Machine learning: lecture 19

Tommi S. Jaakkola
MIT CSAIL
tommi@csail.mit.edu

Topics

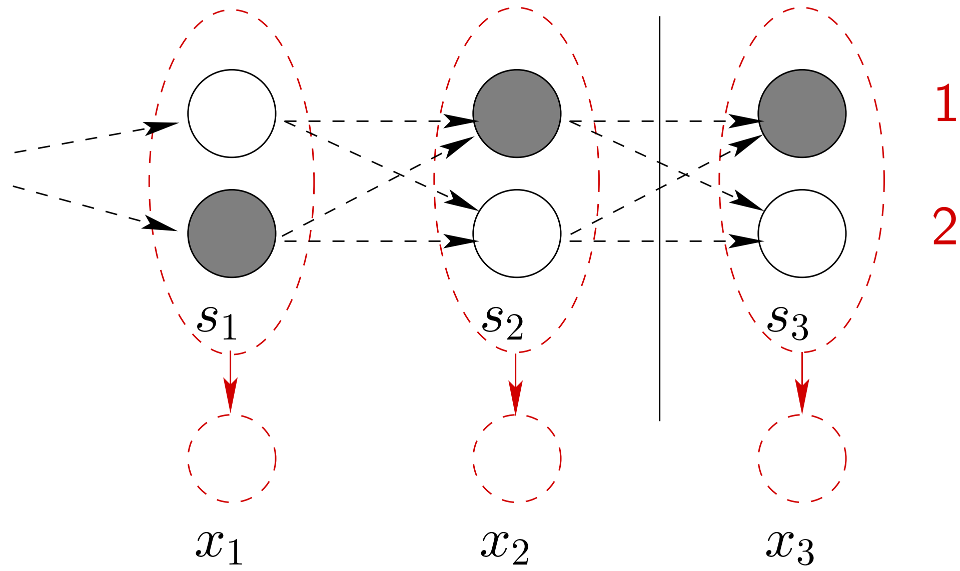
- Hidden markov models
  - Viterbi algorithm, examples
  - alignment
- Graphical models
  - representation
  - examples

HMM problems

- There are several problems we have to solve

1. How do we evaluate the probability of an observation sequence $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$?
   - forward-backward algorithm
2. How do we adapt the parameters of the HMM to better account for the observations?
   - the EM-algorithm
3. How do we uncover the most likely hidden state sequence corresponding to the observations?
   - dynamic programming (Viterbi algorithm)

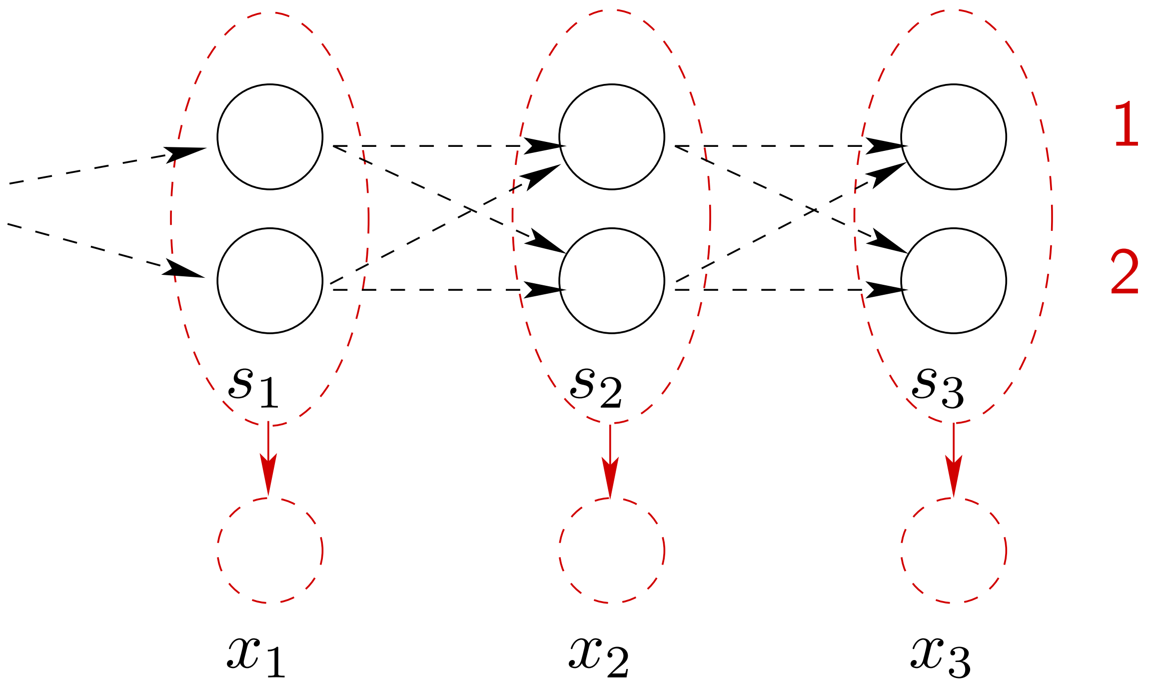
Max-probabilities



- We can recover the most likely hidden state sequence corresponding to a sequence of observations by evaluating the following max-probabilities:

