NPFL099 - Statistical dialogue systems

Dialogue management

Belief monitoring II

Filip Jurčíček

Institute of Formal and Applied Linguistics
Charles University in Prague
Czech Republic

Home page: http://ufal.mff.cuni.cz/~jurcicek

Version: 25/03/2013

Belief monitoring

- Some researchers:
  - enumerate the most likely states and prune the others
  - mixture model belief monitoring
    - J. Henderson and O. Lemon, “**Mixture model POMDPs for efficient handling of uncertainty in dialogue management**,” pp. 73-76, Jun. 2008.
  - group similar states
    - S. Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson and K. Yu (2010). "**The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management.**"
  - belief propagation
    - B. Thomson and S. Young (2010). "**Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems.**"

Grouping similar states

- Hidden Information State model
  - key idea – group states for which there is no evidence that their probabilities differ
  - this is similar to what we explored in the mixture model approach
  - however, we do not work with states directly
  - instead, we have partitions aka groups of states

- S. Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson and K. Yu (2010). "The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management."

Hidden Information State basics

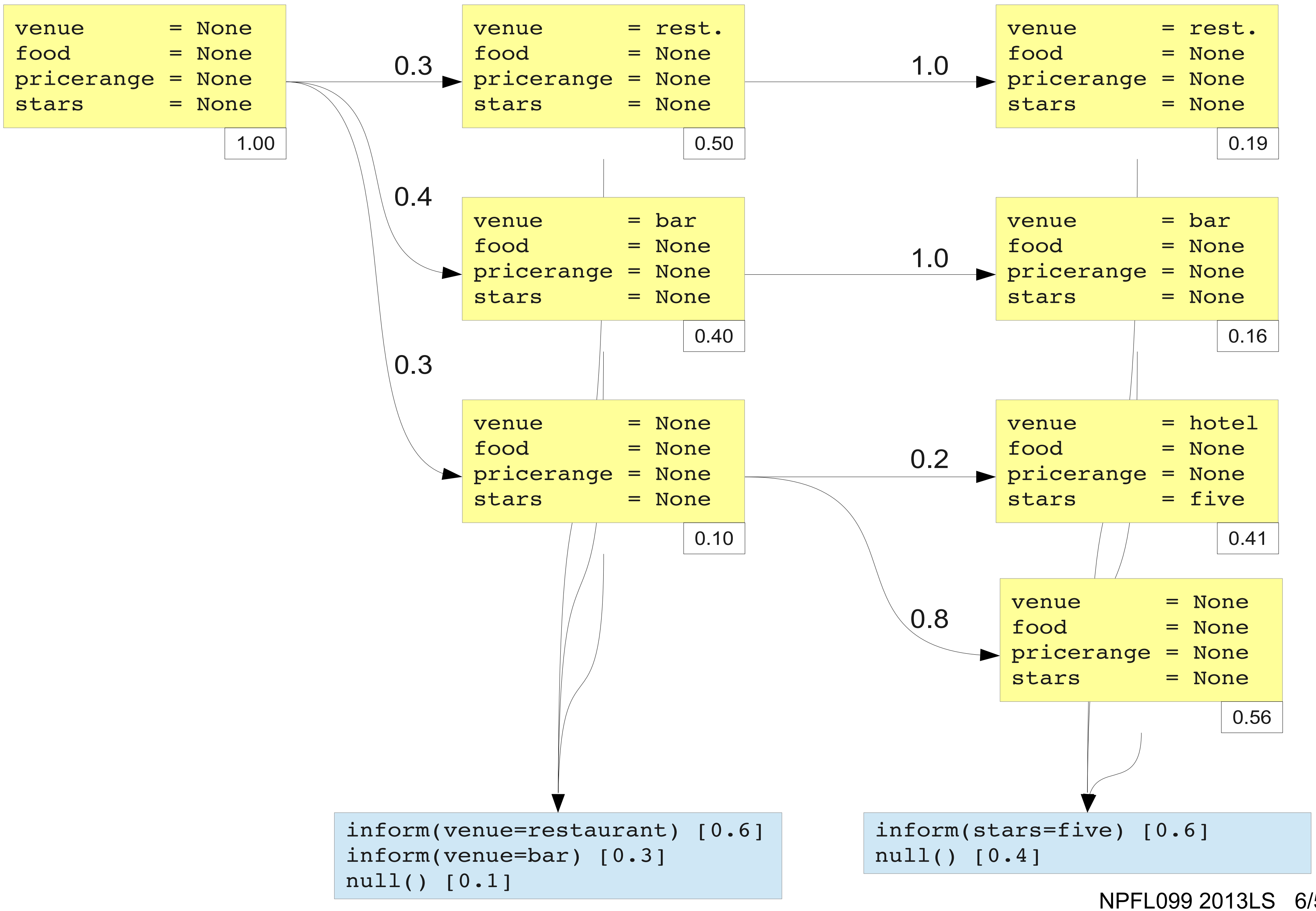
- Initially, there is only one partition
- Then each turn,
  - based on the observations (as in the mixture model) and some domain ontology
  - the partitions are expanded to accommodate new evidence
    - split the partitions matching the observations
    - otherwise leave the partitions as they are
- Partitions are split according an ontology

