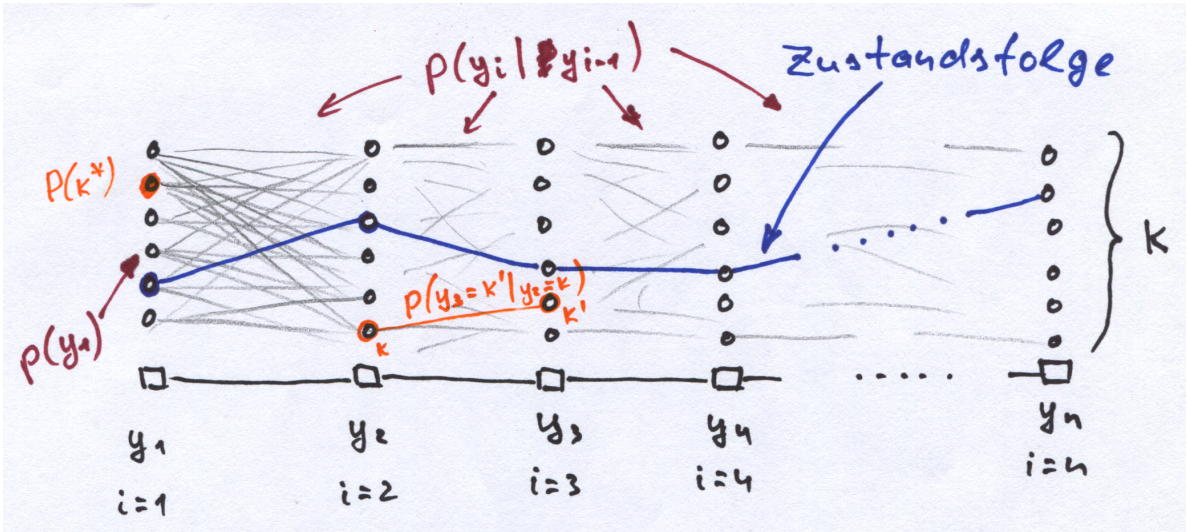
Data analysis: Statistical principals and computational methods

Summary

Dmitrij Schlesinger, Carsten Rother

SS2014, 16.07.2014

I. Markov Chains – the probabilistic model

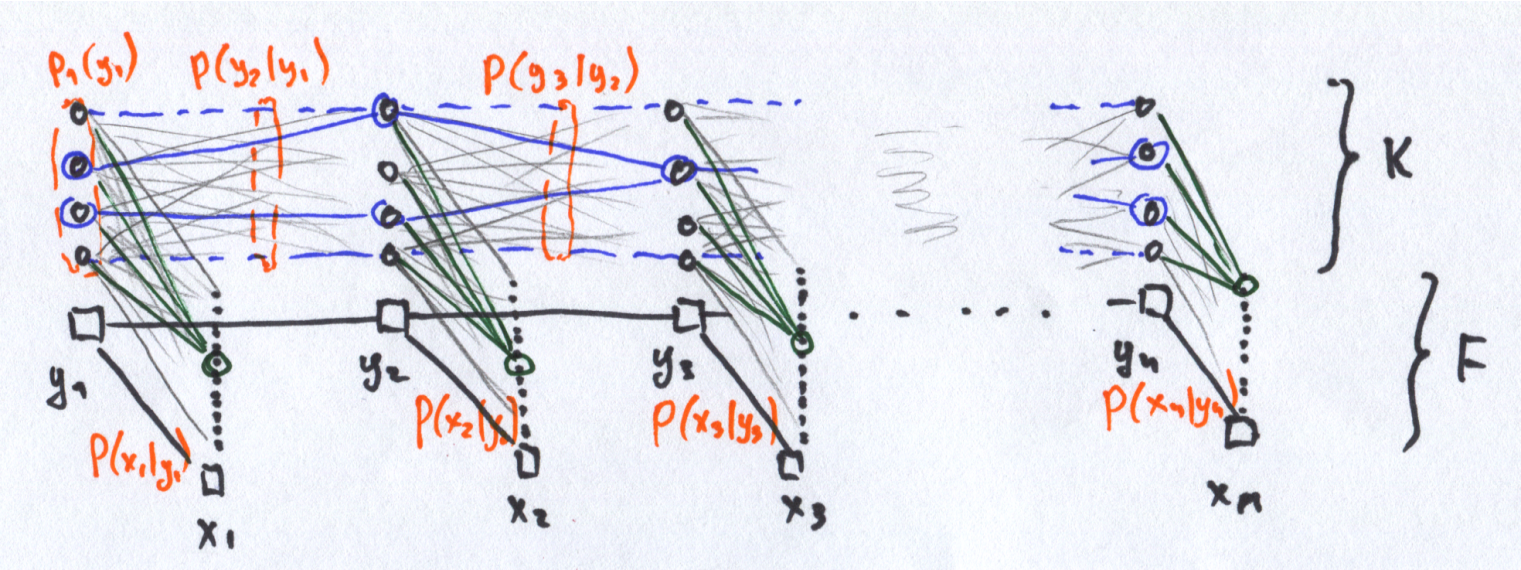


Random variables $y_i \in K$ for each $i \in I$
state sequence $y = (y_1, y_2, \ldots, y_n)$ with $y_i \in K$