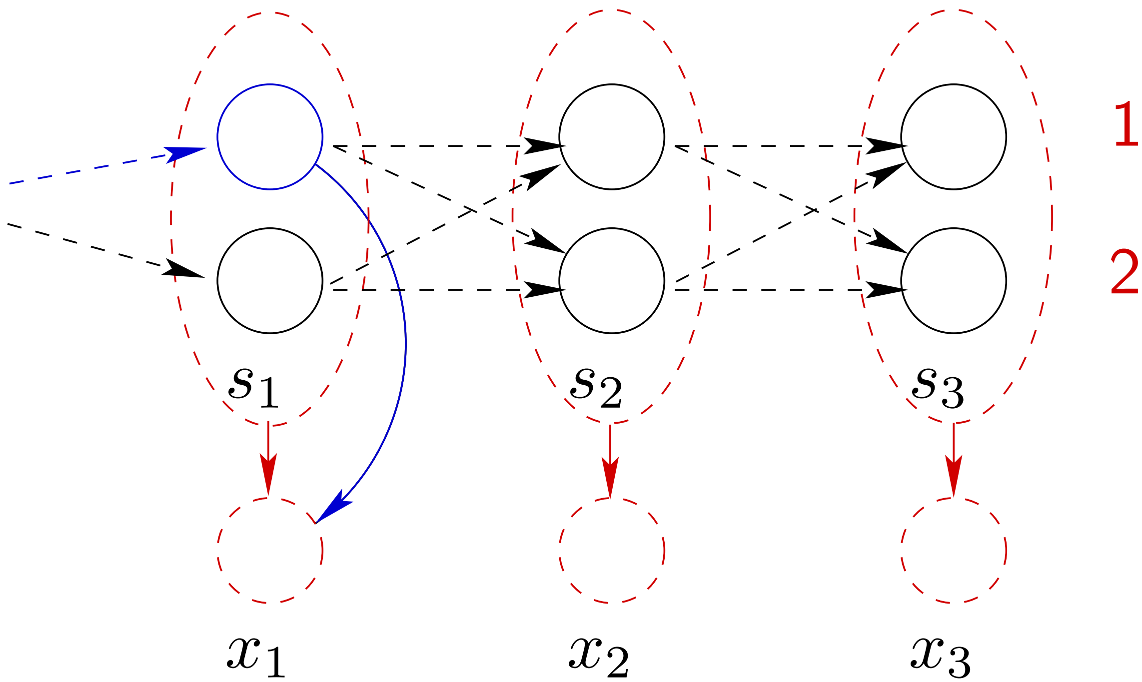
$$\delta_t(i) = \max_{s_1,\ldots,s_{t-1}} P(\mathbf{x}_1,\ldots,\mathbf{x}_t, s_1,\ldots,s_{t-1}, s_t = i)$$

