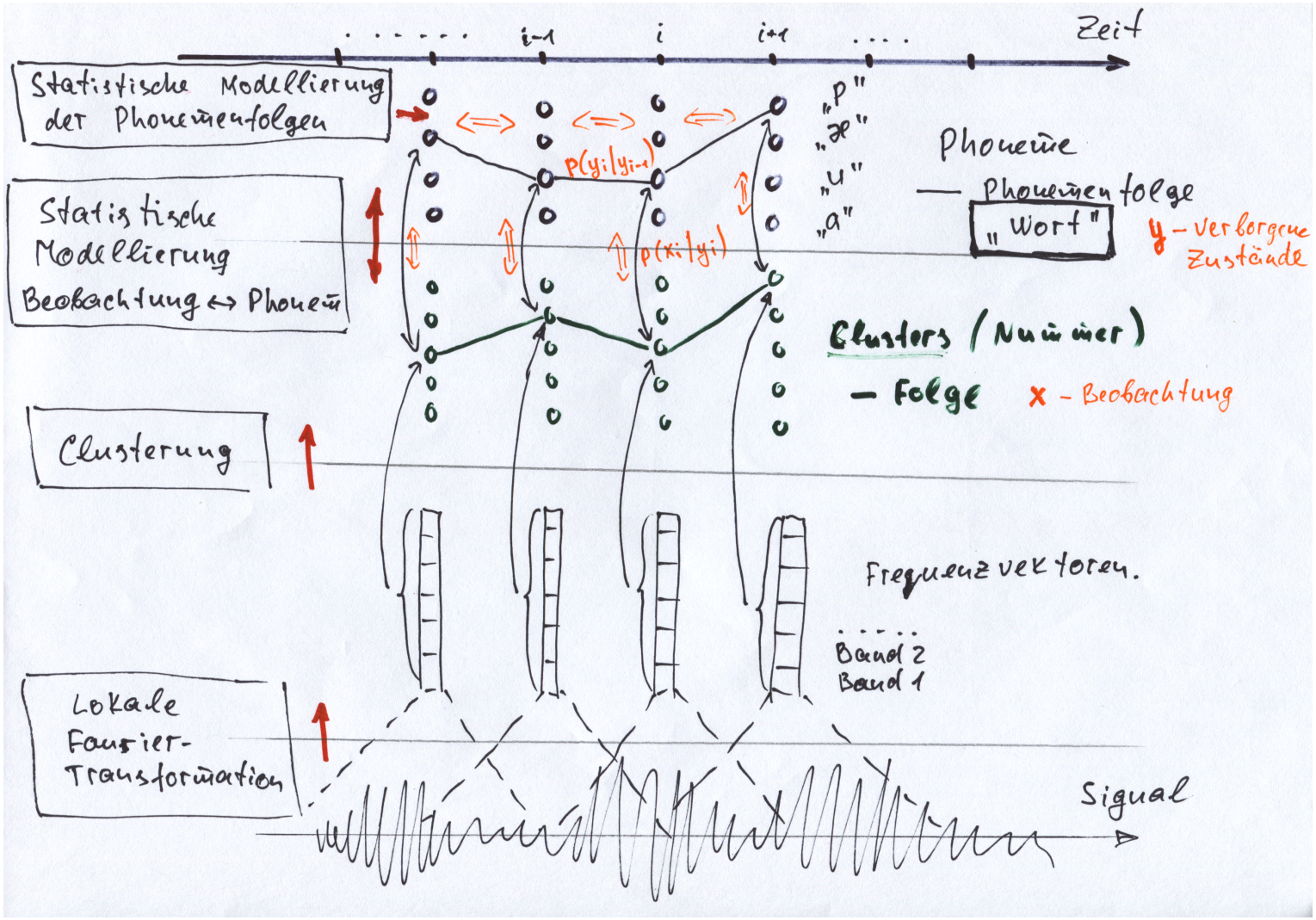
Data analysis: Statistical principals and computational methods

Markov Chains

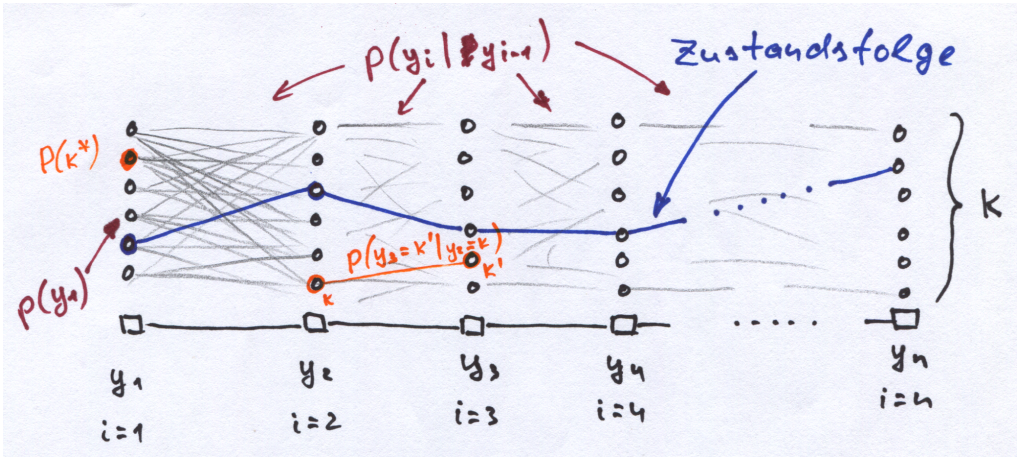
Dmitrij Schlesinger, Carsten Rother

SS2014, 04.06.2014

Application: Speech Recognition



Markov Chains – the probabilistic model



Let $I = \{1, 2, \ldots, n\}$ be the set of "time points"