The ontology

- Defines of the structure of the states/partitions
- Defines prior for splitting of the partitions

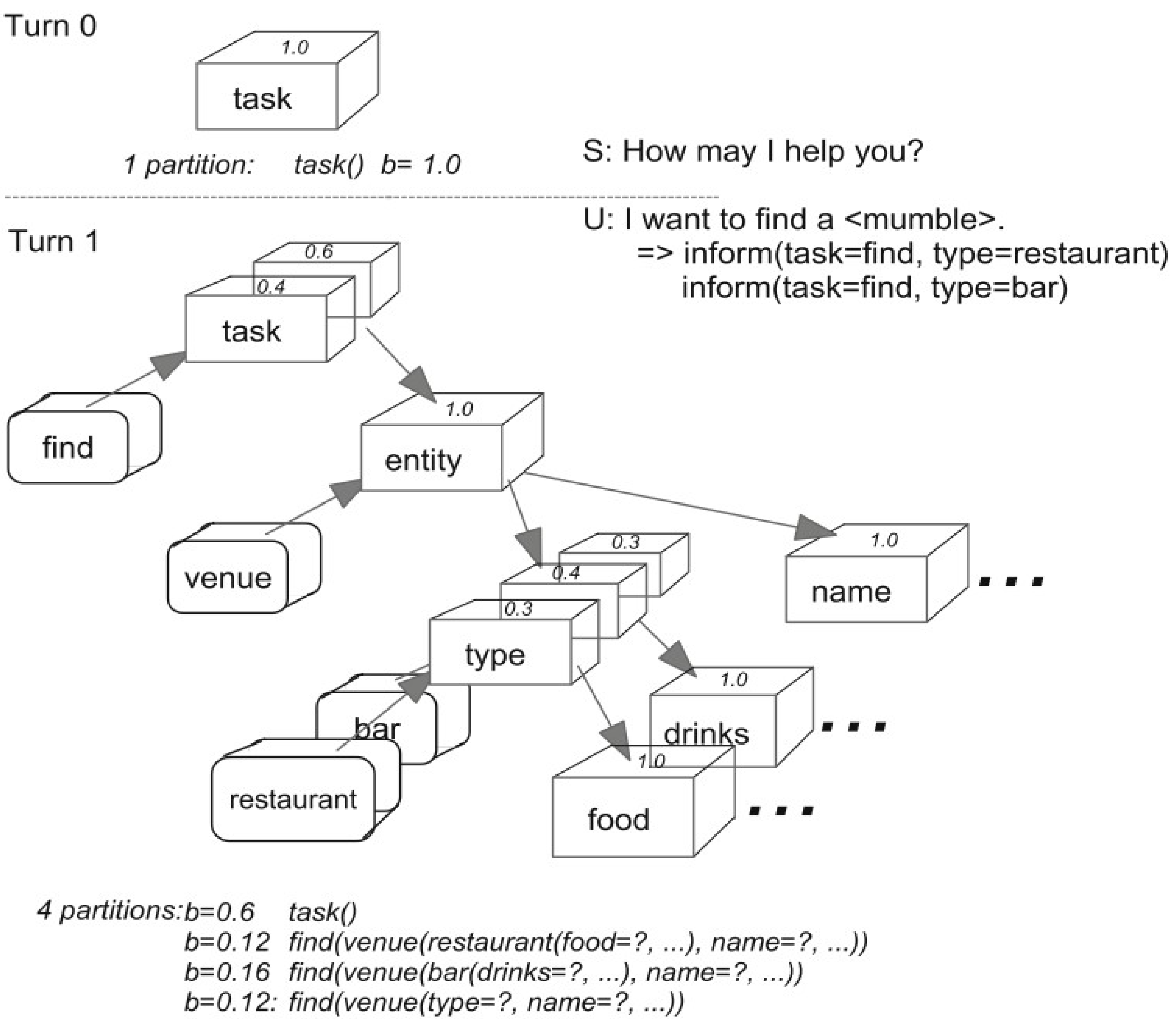
```
# define main entities in the domain
entity -> venue(type, +area, +near, -addr, -phone, -postcode,
*reviews, *rating, +pricerange, -price) [0.8];

# places to eat
type -> restaurant(+food) [0.3];
type -> bar(childrenallowed, hasinternet, hastv) [0.4];
type -> hotel(stars) [0.2];

# atributes
pricerange      = ( free | cheap | moderate | expensive);
area            = ( girton | arbury| … | citycentre | castlehill);
food            = ( American | … | "Chinese takeaway");
hasinternet     = ( true | false);
hastv           = ( true | false);
childrenallowed = ( true | false);
stars           = ( one | two | … | five );
```