Viterbi algorithm



$$\delta_1(1) = P(x_1, s_1 = 1)$$
$$\delta_1(2) = P(x_1, s_1 = 2)$$

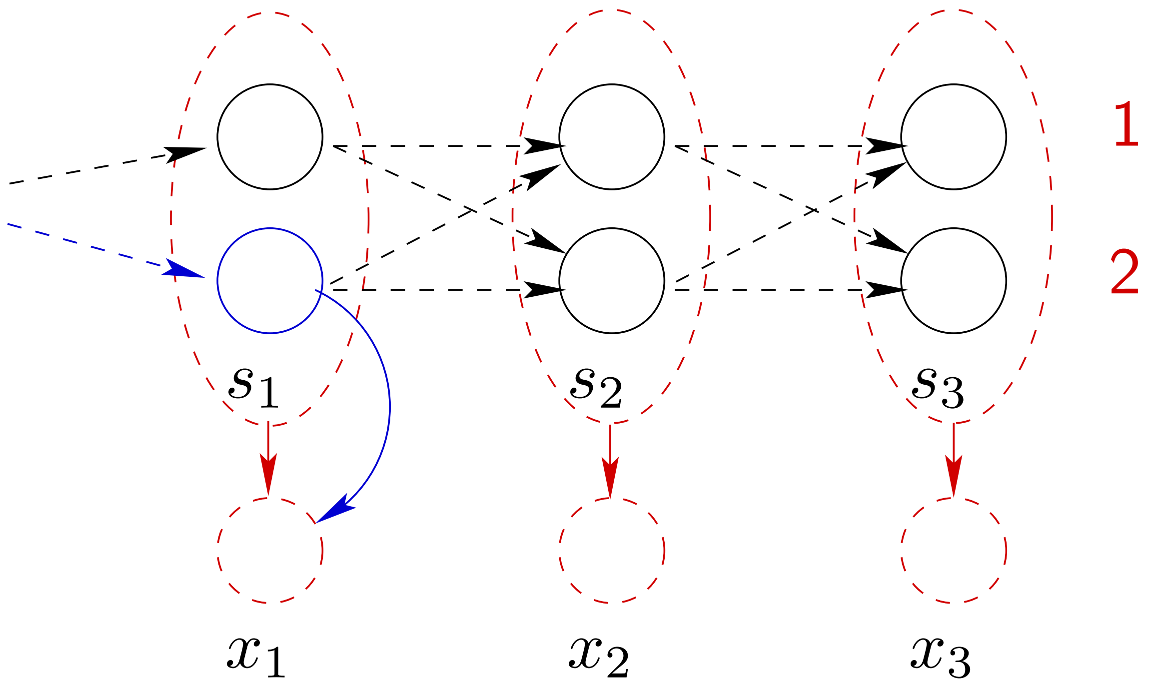
Viterbi algorithm



$$\delta_1(1) = P(x_1, s_1 = 1)$$
$$\delta_1(2) = P(x_1, s_1 = 2)$$

$$\delta_1(1) \quad = \quad P(1)P_x(\mathbf{x}_1|1)$$

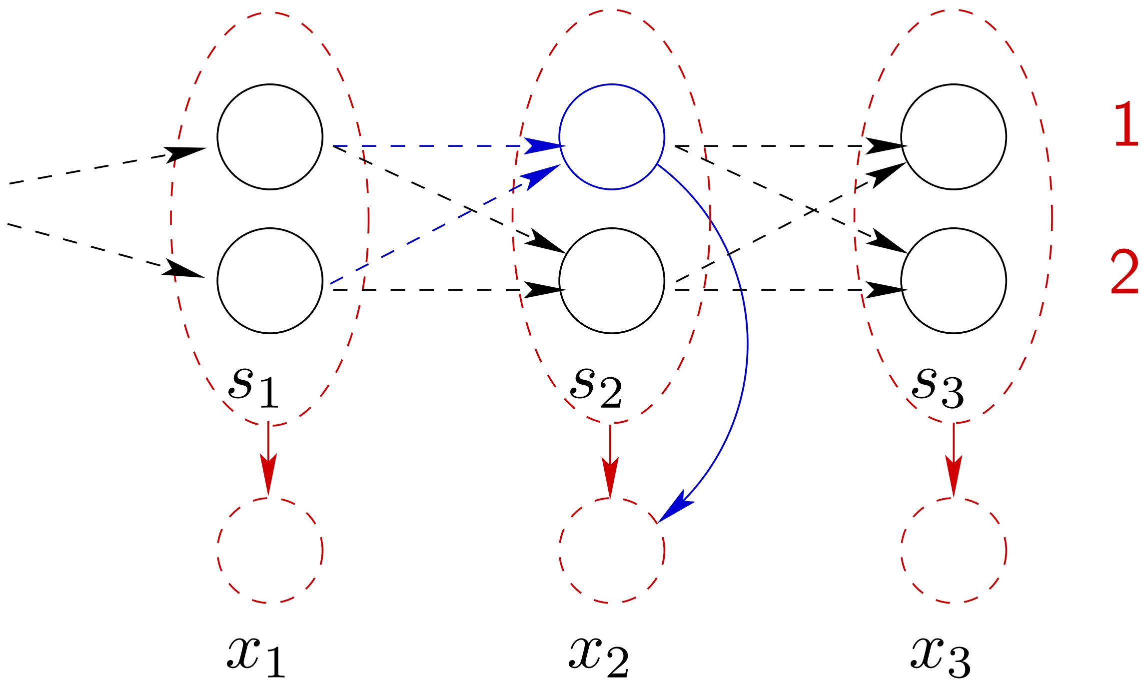
Viterbi algorithm



$$\delta_1(1) = P(x_1, s_1 = 1)$$
$$\delta_1(2) = P(x_1, s_1 = 2)$$

$$\begin{aligned} \delta_1(1) &= P(1)P_x(\mathbf{x}_1|1) \\ \delta_1(2) &= P(2)P_x(\mathbf{x}_1|2) \end{aligned}$$

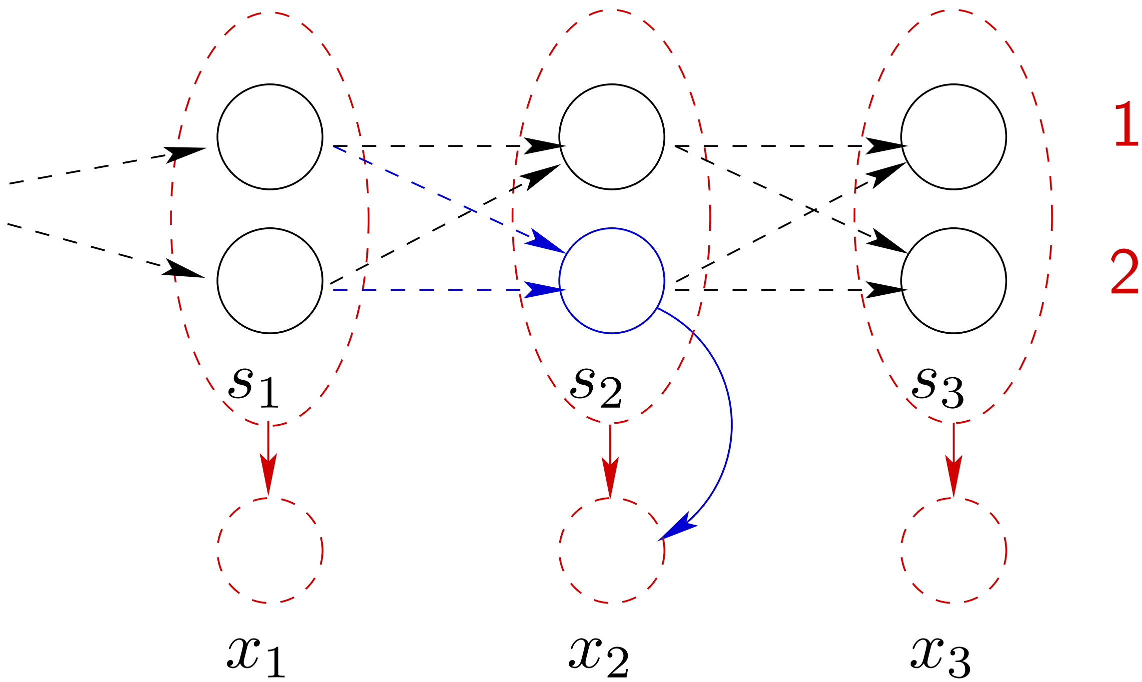
Viterbi algorithm



$$\delta_2(1) = \max_{s_1} P(x_1, x_2, s_1, s_2 = 1)$$
$$\delta_2(2) = \max_{s_1} P(x_1, x_2, s_1, s_2 = 2)$$

$$\delta_2(1) \quad = \quad \max\left\{\delta_1(1)P_1(1|1), \delta_1(2)P_1(1|2)\right\}P_x(\mathbf{x}_2|1)$$

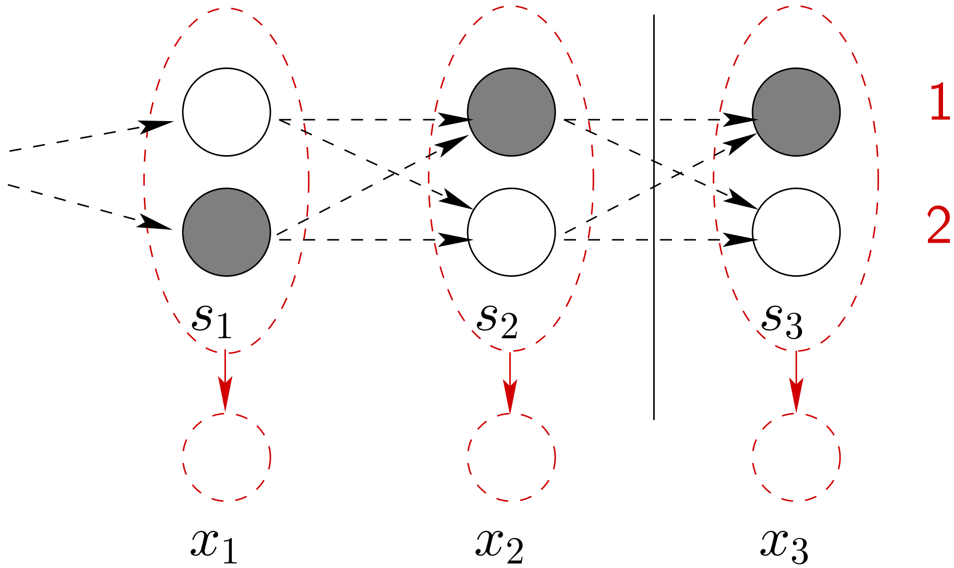
Viterbi algorithm



$$\delta_2(1) = \max_{s_1} P(x_1, x_2, s_1, s_2 = 1)$$
$$\delta_2(2) = \max_{s_1} P(x_1, x_2, s_1, s_2 = 2)$$

$$\begin{aligned}
\delta_2(1) &= \max\left\{\delta_1(1)P_1(1|1), \delta_1(2)P_1(1|2)\right\}P_x(\mathbf{x}_2|1) \\
\delta_2(2) &= \max\left\{\delta_1(1)P_1(2|1), \delta_1(2)P_1(2|2)\right\}P_x(\mathbf{x}_2|2)
\end{aligned}$$