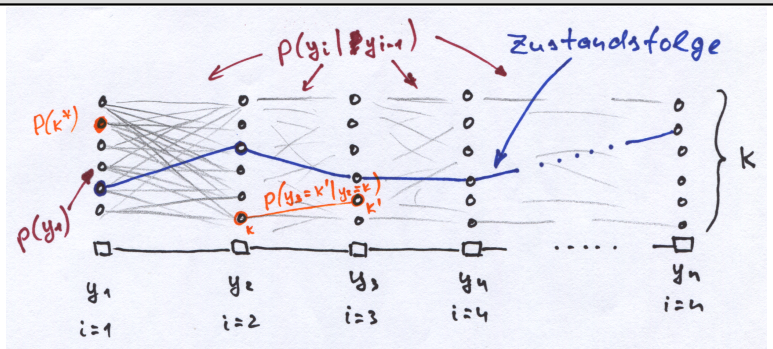
There is a **random variable** $y_i \in K$ for each $i \in I$, where K is a discrete domain (state set, label set, phoneme set etc.).

$y = (y_1, y_2, \ldots, y_n)$ with $y_i \in K$ is a **state sequence**

The set of all such sequences is $\mathcal{Y} = K^n$
– the sample space (the set of elementary events)

$p(y) = p(y_1, y_2, \ldots, y_n)$ are their probabilities

Markov Chains – the probabilistic model



In general: $p(y_1 \dots y_n) = p(y_1 \dots y_{n-1}) \cdot p(y_n | y_1 \dots y_{n-1})$

The **Markovian property** (simplified):

$$p(y_n | y_1 \dots y_{n-1}) = p(y_n | y_{n-1})$$

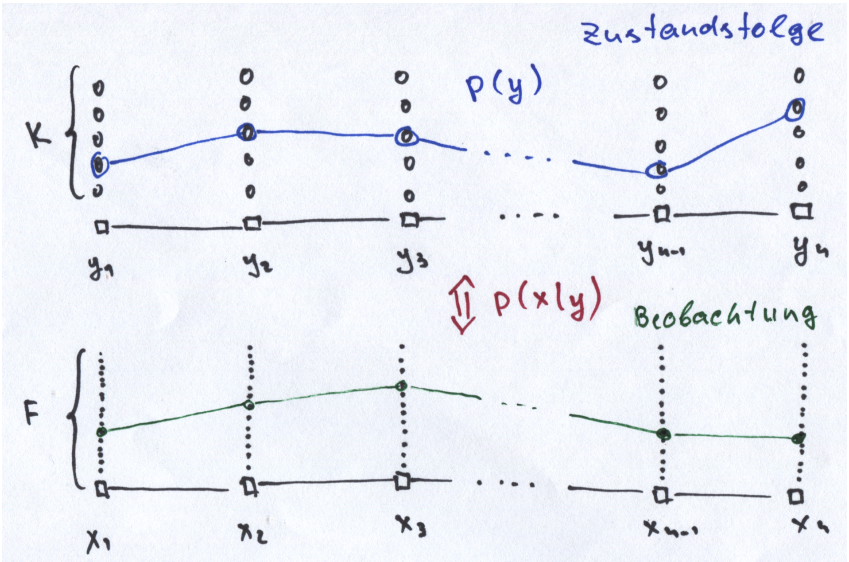
It follows:

$$p(y) = p(y_1) \cdot \prod_{i=2}^{n} p(y_i | y_{i-1})$$

The model parameters are:

- the starting probability distribution $p(y_1)$
- the transition probability distributions $p(y_i | y_{i-1})$

Hidden Markov Chains (HMM)



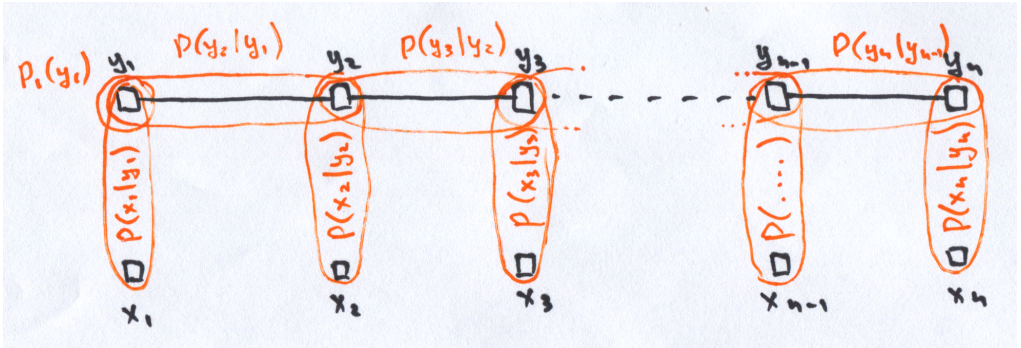
There are two "kinds" of Variables (both are sequences):

$y = (y_1, y_2 \dots y_n)$, $y_i \in K$ (hidden Variables) and

$x = (x_1, x_2 \dots x_n)$, $x_i \in F$ (observed Variables).