Partition splitting



Another view: partition splitting

S. Young et al. / Computer Speech and Language 24 (2010) 150–174



Partition splitting

- Although, no proper transition model defined
  - it defines some prior on some types of states
- The ontology prevents generation of partitions not supported by the ontology
- The way how the probability mass is distributed depends on the order of splitting
- The most interesting is the observation model

HIS observation model

- Observation model: $p(o_t | s_t)$
- HIS Observation model: $p(o_t | s_t, a_{t-1})$
  - factor the model into
    - bigram dialogue act type model
    - item matching model

$$p(o_t | p_t, a_{t-1}) \approx p(T(o_t) | T(a_{t-1}))\, p(M(o_t, p_t, a_{t-1}))$$

- T(...) - denotes the dialogue act type
- M(...) - denotes whether the observation matches the partition and the system dialogue act

Matching the user dialogue act

- The matching process is defined by a set of heuristic rules

- To get positive match
  - For inform, confirm dialogue acts
    - the act slot values should equal to the partition values
  - For affirm dialogue act
    - tries to match the system's confirmed value
    - e.g.
      - S: confirm(food=English)
      - U: affirm()
  - For negate dialogue act
    - the act slot value should not equal to the partition values

HIS summary

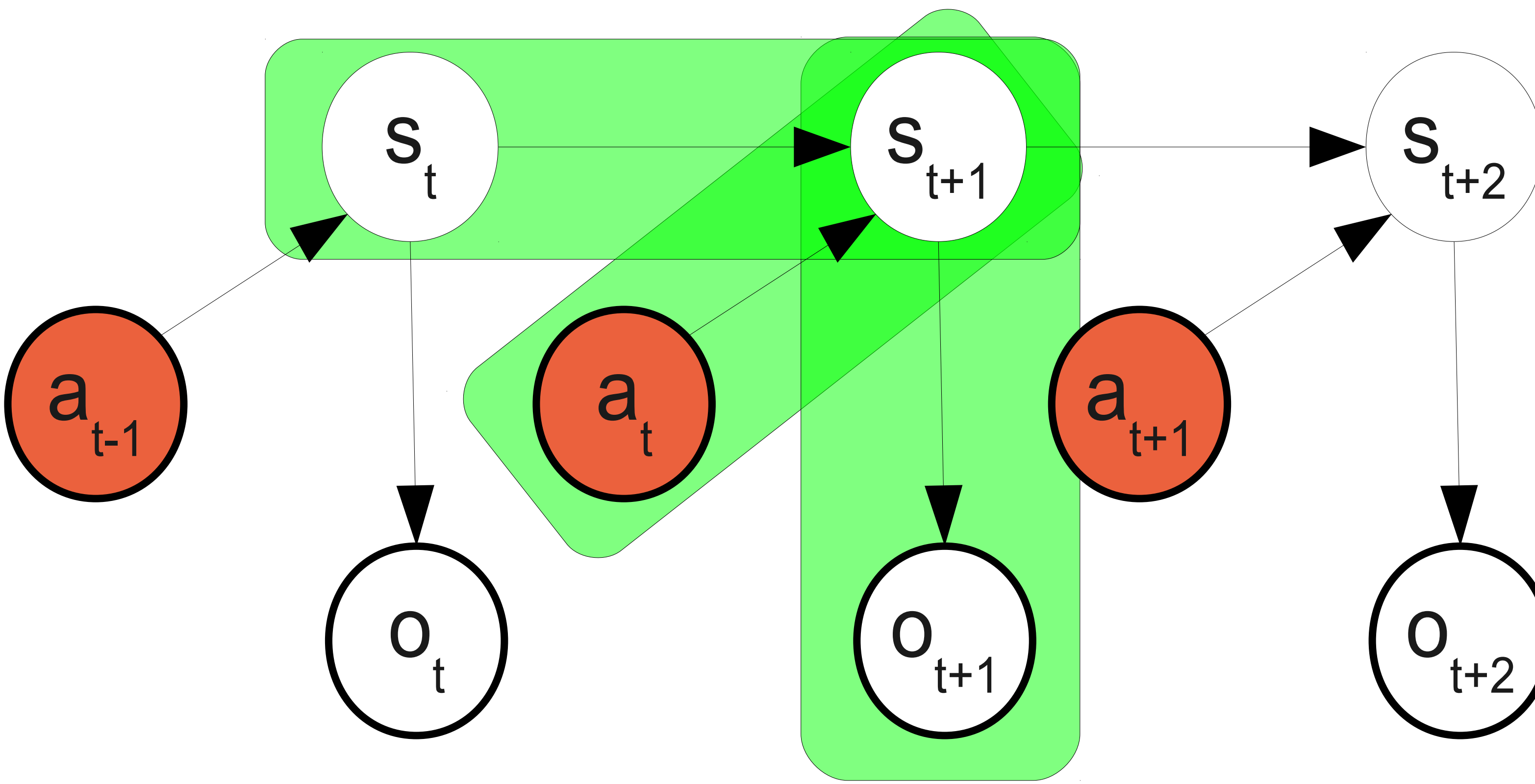
- Efficiently groups similar states into partitions
- Updates performed only on partitions
- Although there is prior for splitting partition,
  - it does no explicit model dynamics of states
- The model of splitting can be extended for efficient partition merging and pruning
- The model allows for explicit tracking of
  - “I do not want Chinese”

Bayesian approach to belief monitoring

- Maintain prob. distribution over all possible states: **b(s)**

$$b(s_{t+1}) \approx p(o_{t+1}|s_{t+1}) \sum_{s_t} p(s_{t+1}|a_t, s_t)\, b(s_t)$$

$$\approx p(o_{t+1}) \sum_{o_{t+1}} p(o_{t+1}|s_{t+1}) \sum_{s_t} p(s_{t+1}|a_t, s_t)\, b(s_t)$$

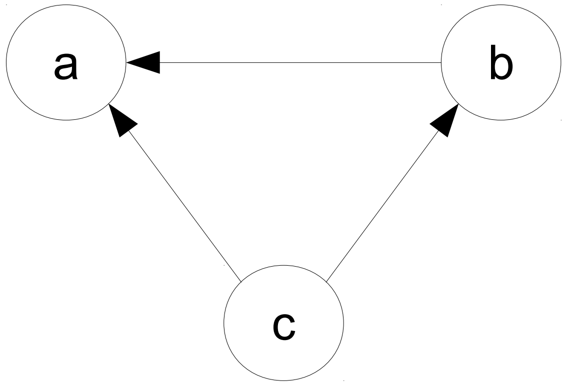


Bayesian approach

- The key idea is to represent the model
  - as a graphical model
- And then, to use general exact or approximate inference methods
  - to monitor the belief state

Graphical models

- Provide simple way to visualize probabilistic models
- Give insight into properties of the model, e.g. conditional independence
- Help to understand complex inference methods



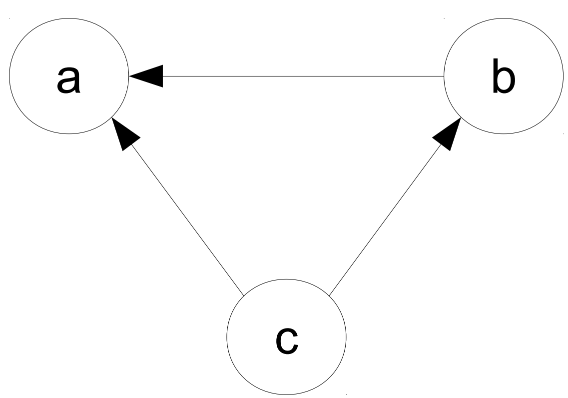
Bayesian Networks

- BN is a directed graphical model consisting of
  - nodes – random variables
  - links – probabilistic relationship between random var.