$$p(y) = p(y_1, y_2, \ldots, y_n) = p(y_i) \prod_{i=2}^{n} p(y_i|y_{i-1})$$

HMM: $p(x, y) = p(y) \cdot p(x|y)$ ...

I. Markov Chains – the probability of observation



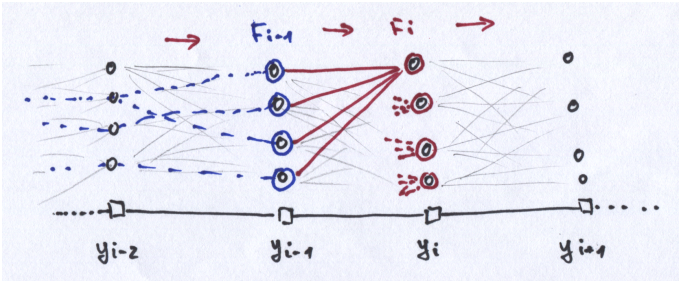
$$p(x) = \sum_y p(x, y) =$$

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

$+$ other marginal probabilities ...

I. Markov Chains – 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 ($k' = 1 \dots K$)

      $F_i(k) = F_i(k) + F_{i-1}(k')g_i(k', k)$

    $F_i(k) = F_i(k) \cdot q_i(k)$

$Z = \sum_k F_n(k)$

I. Most probable state sequence

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

$$\Downarrow$$

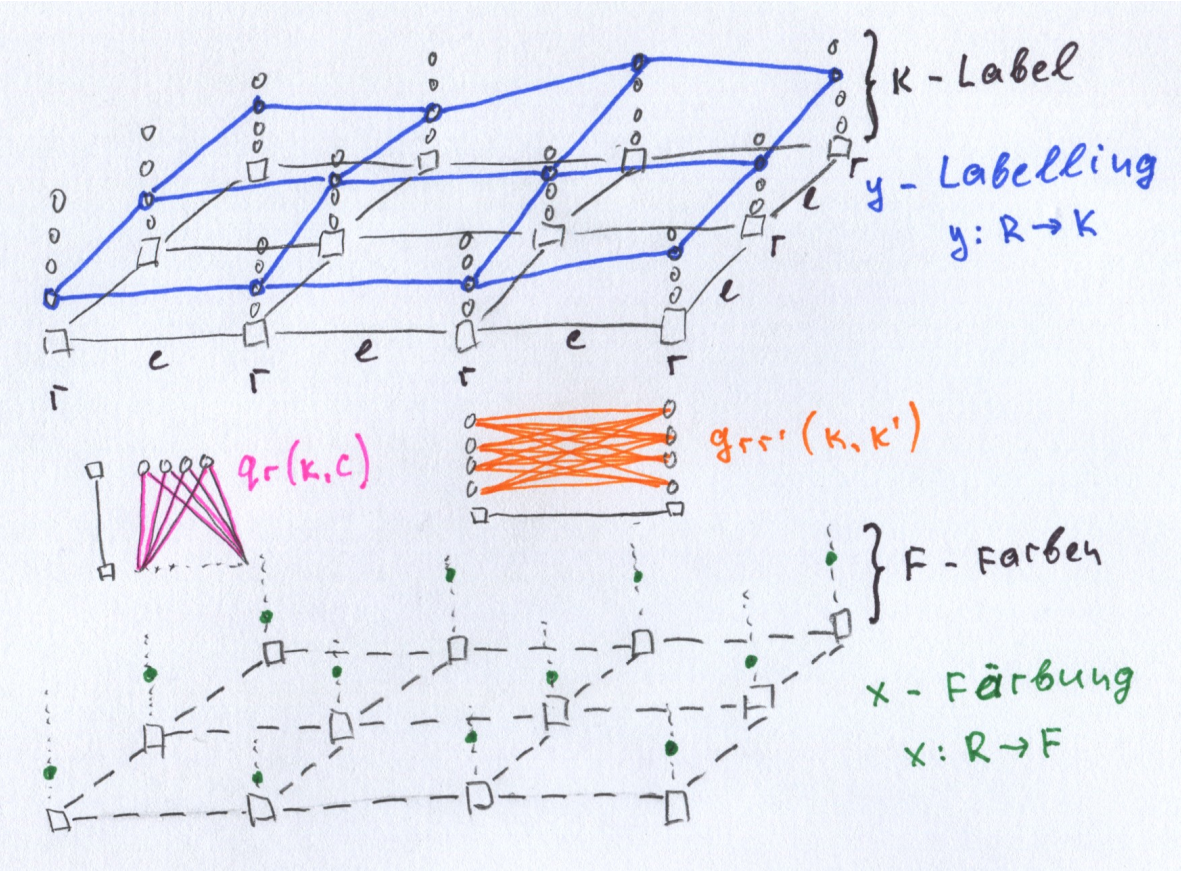
$$\arg\min_y \left[\sum_{i=1}^{n} \psi_i(y_i) + \sum_{i=2}^{n} \psi_{i-1,i}(y_{i-1}, y_i) \right]$$