A pair (x, y) is an elementary event.

Hidden Markov Chains



$$p(x,y) = p(y) \cdot p(x|y) = p(y_1) \prod_{i=2}^{n} p(y_i|y_{i-1}) \prod_{i=1}^{n} p(x_i|y_i)$$

The conditional p.d. $p(x|y)$ is **conditionally independent**.
Do not confuse independence and conditional independence.
For instance, some x_i and x_j are conditionally independent given an arbitrary y, but they are of course dependent through the hidden part, i.e. $p(x_i, x_j) \neq p(x_i) \cdot p(x_j)$.

Note: summation over x gives 1 for any y:

$$\sum_{y} p(x|y) = \sum_{y} \prod_{i} p(x_i|y_i) = \prod_{i} \sum_{k} p(k|y_i) = 1$$

The probability of observation

$$p(x) = \sum_y p(x, y) =$$

$$= \sum_y \left[p(y_1) \prod_{i=2}^{n} p(y_i|y_{i-1}) \prod_{i=1}^{n} p(x_i|y_i) \right]$$

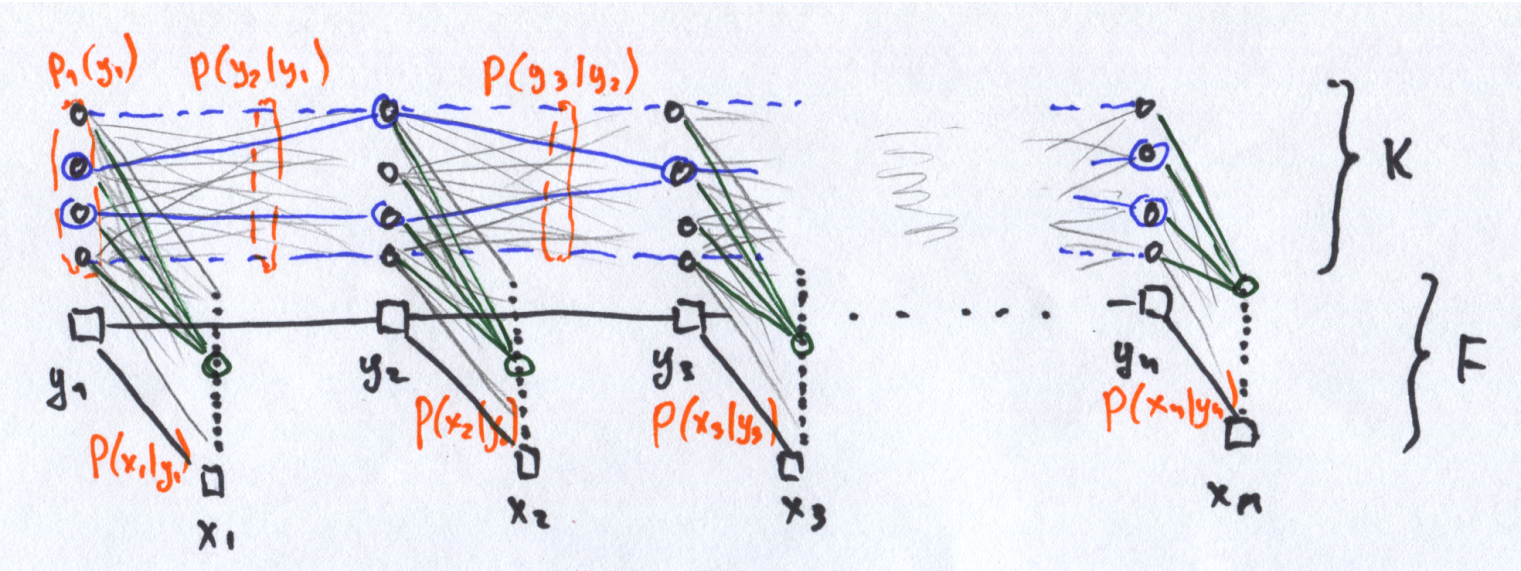
– so-called **SumProd** problem.

For instance, it can be used to recognize the model:

let two Markov chains be given (specified by their parameters $p(y_1)$, $p(y_i|y_{i-1})$ and $p'(y_1)$, $p'(y_i|y_{i-1})$ respectively).

For a particular observation x decide, which model has generated it.