- The basic idea is to represent a complex distribution by a product of simpler distribution

$$p(a,b,c)=p(a|b,c)\,p(b|c)\,p(c)$$

- This can be graphically represented as

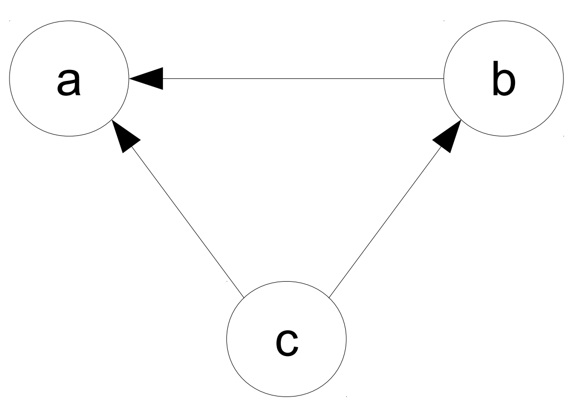


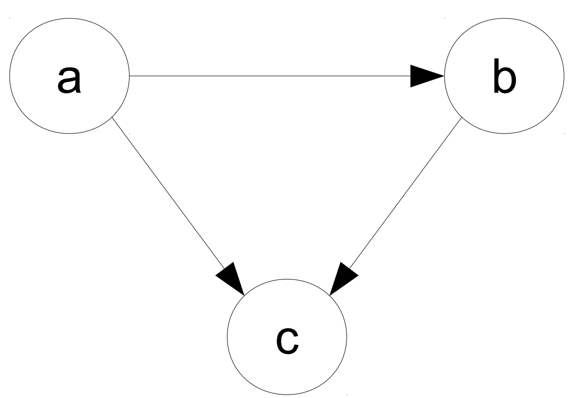
Factorization

- Factorization is not unique
  - it can have many, theoretically equivalent, forms

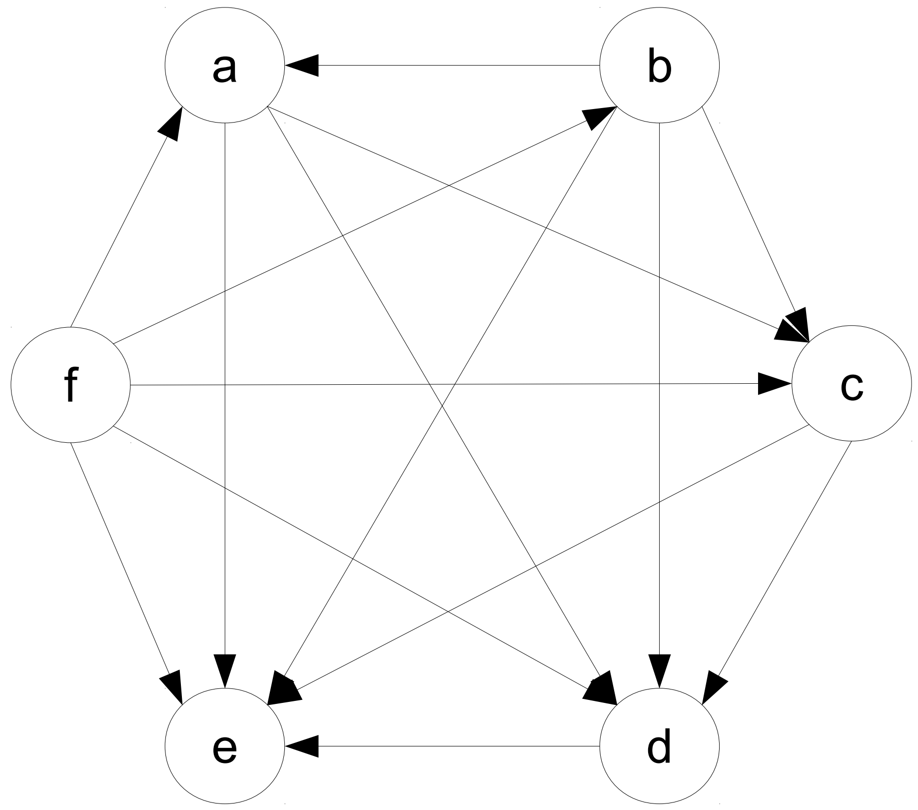
$$p(a,b,c) = p(a|b,c)\,p(b|c)\,p(c)$$

$$= p(c|a,b)\,p(b|a)\,p(a)$$





Fully connected networks



$$p(a,b,c,d,e,f)=p(e|a,b,c,d,f)\,p(d|a,b,c,f)\,p(c|a,b,f)\,p(a|b,f)\,p(b|f)\,p(f)$$