Viterbi algorithm

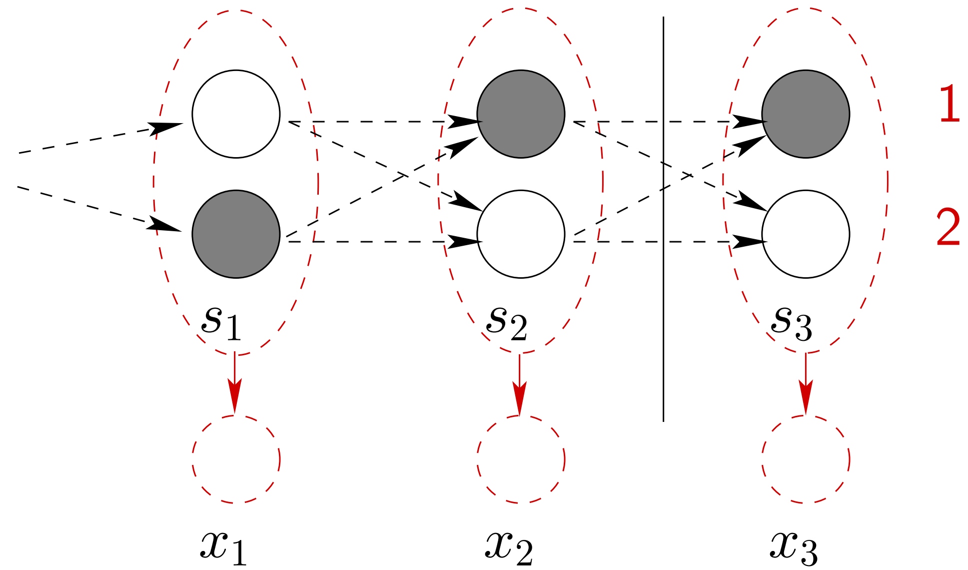


$$\delta_t(i) = \max_{s_1,\ldots,s_{t-1}} P(\mathbf{x}_1, \ldots, \mathbf{x}_t, s_1, \ldots, s_{t-1}, s_t = i)$$

- We get the following recursive equation for calculating the max probabilities:

$$\begin{aligned} \delta_1(i) &= P(i) P_x(\mathbf{x}_1 | i) \\ \delta_t(i) &= \max_j \left\{ \delta_{t-1}(j) P_1(i|j) \right\} P_x(\mathbf{x}_t | i) \end{aligned}$$

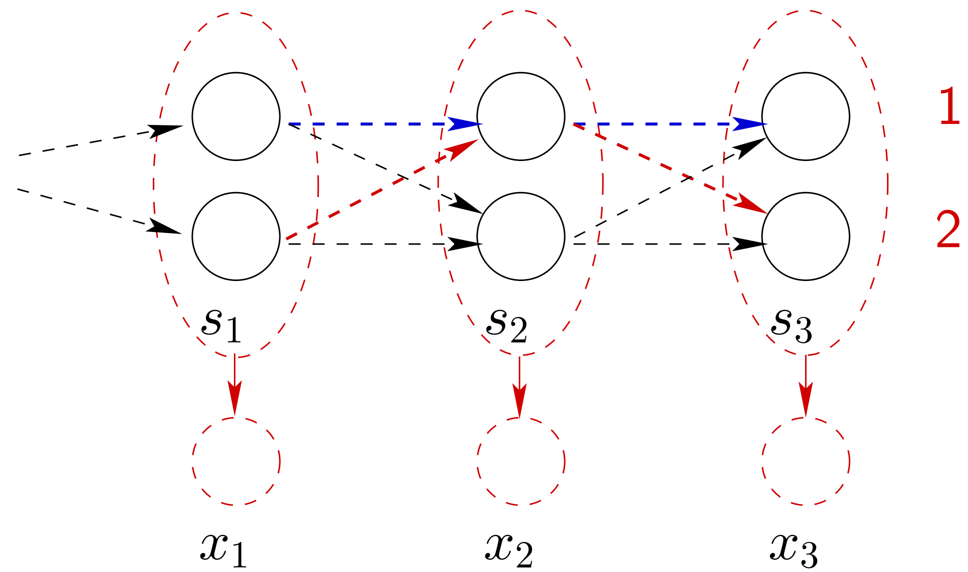
Viterbi algorithm: back-tracking



- We can recover the most likely state sequence by working backwards:

$$
\begin{aligned}
s_n^* &= \arg\max_i \delta_n(i) \\
s_t^* &= \arg\max_j \left\{ \delta_t(j) P_1(s_{t+1}^* | j) \right\}
\end{aligned}
$$