The probability of observation



$$p(x) = \sum_y p(x, y) =$$

$$= \sum_y \left[p(y_1) \prod_{i=2}^{n} p(y_i|y_{i-1}) \prod_{i=1}^{n} p(x_i|y_i) \right]$$

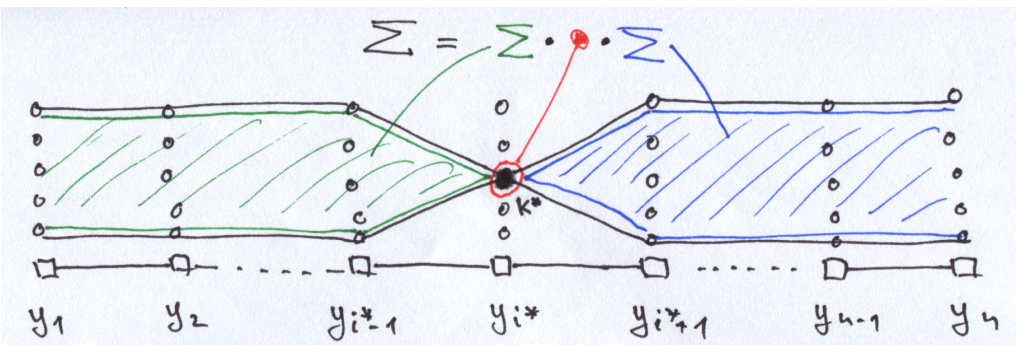
Prior marginal probabilities

Suppose, we are interested in the (prior) probability that at a given position i^* the model generates a particular label k^*.

Note: this is a composite event whose probability is a sum over all corresponding elementary ones:

$$p(y_{i^*}{=}k^*) = \sum_{y:y_{i^*}=k^*} p(y) = \sum_{y:y_{i^*}=k^*} \left[p(y_1) \prod_{i=2}^{n} p(y_i|y_{i-1}) \right]$$

– a SumProd problem again.



Posterior marginal probabilities

For a given observation x calculate the probability that at a given position i^* the model generates a particular label k^*:

$$p(y_{i^*}{=}k^*|x) \propto p(x, y_{i^*}{=}k^*) = \sum_{y:y_{i^*}=k^*} p(x,y) =$$

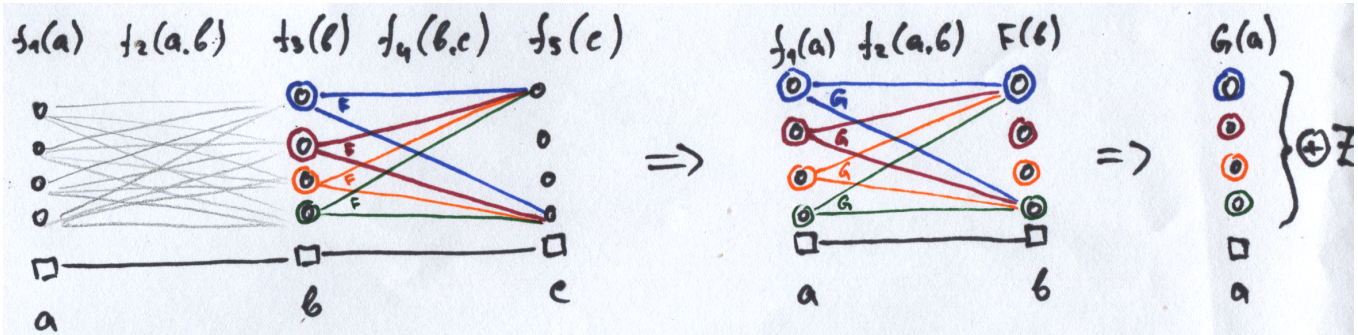
$$= \sum_{y:y_{i^*}=k^*} \left[p(y_1) \prod_{i=2}^{n} p(y_i|y_{i-1}) \prod_{i=1}^{n} p(x_i|y_i) \right]$$

In all cases (+ the Partition Function) the occurring SumProd problem can be expressed as

$$Z = \sum_y \left[\prod_{i=1}^{n} \psi_i(y_i) \cdot \prod_{i=2}^{n} \psi_{i-1,i}(y_{i-1}, y_i) \right]$$

SumProd algorithm

Example: $f(a,b,c) = f_1(a) \cdot f_2(a,b) \cdot f_3(b) \cdot f_4(b,c) \cdot f_5(c)$



$$Z = \sum_{abc} f(a,b,c) =$$

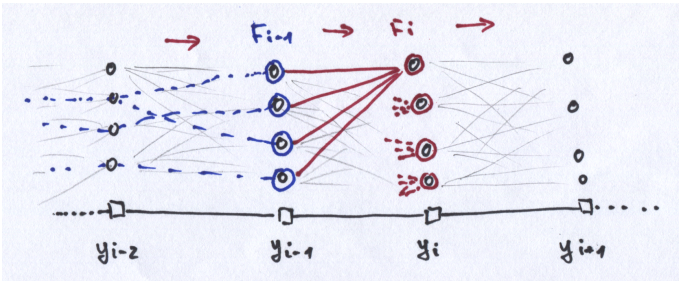
$$\sum_a \sum_b \sum_c \Big[f_1(a) f_2(a,b) f_3(b) f_4(b,c) f_5(c) \Big] =$$

$$= \sum_a f_1(a) \cdot \Big[\sum_b f_2(a,b) f_3(b) \cdot \Big[\sum_c f_4(b,c) f_5(c) \Big] \Big] =$$

$$= \sum_a f_1(a) \cdot \Big[\sum_b f_2(a,b) F(b) \Big] = \sum_a G(a)$$

SumProd algorithm

The Idea: propagate **Bellmann functions** F_i (aka messages) that represent partial solutions (sums)



for ($k = 1 \dots K$) $F_1(k) = q_1(k)$
for ($i = 2 \dots n$)
  for ($k = 1 \dots K$)
    $F_i(k) = 0$
    for ($k' = 1 \dots K$)
      $F_i(k) = F_i(k) + F_{i-1}(k')g_i(k', k)$
    $F_i(k) = F_i(k) \cdot q_i(k)$
$Z = \sum_k F_n(k)$