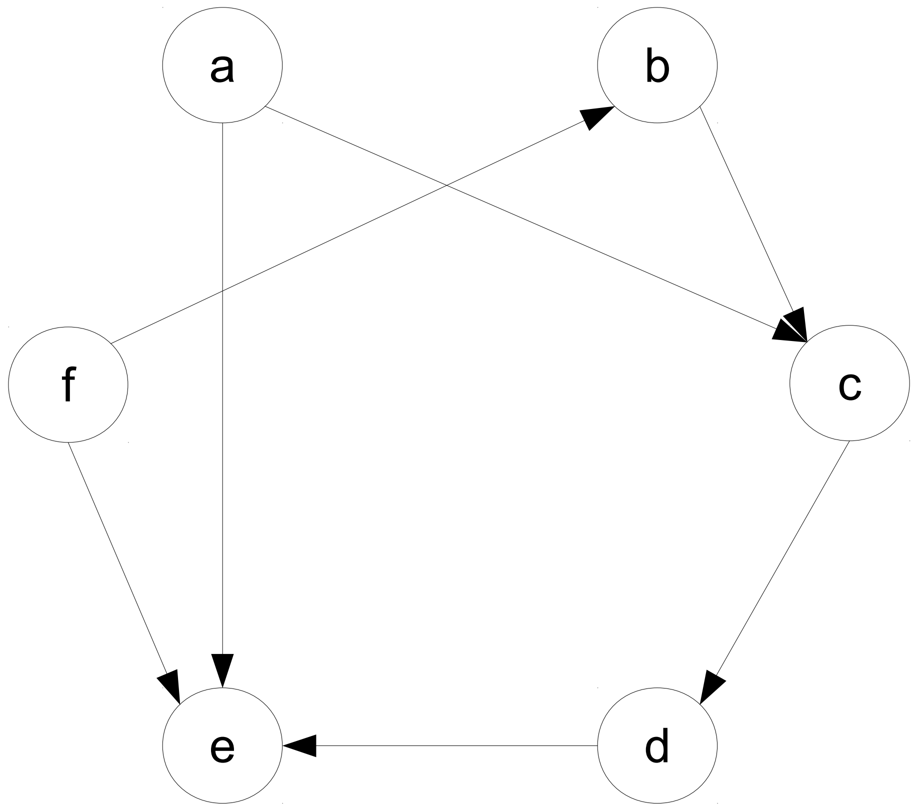
Marginalization

- Computing joint distribution of only some subset of variables

$$p(a,b,c)=\sum_d \sum_e \sum_f p(a,b,c,d,e,f)$$

- Trivial. However, it can be slow.

Partially connected networks



$$p(a,b,c,d,e,f) = p(e|a,d,f)\,p(d|c)\,p(c|a,b)\,p(b|f)\,p(a)\,p(f)$$

Marginalization

- Marginalization on factored partially connected network speeds up the inference

$$p(a,b,c)=\sum_d \sum_e \sum_f p(e|a,d,f)\,p(d|c)\,p(c|a,b)\,p(b|f)\,p(a)\,p(f)$$

- Use the fact that x distributes over +

$$xy+xz \qquad = \qquad x(y+z)$$

2 multiplies + 1 addition   1 multiply + 1 addition

Marginalization on factored joint dist.