Dynamic Programming (Vitterbi, Dijkstra ...) – propagate Bellman Functions F_i by

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

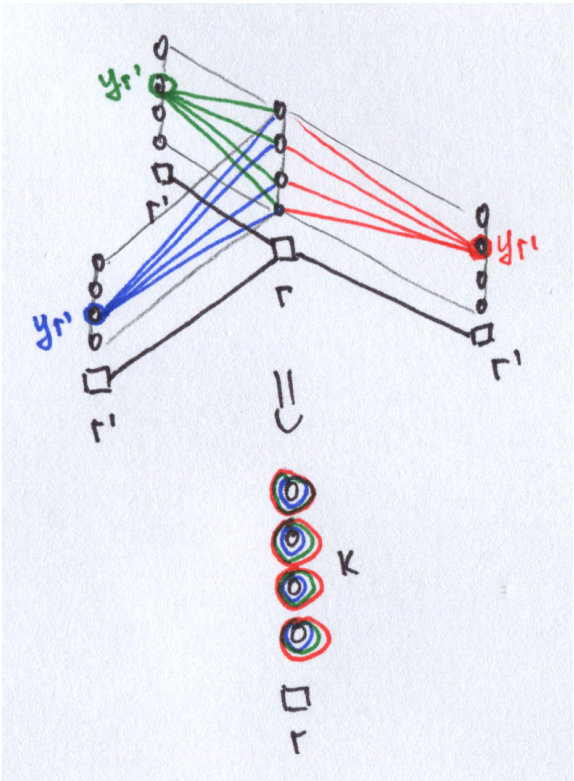
The functions F_i represent the quality of the the best extension into the already processed part