Viterbi algorithm: properties

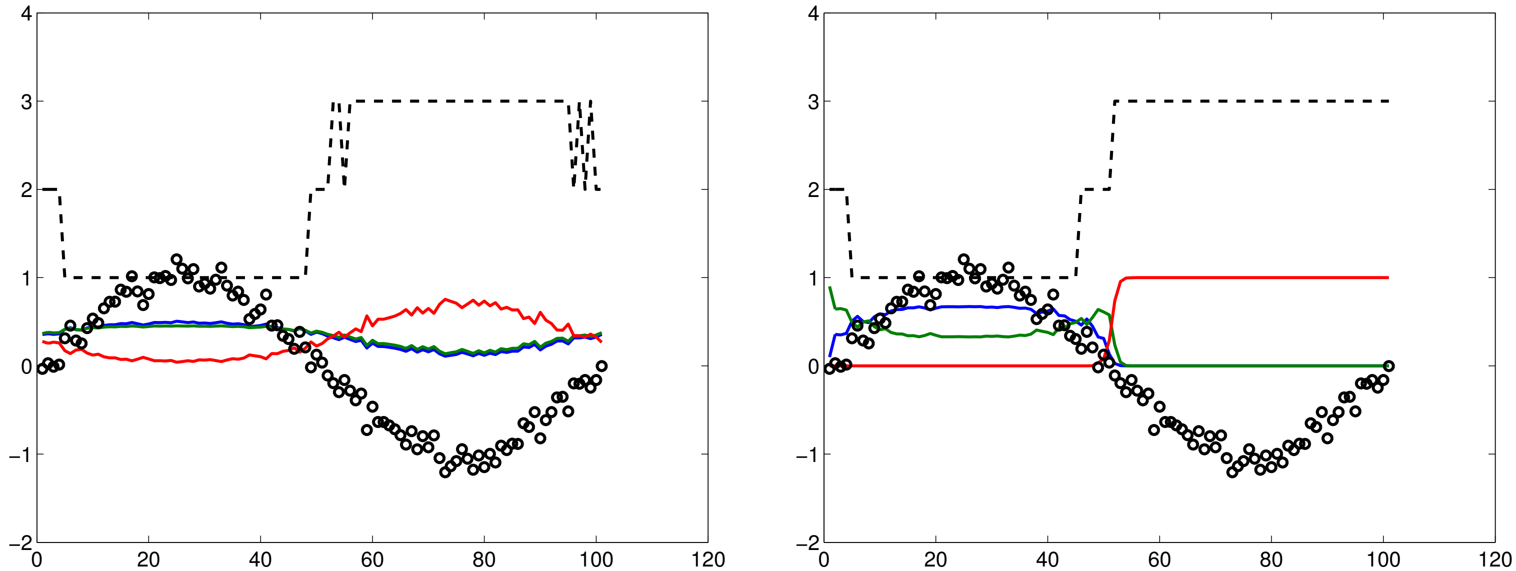


- The most likely path has the property that any partial path is also optimal:

  If $s_t^* = i$ then $\{s_1^*, \ldots, s_t^*\}$ is also the most likely state sequence forced to end up in $s_t = i$ at time t given only $\mathbf{x}_1, \ldots, \mathbf{x}_t$.

Viterbi algorithm: example

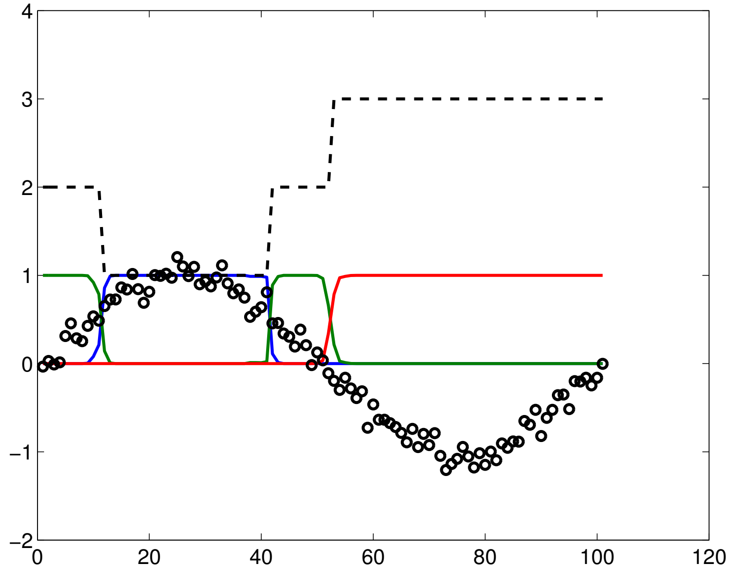
- Same example as in the EM case (3 states, Gaussian outputs)



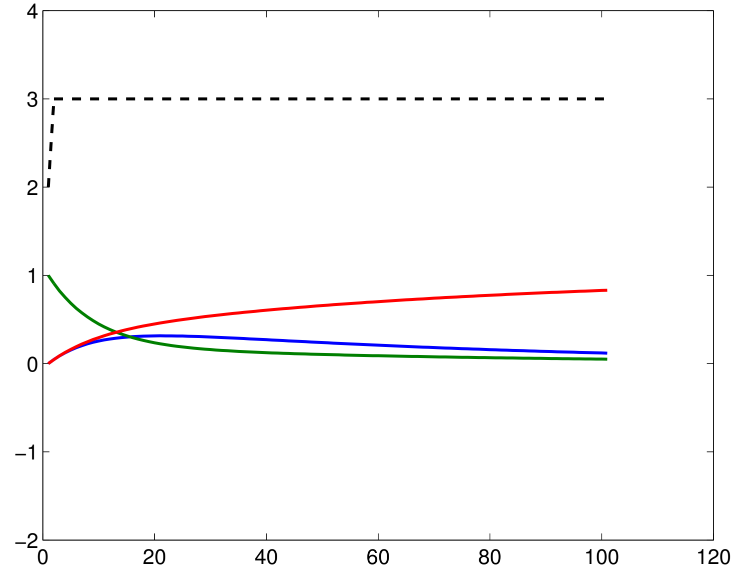
after 0 iterations   after 7 iterations

- The most likely hidden state sequence $\{s_0^*, \ldots, s_n^*\}$ need not agree with the most likely states derived from the posterior marginals $\gamma_t(i)$