Time complexity – $O(nK^2)$

Inference

We have a model for $p(x, y)$

We observe x

Find the "corresponding" y in a reasonable way

Most probable state sequence

$$p(x,y) = p(y_1)\prod_{i=2}^{n} p(y_i|y_{i-1})\prod_{i=1}^{n} p(x_i|y_i)$$

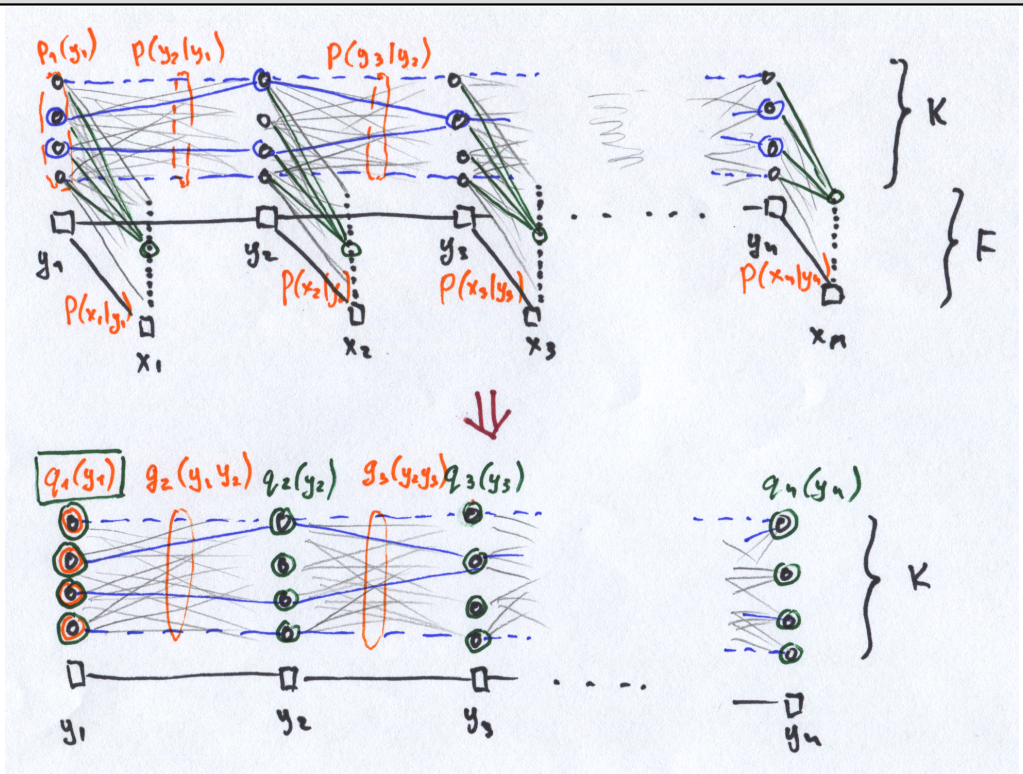
Given an x estimate y. A "reasonable" approach – the most (a-posteriori) probable state sequence:

$$\begin{aligned}
\arg\max_y p(y|x) &= \arg\max_y \frac{p(x,y)}{p(x)} = \arg\max_y p(x,y) = \\
&= \arg\max_y \left[p(y_1)\prod_{i=2}^{n} p(y_i|y_{i-1})\prod_{i=1}^{n} p(x_i|y_i) \right] = \\
\text{/ take logarithm} &= \arg\min_y \left[\sum_{i=1}^{n} \psi_i(y_i) + \sum_{i=2}^{n} \psi_{i-1,i}(y_{i-1}, y_i) \right]
\end{aligned}$$

with

$$\psi_i(y_i) = \ln p(x_i|y_i)\Big(+\ln p(y_1)\Big) \quad \psi_{i-1,i}(y_{i-1}, y_i) = \ln p(y_i|y_{i-1})$$