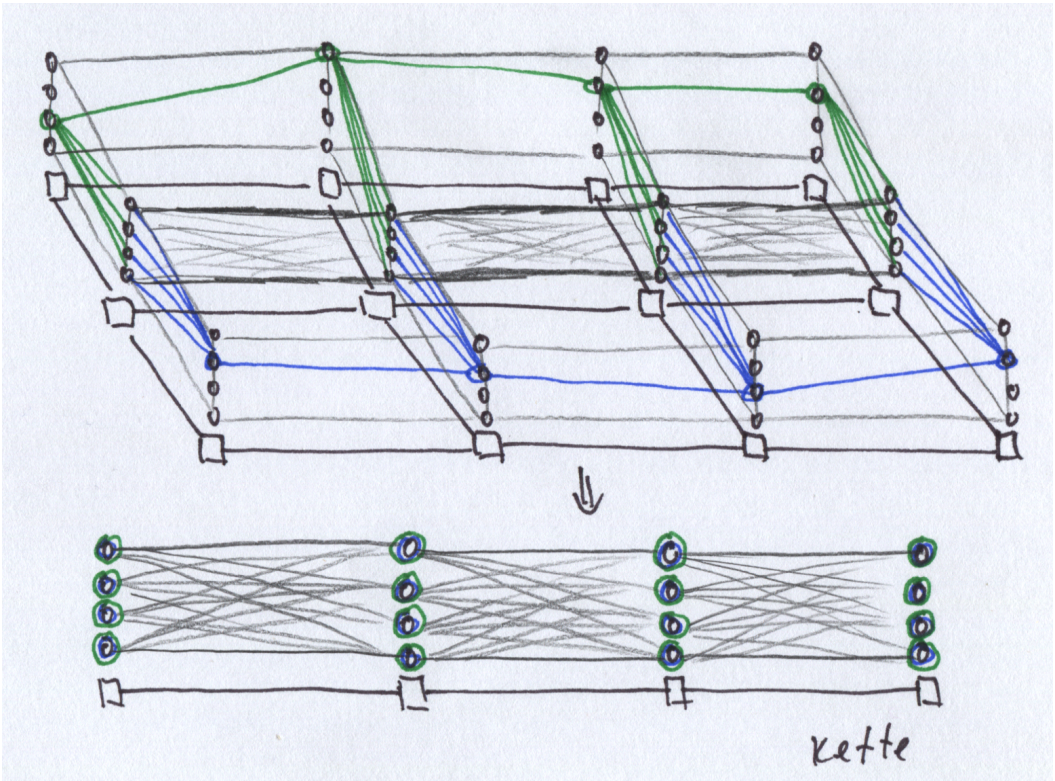
II. Energy Minimization



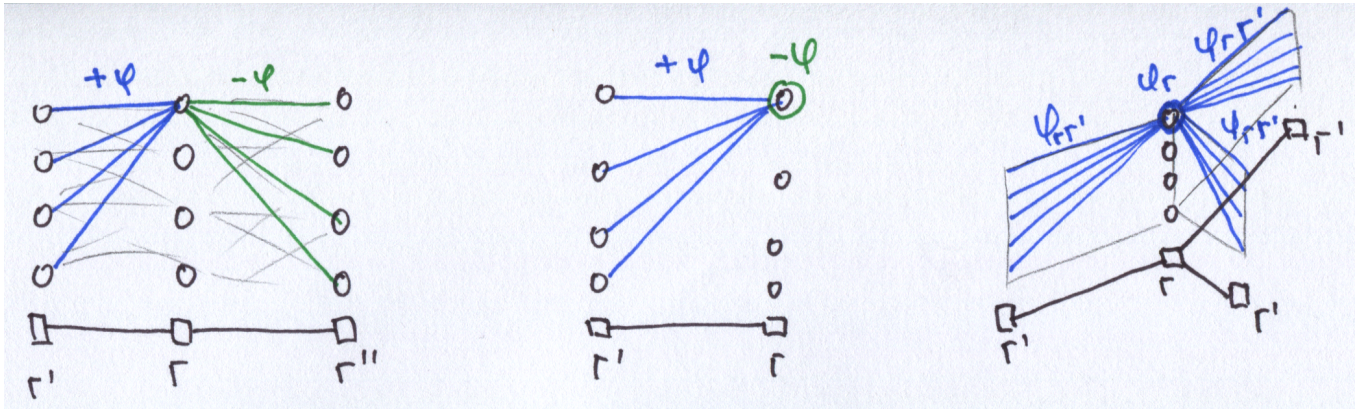
$$y^* = \arg\min_y \left[\sum_i \psi_i(y_i) + \sum_{ij} \psi_{ij}(y_i, y_j) \right]$$