Example cont'd



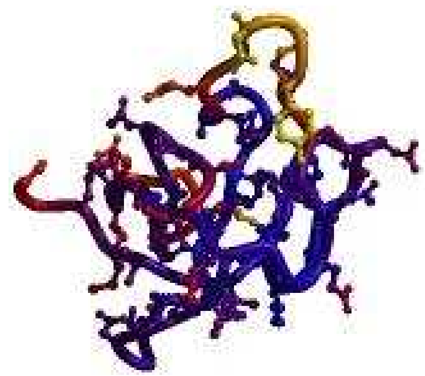
final



ML model, no observations

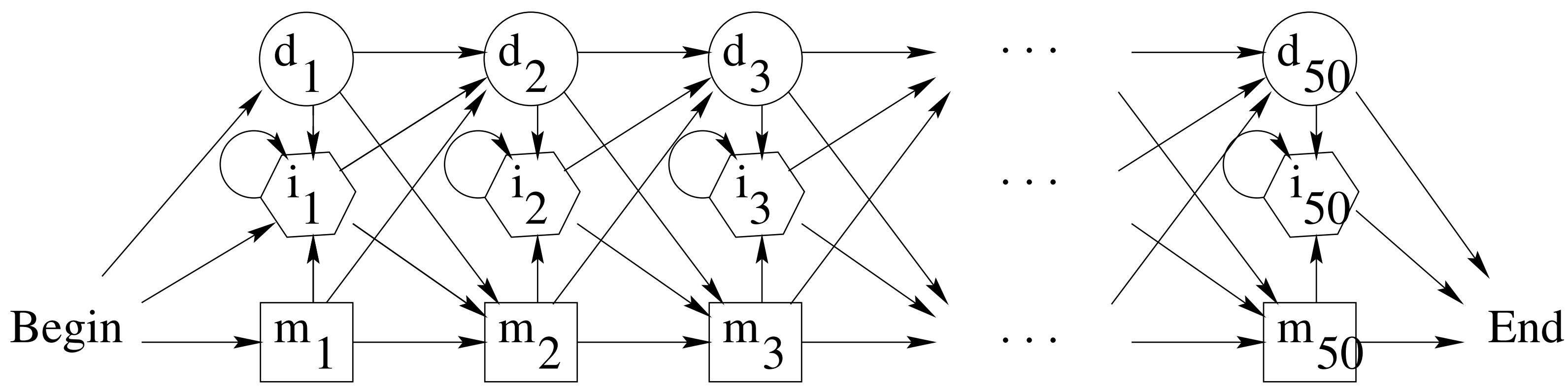
Linear HMMs, alignment

- Proteins



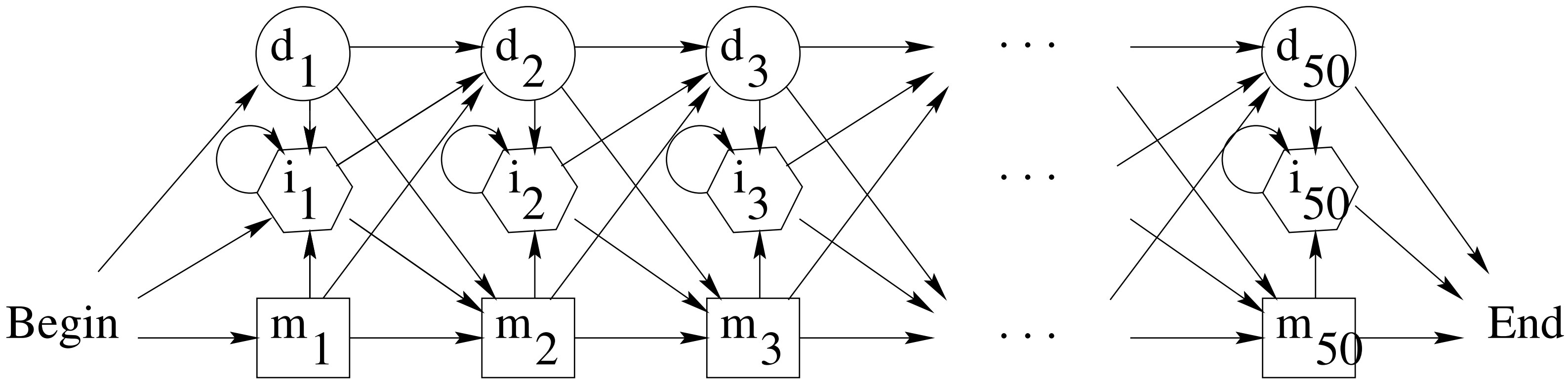
GVAALGAKVLAQIGVAVSHLGDEGKMVAQMKAVGVRHKGYGNKHIKAQYFEPLGASLLSAMEHRIG

- A Linear HMM model for protein sequences



GVAALGAKVLAQIGVAVSHLGDegkMVAQMKAVGVRHKgygNK-HIKAQYFEPLGASLLSAMEHRIG

Linear HMMs, multiple alignment




```
-VKGHGKKVADALTNAVAHVDD.....MPNALSALSDLHA...HKLRVDPV.NFKLLSHCLLVTLAAHLP
KVKAHGKKVLGAFSDGLAHLDN.....LKGTFATLSELHC...DKLHVDPE.NFRLLGNVLVCVLAHHFG
DLKKHGVTVLTALGAILKKKGH.....HEAELKPLAQSHA...TK-HKIPIkYLEFISEAIIHVLHSRHP
PFETHANRIVGFFSKIIGELPN.....IEADVNTFVASHK...PR-GVTHD.QLNNFRAGFVSYMKAH--
DVRWHAERIINAVNDAVASMDDtek..MSMKLRDLSGKHA...KSFQVDPQ.YFKVLAAVIADTVAA---
ELQAHAGKVFKLVYEAAIQLQVtgvvvTDATLKNLGSVHV...SK-GVADA.HFPVVKEAILKTIKEVVG
GVAALGAKVLAQIGVAVSHLGDegk..MVAQMKAVGVRHKgygNK-HIKAQ.YFEPLGASLLSAMEHRIG
```

Topics

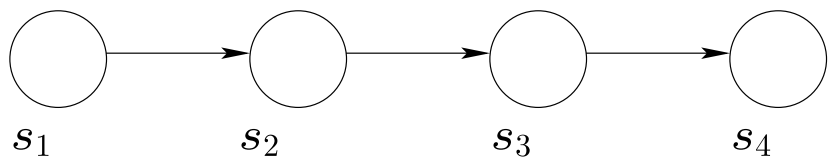
- Hidden markov models
  - Viterbi algorithm, examples
  - alignment
- Graphical models
  - representation
  - examples

What is a good representation?

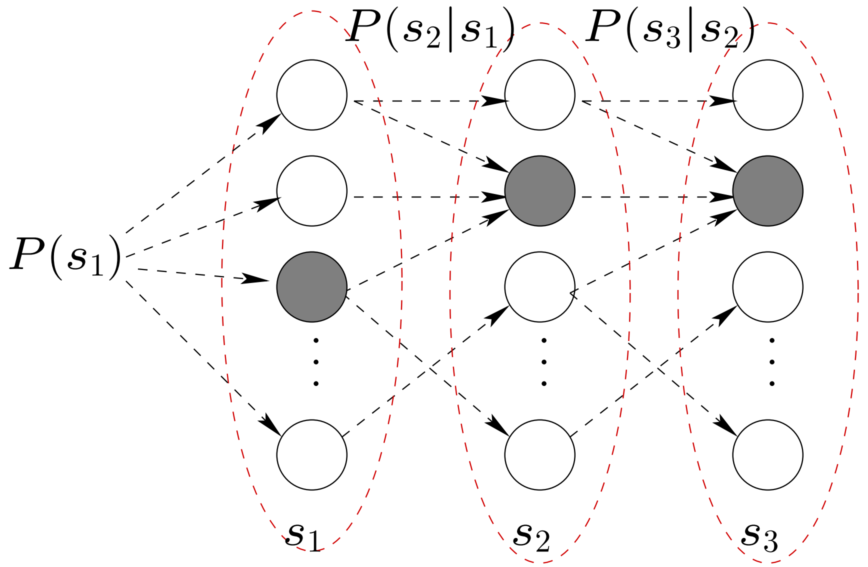
- Properties of good representations
  1. Explicit
  2. Modular
  3. Permits efficient computation
  4. etc.