Most probable state sequence



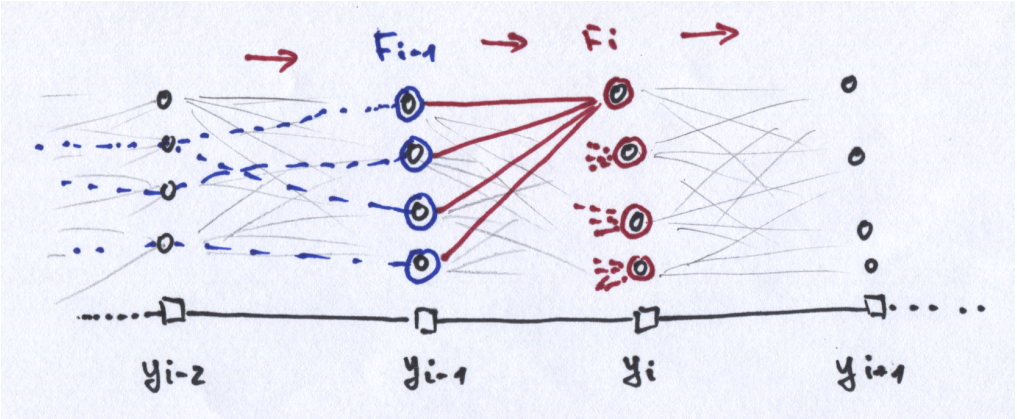
In contrast to SumProd we have to **minimize** over all state sequences rather than to **sum** them up

The "quality" of a sequence is not a **product** but a **sum**

The same as SumProd but in another Semiring:

$$\text{SumProd} \Leftrightarrow \text{MinSum}$$

Dynamic Programming (Vitterbi, Dijkstra ...)



The Idea – propagate Bellman Functions F_i by

$$F_i(k) = \psi_i(k) + \min_{k'} \left[F_{i-1}(k') + \psi_{i-1,i}(k', k) \right]$$

The functions F_i represent the quality of the the best extension into the already processed part

Dynamic Programming (derivation)

$$\min_{y_1}\min_{y_2}\ldots\min_{y_n}\left[\sum_{i=1}^{n}\psi_i(y_i)+\sum_{i=2}^{n}\psi_{i-1,i}(y_{i-1},y_i)\right]=$$

$$\min_{y_2}\ldots\min_{y_n}\left[\sum_{i=2}^{n}\psi_i(y_i)+\sum_{i=3}^{n}\psi_{i-1,i}(y_{i-1},y_i)+\right.$$

$$\left.+\min_{y_1}\Big(\psi_1(y_1)+\psi_{1,2}(y_1,y_2)\Big)\right]=$$

$$\min_{y_2}\ldots\min_{y_n}\left[\sum_{i=2}^{n}\psi_i(y_i)+\sum_{i=3}^{n}\psi_{i-1,i}(y_{i-1},y_i)+F(y_2)\right]=$$

$$\min_{y_2}\ldots\min_{y_n}\left[\sum_{i=2}^{n}\tilde{\psi}_i(y_i)+\sum_{i=3}^{n}\psi_{i-1,i}(y_{i-1},y_i)\right]$$

with $\tilde{\psi}_2(k)=\psi_2(k)+F(k)$, $\tilde{\psi}_i(k)=\psi_i(k)$ for $i=3\ldots n$.

Dynamic Programming (the algorithm)

// Forward pass

for $i = 2$ to n

  for $k = 1$ to K

    $best = \infty$

    for $k' = 1$ to K

      if $\psi_{i-1}(k') + \psi_{i-1,i}(k', k) < best$

        $best = \psi_{i-1}(k') + \psi_{i-1,i}(k', k)$, $pointer_i(k) = k'$

    $\psi_i(k) = \psi_i(k) + best$

// Backward pass

$best = \infty$

for $k = 1$ to K

  if $\psi_n(k) < best$

    $best = \psi_n(k)$, $x_n = k$

for $i = n - 1$ to 1

  $x_i = pointer_{i+1}(x_{i+1})$

$pointer_i(k)$ is the best predecessor for k

time complexity: $O(nK^2)$ (due to the forward pass)

Statistical learning

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

Let the training data be given, e.g. $L = (y^1, y^2, \ldots, y^{|L|})$

One have to decide for a particular probability distribution from the given family, i.e. for the "best" parameter θ^*

Maximum Likelihood for Markov Chains

The model (not hidden for simplicity):

$$p(y;\theta) = p(y_1) \cdot \prod_{i=2}^{n} p(y_i|y_{i-1})$$

The parameters to be learned: $\psi = (\psi_1, \psi_{i-1,i})$ i.e.

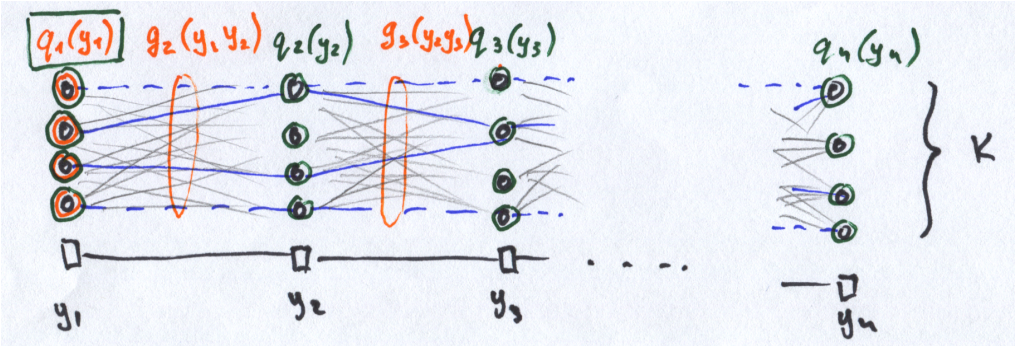
$$p(y;\psi) = \psi_1(y_1) \cdot \prod_{i=2}^{n} \psi_{i-1,i}(y_{i-1}, y_i)$$