II. Iterated Conditional Modes



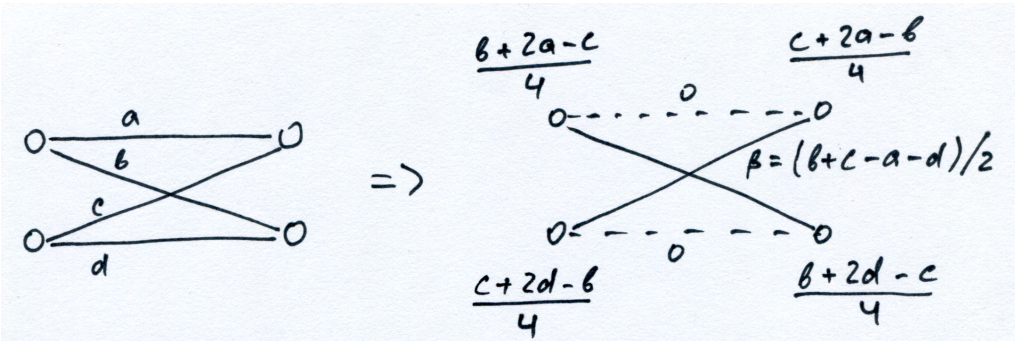


II. Equivalent transformations

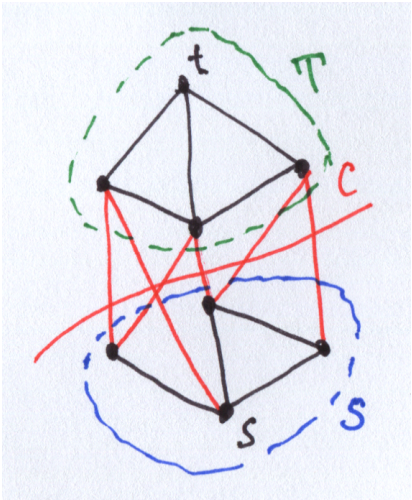


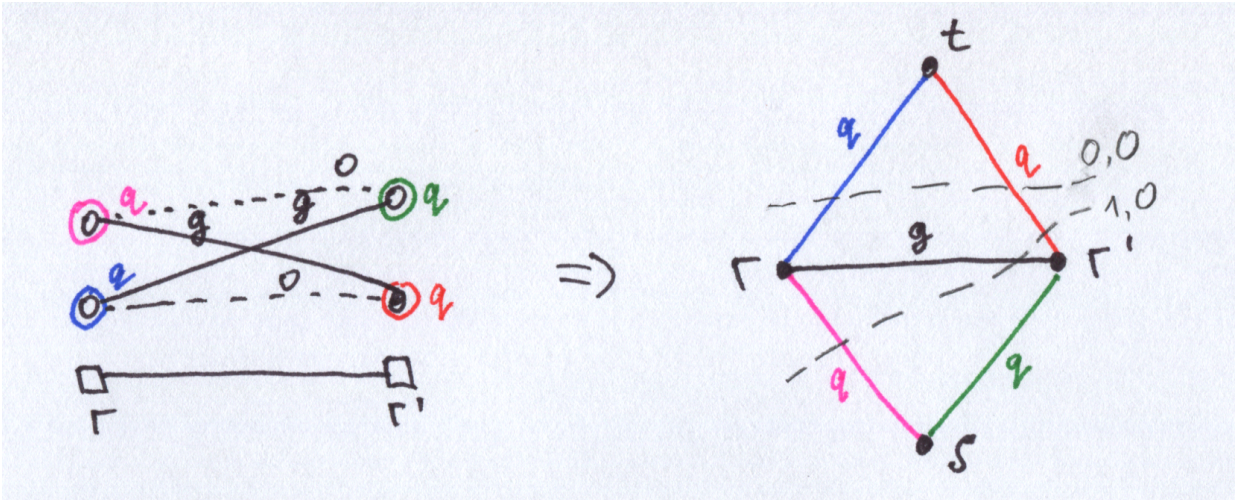
Binary MinSum Problems – canonical forms