Representation: explicit

- Representation in terms of variables and dependencies (a graphical model):



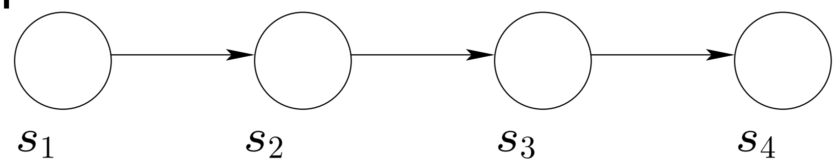
- Representation in terms of state transitions (transition diagram)



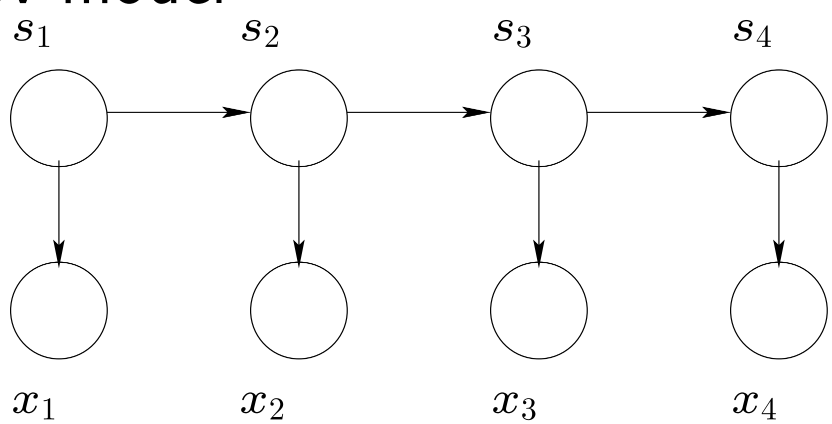
Representation: modular

- We can easily add/remove components of the model

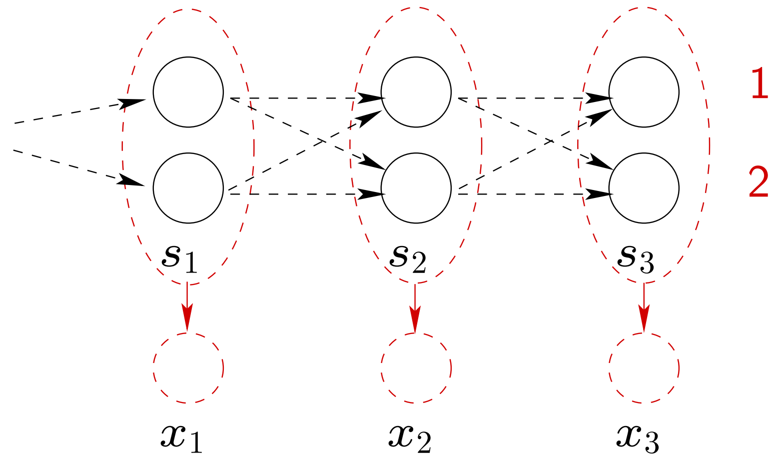
Markov model



Hidden Markov model



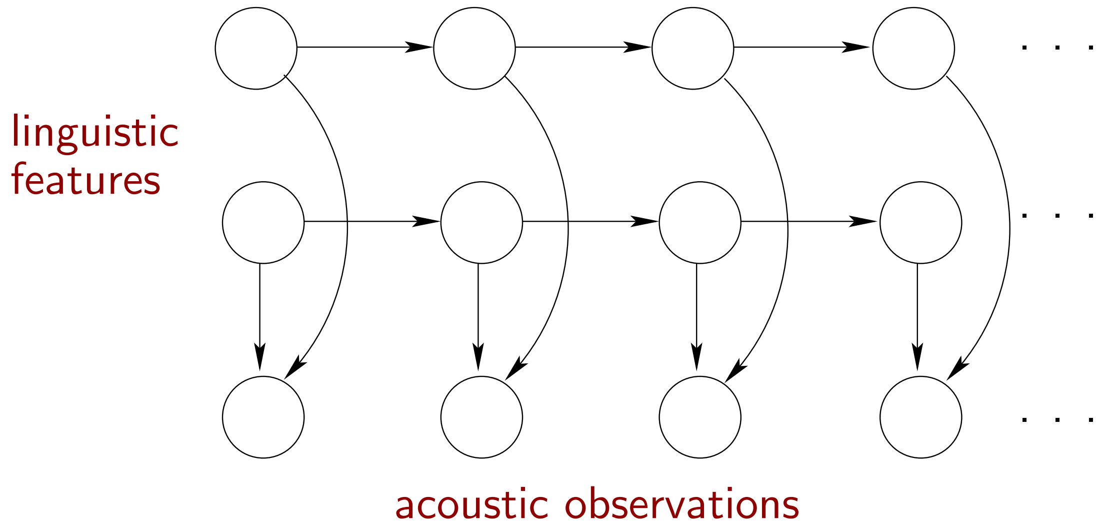
Representation: efficient computation



- Forward-backward probabilities
- EM algorithm
- Max-probabilities (viterbi)