Note: the parameter set is restricted, i.e.

$$\psi_\cdot \geq 0, \quad \sum_k \psi_1(k) = 1, \quad \sum_{k'} \psi_{i-1,i}(k,k') = 1 \quad \forall i,k$$

Training set: $L = (y^1, y^2 \dots y^l)$ (sequences)

Maximum Likelihood for Markov Chains



The Likelihood:

$$\ln p(L, \theta) = \sum_l \ln p(y^l; \theta) =$$

$$= \sum_l \left[\ln \psi_1(y_1^l) + \sum_{i=2}^{n} \ln \psi_{i-1,i}(y_{i-1}^l, y_i^l) \right] =$$

$$= \sum_l \ln \psi_1(y_1^l) + \sum_{i=2}^{n} \sum_l \ln \psi_{i-1,i}(y_{i-1}^l, y_i^l) \to \max_{\psi}$$

Addends can be optimized independently !!!

Maximum Likelihood for Markov Chains

Consider e.g. the first addend:

$$\sum_l \ln \psi_1(y_1^l) = \sum_k \sum_{l:y_1^l=k} \ln \psi_1(k) = \sum_k n_1(k) \cdot \ln \psi_1(k) \to \max_{\psi_1}$$

$$\text{s.t. } \psi_1(k) \geq 0, \quad \sum_k \psi_1(k) = 1$$

where $n_1(k)$ are the (absolute) state frequencies (in the training set) for the first time point.

The optimum is reached at $\psi_1(k) \sim n_1(k)$, i.e.

$$\psi_1(k) = \frac{n_1(k)}{\sum_{k'} n_1(k')}$$

– relative state frequencies.

Maximum Likelihood for Markov Chains

Almost analogous for transition matrices $\psi_{i-1,i}(k, k')$:

$$\sum_l \ln \psi_{i-1,i}(y_{i-1}^l, y_i^l) = \sum_k \sum_{k'} \sum_{l: y_{i-1}=k, y_i^l=k'} \ln \psi_{i-1,i}(k, k') =$$

$$\sum_k \sum_{k'} n_{i-1,i}(k, k') \cdot \ln \psi_{i-1,i}(k, k') \to \max_{\psi_{i-1,i}}$$

again, it can be done for each k independently

$$\sum_{k'} n_{i-1,i}(k, k') \cdot \ln \psi_{i-1,i}(k, k') \to \max_{\psi_{i-1,i}} \quad \forall k$$

$$\Rightarrow \psi_{i-1,i}(k, k') = \frac{n_{i-1,i}(k, k')}{\sum_{k''} n_{i-1,i}(k, k'')}$$

Conclusion: "Set the parameters to the statistics computed from the training set"

Unsupervised Learning

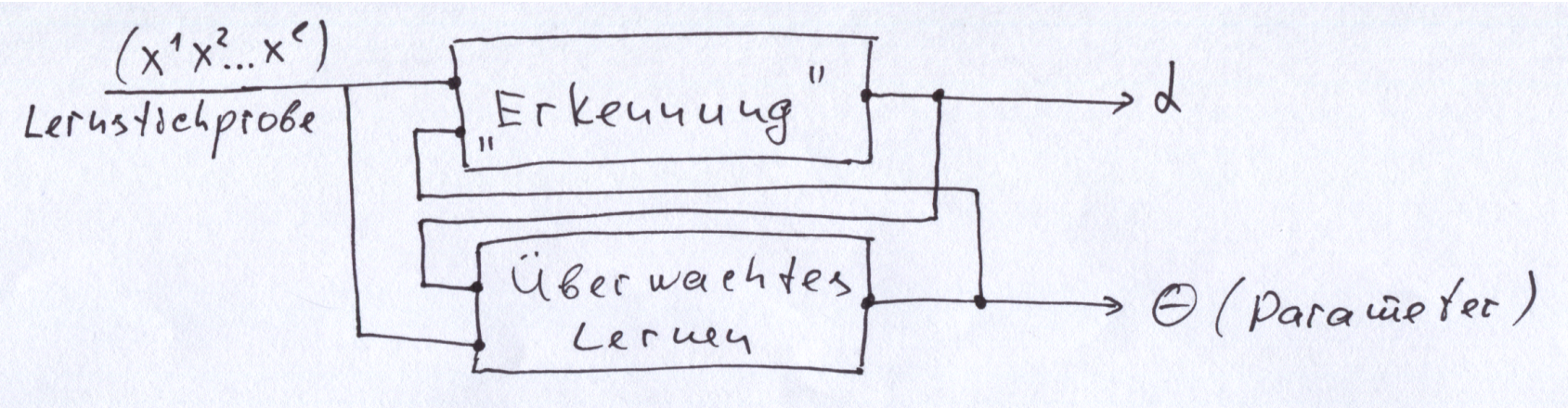
The probability model is $p(x, y; \psi)$ (HMM)

training data are **incomplete**, i.e. $L = (x^1, x^2, \ldots, x^{|L|})$
– hidden part is not observed – samples are **composite** events