$$E(y) = (\ldots) + \sum_{rr'} \beta_{ij} \cdot \delta(y_i \neq y_j)$$



II. Binary MinSum Problems $\leftrightarrow$ MinCut



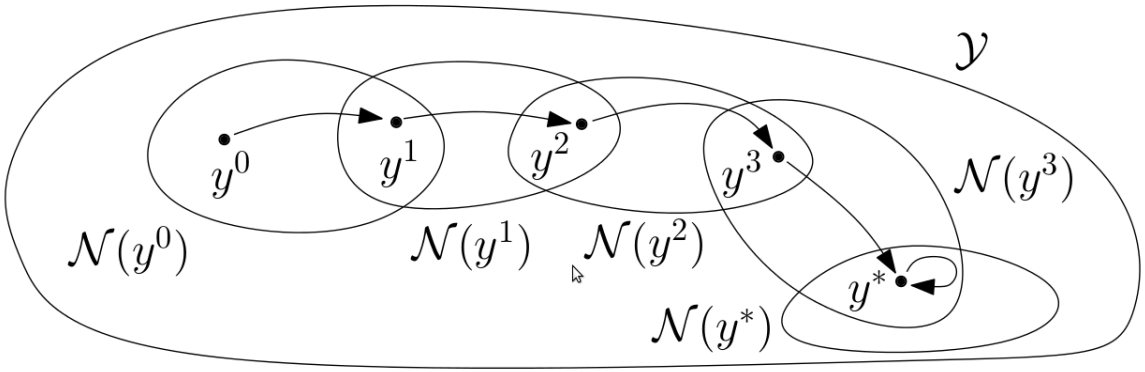


$$C^* = \arg\min_C \sum_{ij \in C} c_{ij}$$

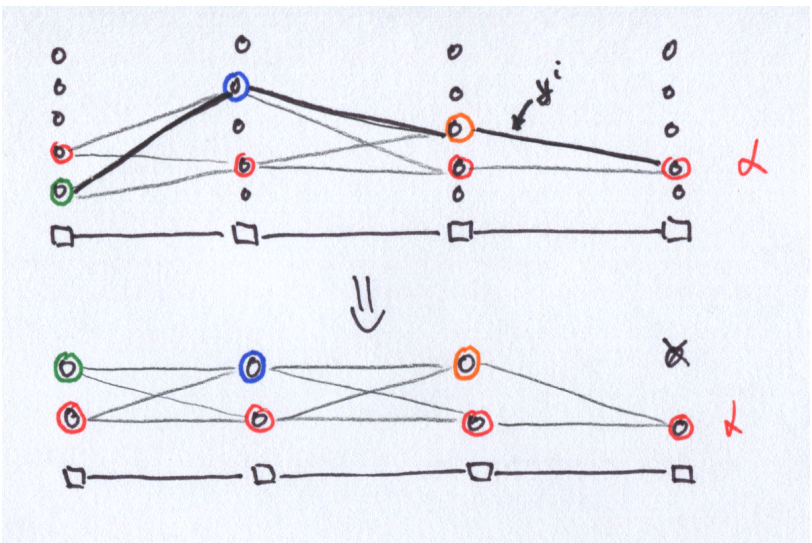
- The relation MinSum $\leftrightarrow$ MinCut works **always**
- MinCut is NP-complete in general
- MinCut is **polynomially** solvable if all edge costs are **non-negative**, i.e. $a + d \geq b + c$ holds for all edges
- Such problems are called **submodular**

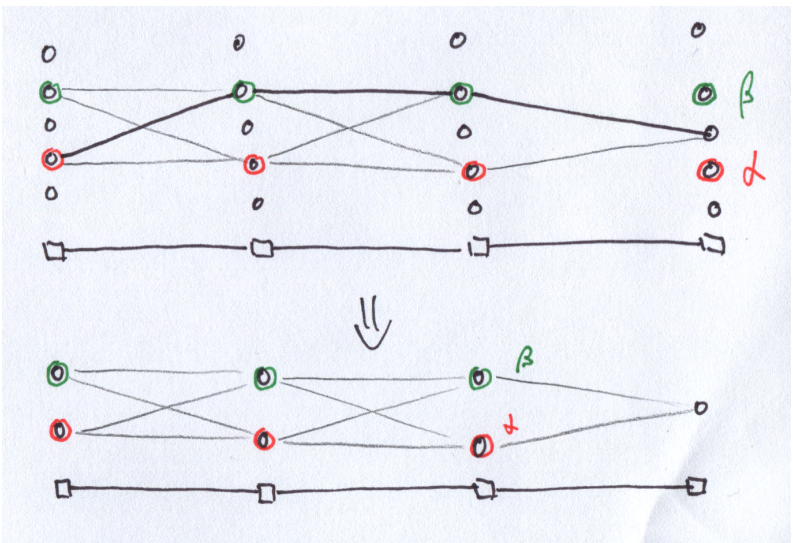
II. Search techniques

General idea:



α-expansion, $\alpha\beta$-swap:





III. Bayesian Decision Theory

The **Bayesian Risk** of a strategy e is the expected loss:

$$R(e) = \sum_x \sum_k p(x,k) \cdot C\Big(e(x), k\Big) \rightarrow \min_e$$

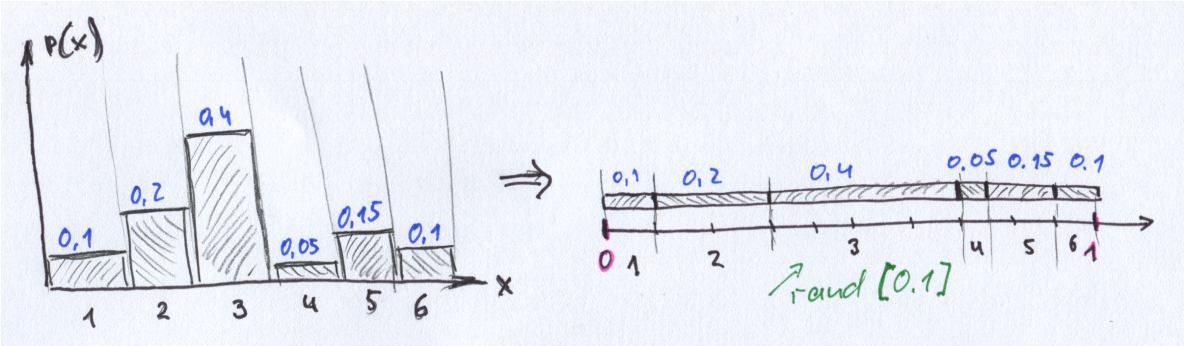
It should be minimized with respect to the decision strategy

Special cases:

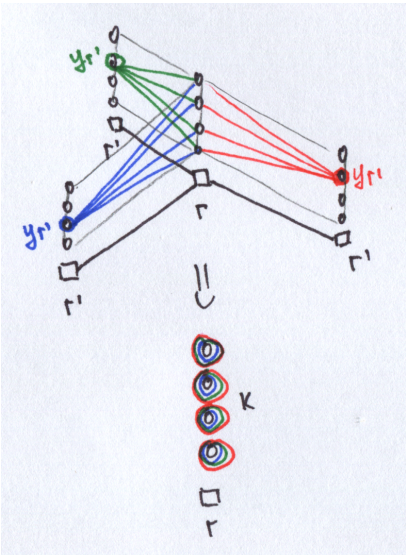
- $C(k,k') = \delta(k \neq k') \rightarrow$ Maximum A-posteriori decision
- Additive loss $C(k,k') = \sum_i c_i(k_i, k'_i) \rightarrow$ the strategy is based on marginal probability distributions
  - Hamming loss $C(k,k') = \sum_i \delta(k_i \neq k'_i)$
    $\rightarrow$ Maximum Marginal decision
  - "Metric" loss $C(k,k') = \sum_i (k_i - k'_i)^2$
    $\rightarrow$ Minimum Marginal Square Error

IV. Gibbs Sampling

How to sample in general:



Sampling in MRF-s:



Markovian property:

$$p(y_i|y_{V\setminus i}) = p(y_i|y_{N(i)})$$

It should be sampled from

$$p(y_i{=}k|y_{N(i)}) \propto$$

$$\propto \exp\left[-\psi_i(k) - \sum_{j\in N(i)} \psi_{ij}(k, y_j)\right]$$

IV. Maximum Likelihood for MRF-s (supervised)

From the Maximum Likelihood formulation

$$F(\theta) = \ln p(L;\theta) = \sum_l \Big[-E(y^l;\theta) - \ln Z(\theta) \Big] =$$

$$= -\sum_l E(y^l;\theta) - |L| \cdot \ln Z(\theta) \to \max_\theta$$

... to the gradient

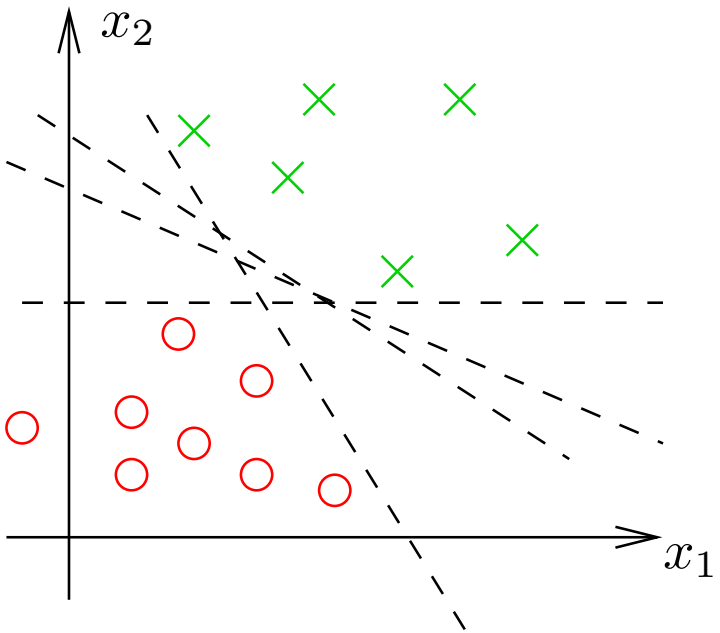
$$\frac{\partial F(\theta)}{\partial \theta} = -\mathbb{E}_{data}\left[\frac{\partial E(y;\theta)}{\partial \theta}\right] + \mathbb{E}_{model}\left[\frac{\partial E(y;\theta)}{\partial \theta}\right]$$

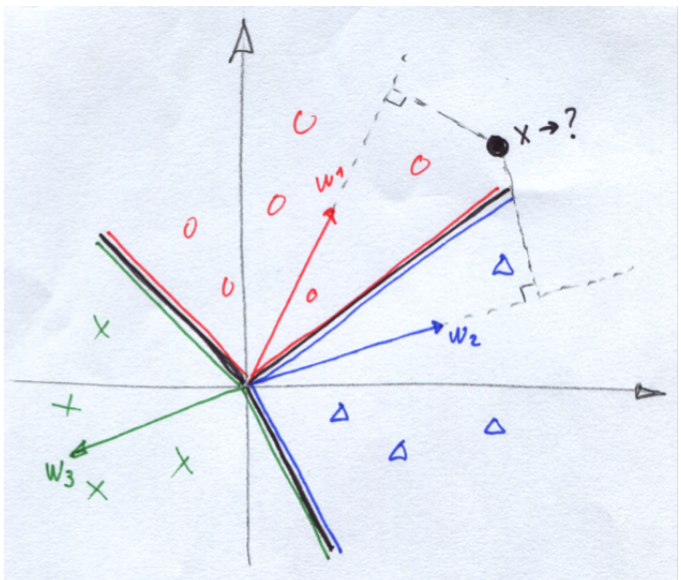
V. Discriminative Learning

A "hierarchy of abstraction":