- Here, you need
  - |d|.|e|.|f| additions
  - |d|.|e|.|f|*5 multiplications

$$p(a,b,c)=\sum_d \sum_e \sum_f p(e|a,d,f)\, p(d|c)\, p(c|a,b)\, p(b|f)\, p(a)\, p(f)$$

- In this case, you need
  - |d|.|e|.|f| additions
  - |d|.|f| + 3 multiplications

$$p(a,b,c)=p(a)\, p(c|a,b) \sum_f p(b|f)\, p(f) \left(\sum_d p(d|c) \left(\sum_e p(e|a,d,f)\right)\right)$$

Conditional independence

- Independence of two random variables

$$p(a, b) = p(a)\,p(b)$$

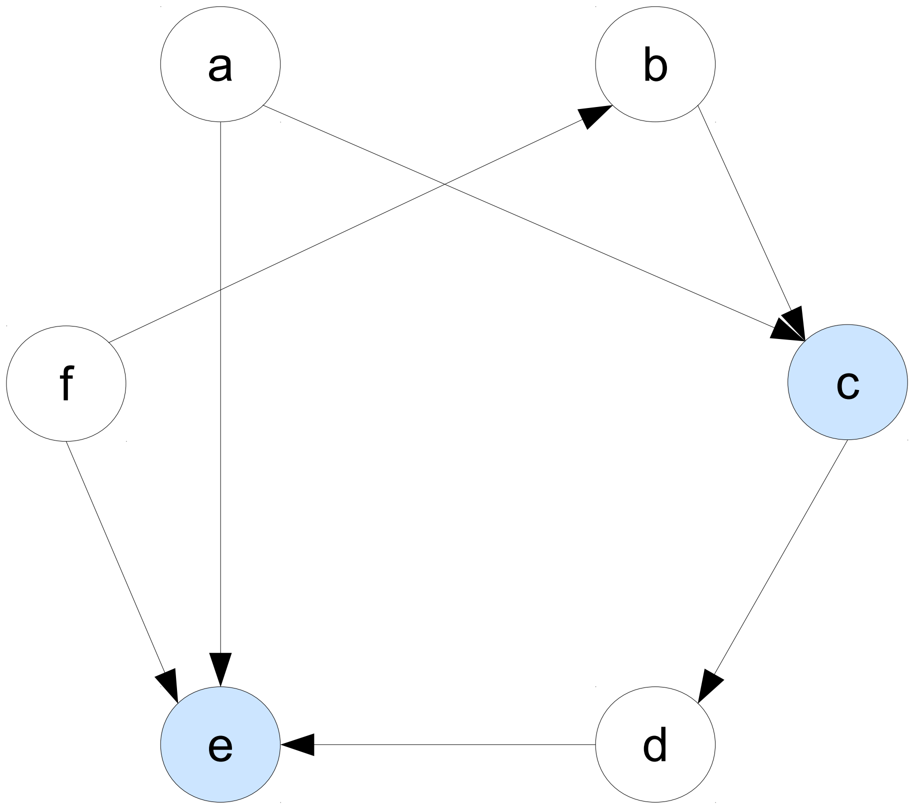
- Conditional independence of two random variables

$$p(a, b|c) = p(a|c)\,p(b|c)$$

- The previous observation that with less links the easier is the inference is equivalent to increasing the number of conditionally independent variables

Posterior distribution

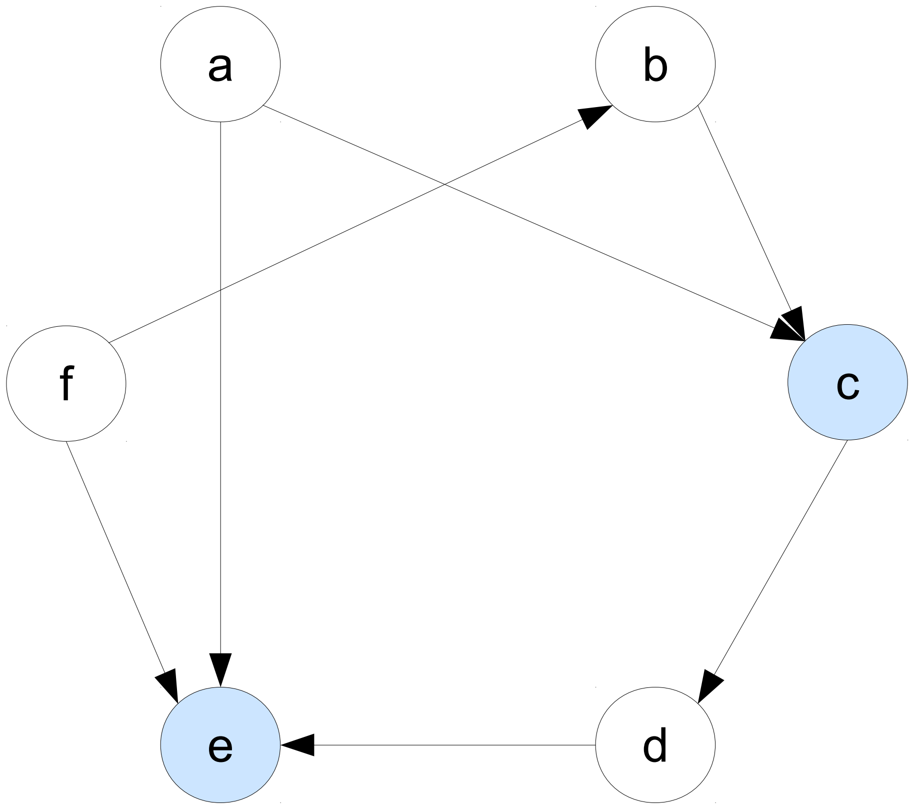
- We are not much interested in joint distribution
- More often we want to know posteriors
  - for some of the random variables given some observed data