Maximum Likelihood reads:

$$\ln p(L; \psi) = \sum_l \ln p(x^l; \psi) = \sum_l \ln \sum_y p(x^l, y; \psi) \to \max_\psi$$

Expectation Maximization Algorithm (idea)



Iterate

1. "Recognition" (complete the data):
   $(x^1, x^2 \ldots), \psi \quad \Rightarrow$ "hidden parts"

2. Supervised learning:
   "hidden parts", $(x^1, x^2 \ldots) \Rightarrow \quad \psi$

Expectation Maximization for Markov Chains

The model is a HMM:

$$p(x, y; \theta) = p(y_1) \cdot \prod_{i=2}^{n} p(y_i|y_{i-1}) \cdot \prod_{i=1}^{n} p(x_i|y_i)$$

Unknown parameters are: $\psi = (\psi_1, \psi_{i-1,i}, \psi_i)$ i.e.

$$p(x, y; \psi) = \psi_1(y_1) \cdot \prod_{i=2}^{n} \psi_{i-1,i}(y_{i-1}, y_i) \cdot \prod_{i=1}^{n} \psi_i(x_i, y_i)$$

Training set $L = (x^1, x^2 \dots x^l)$ sequences of observation

Likelihood:

$$\ln p(L, \psi) = \sum_l \ln p(x^l; \psi) = \sum_l \ln \sum_y p(x^l, y; \psi) \to \max_\psi$$

Expectation Maximization for Markov Chains

Introduce $\alpha_l(y) \geq 0$ s.t. $\sum_y \alpha_l(y) = 1$ for all l.

Start from an arbitrary $\psi^{(0)}$
Iterate

1. In the **expectation** step we set

$$\alpha_l^{(t)}(y) = p(y|x^l; \psi^{(t)})$$

   Note: technically, it is not possible, since the number of α-s is huge – an "auxiliary construct"

2. In the **maximization** step we have to optimize

$$\psi^{(t+1)} = \arg\max_{\psi} \sum_l \sum_y \alpha_l^{(t)}(y) \cdot \ln p(x^l, y; \psi)$$

Expectation Maximization for Markov Chains

$$\sum_l \sum_y \alpha_l(y) \ln p(x^l, y; \psi) =$$

$$= \sum_l \sum_y \alpha_l(y) \cdot \left[\ln \psi_1(y_1) + \sum_{i=2}^{n} \ln \psi_{i-1,i}(y_{i-1}, y_i) + \right.$$

$$\left. + \sum_{i=1}^{n} \ln \psi_i(x_i^l, y_i) \right]$$

again, addends can be optimized independently.

Let us consider only the first addend for simplicity

$$\sum_l \sum_y \alpha_l(y) \cdot \ln \psi_1(y_1) =$$

$$\sum_l \sum_k \sum_{y: y_1 = k} \alpha_l(y) \cdot \ln \psi_1(k) = \sum_k \ln \psi_1(k) \cdot \left[\sum_l \sum_{y: y_1 = k} \alpha_l(y) \right]$$

Expectation Maximization for Markov Chains

Remember: $\alpha_l^{(t)}(y) = p(y|x^l;\psi^{(t)})$ (expectation step)

$$\sum_k \ln \psi_1(k) \cdot \left[\sum_l \sum_{y:y_1=k} \alpha_l(y)\right] =$$

$$= \sum_k \ln \psi_1(k) \cdot \left[\sum_l \sum_{y:y_1=k} p(y|x^l;\psi^{(t)})\right] =$$

$$= \sum_k \ln \psi_1(k) \cdot \left[\sum_l p(y_1{=}k|x^l;\psi^{(t)})\right] = \sum_k \ln \psi_1(k) \cdot n_1(k)$$

$\rightarrow$ the optimal ψ_1 is $\psi_1(k) \propto n_1(k)$

It is not necessary to compute $\alpha_l(y)$ for each y individually, for the maximization step only marginals are necessary !!!

Expectation Maximization for Markov Chains

Start with arbitrary $\psi^{(0)}$, repeat

1. **Expectation**: compute

$$n_1(k) = \sum_l p(y_1{=}k|x^l;\psi^{(t)})$$
$$n_{i-1,i}(k,k') = \sum_l p(y_{i-1}{=}k, y_i{=}k'|x^l;\psi^{(t)})$$

   using the SumProd Algorithm.

2. **Maximization**: set (properly normalized)

$$\psi_1^{(t+1)}(k) \propto n_1(k)$$
$$\psi_{i-1,i}^{(t+1)}(k,k') \propto n_{i-1,i}(k,k')$$

Summary

Today: Markov chains

- The probabilistic model
- Some "useful" probabilities
- SumProd algorithm
- Inference – MAP, Dynamic Programming
- Statistical learning, Maximum Likelihood principle
- Unsupervised learning, Expectation-Maximization algorithm

The next lecture: Energy Minimization

- The problem
- The binary case, binary submodular problems, MinCut
- Search techniques: Iterated Conditional Modes, α-expansion, $\alpha\beta$-swap