Graphical models: examples

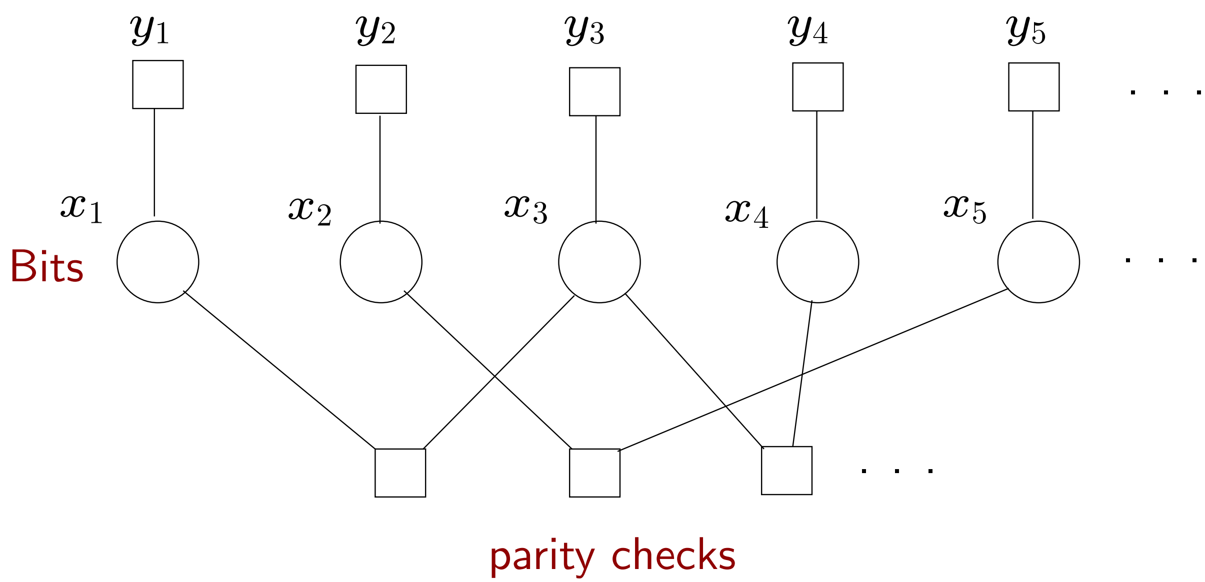
- (Factorial) Hidden Markov models



(a.k.a. directed graphs, Bayesian networks)

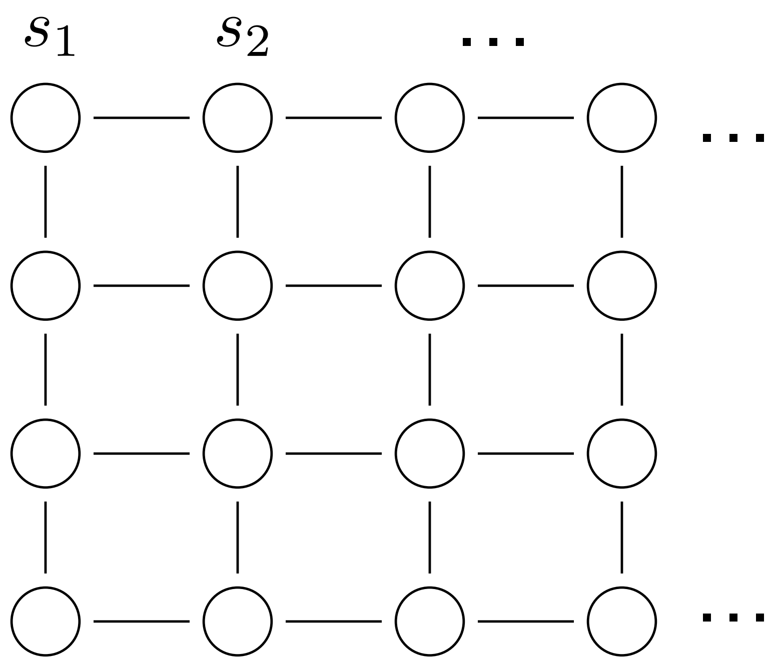
Graphical models: examples

- Factor graphs and codes (information theory)



Graphical models: examples

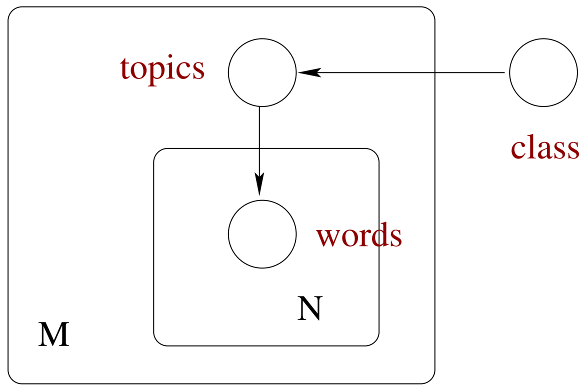
- Lattice models, Ising model (physics)

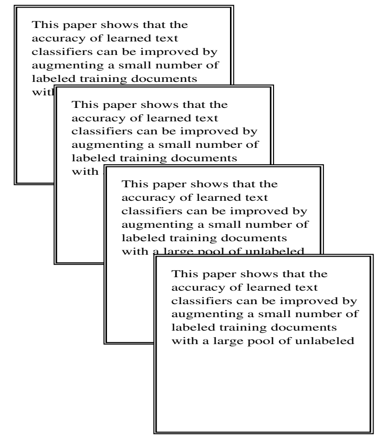


(a.k.a Markov random field)

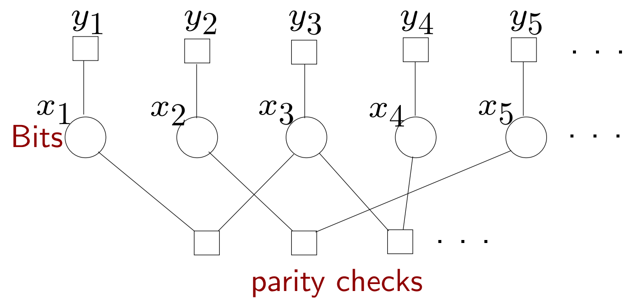
Graphical models: examples

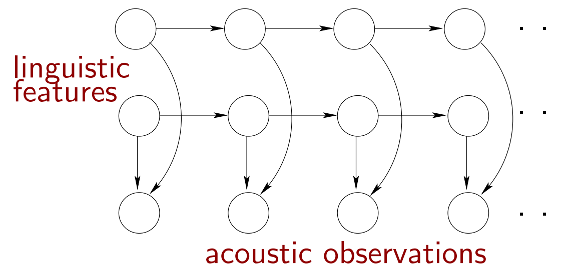
- Plates and relations (information retrieval, biology, etc.)

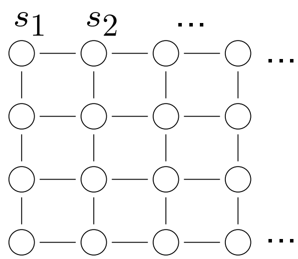


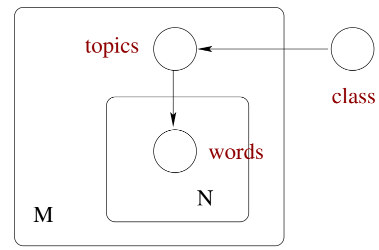


Graphical models









- Graph semantics:

  graph $\Rightarrow$ separation properties $\Rightarrow$ independence

- Association with probability distributions:

  independence $\Rightarrow$ family of distributions

- Inference and estimation:

  graph structure $\Rightarrow$ efficient computation