Generative models $\rightarrow$ Discriminative models $\rightarrow$ Classifiers

Linear classifiers, Perceptron Algorithm, Multi-class Perceptron

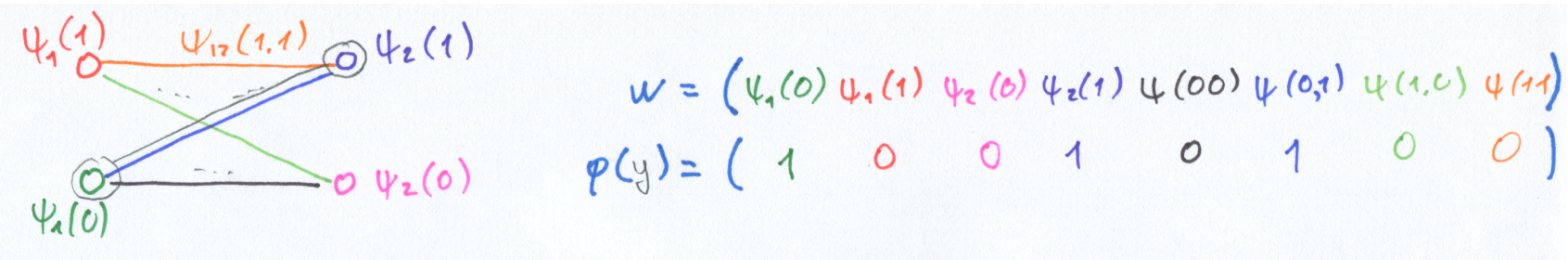




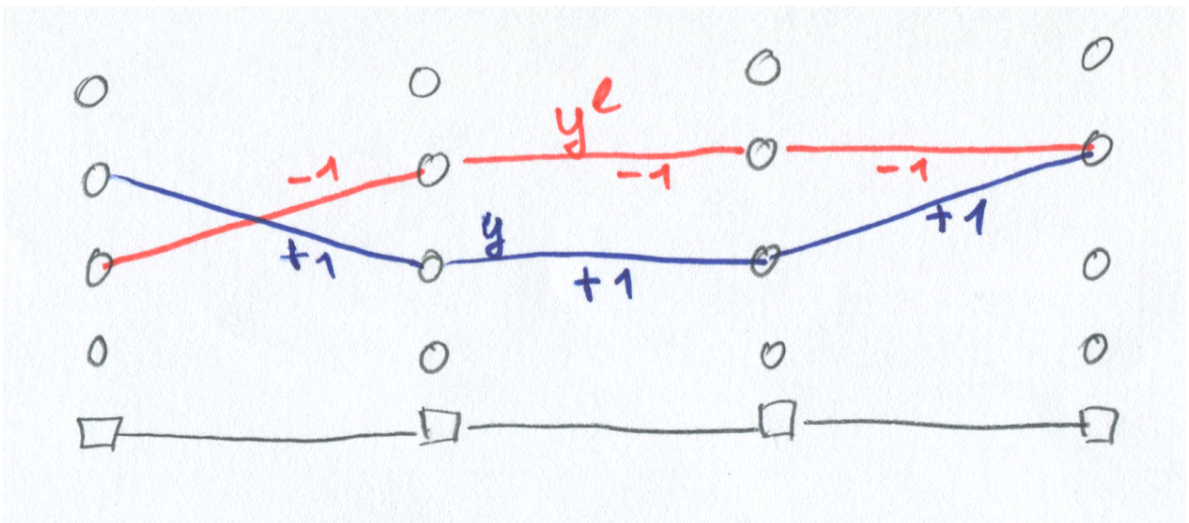
Feature spaces – mappings $\phi(x)$

V. Discriminative Learning

Energy Minimization is a linear classifier



Multi-class perceptron + Energy Minimization:



Exam

On 5.08 at 10:00 (the room not known jet – see www later)

Questions, examples

- Which loss leads to MAP (explain, derive)?
- Write down the probability of a state sequence in a Markov Chain (explain the notations)
- How to obtain an auxiliary binary Energy minimization problem for α-expansion?
- An simple Energy Minimization Problem is given. Perform the ICM (or α-expansion or $\alpha\beta$-swap), starting from a given labeling.
- A simple Energy Minimization Problem for a chain is given. Find the optimal labeling by Dynamic Programming, compute all necessary Bellman functions
- Transform a given pairwise term $\psi_{ij}(y_i, y_j)$ into the canonical form

All (i.e. our part) should be manageable in 15-20 minutes