Posterior given the some variables

- Joint a, b, d, f given c, e

$$p(a, b, d, f | c, e) = ?$$



Posterior given the some variables

- Joint prob. of a, b, d, f given c, e

$$p(a,b,d,f|c,e) = \frac{p(a,b,c,d,e,f)}{p(c,e)}$$

$$= \frac{p(a,b,c,d,e,f)}{\sum_{a,b,d,f} p(a,b,c,d,e,f)}$$

Posterior given the known variables

- Joint prob. of a, b, d, f given c = C, e = E

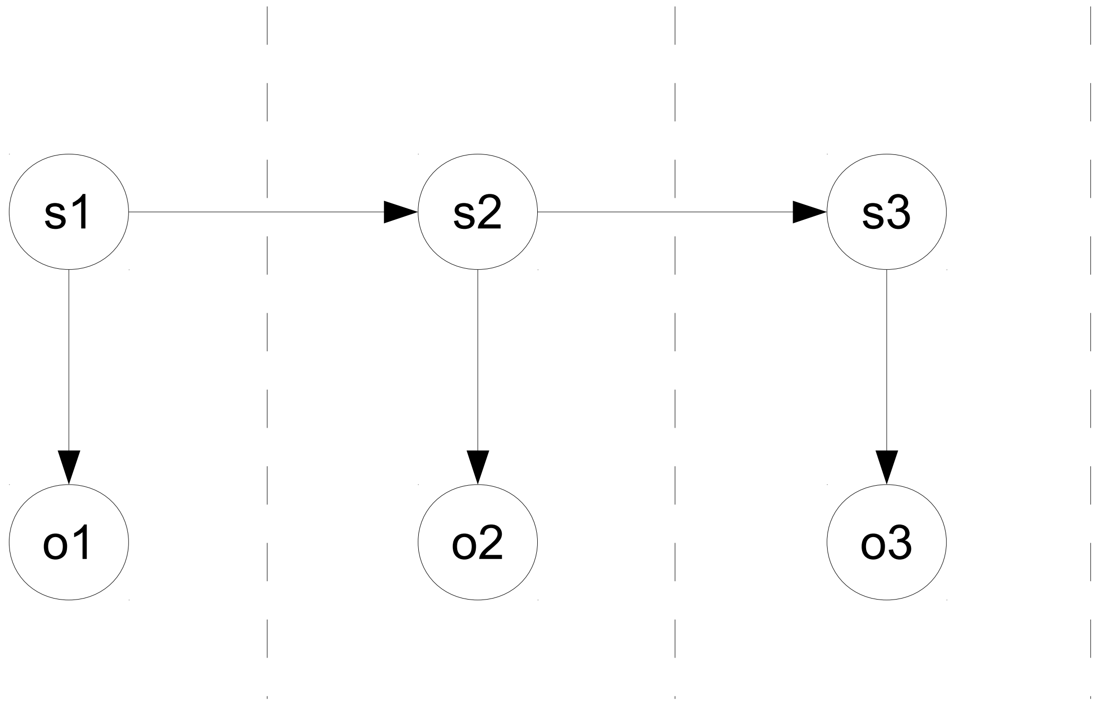
$$p(a,b,d,f|c=C,e=E)=\frac{p(a,b,c=C,d,e=E,f)}{p(c=C,e=E)}$$

$$=\frac{p(a,b,c=C,d,e=E,f)}{\sum_{a,b,d,f} p(a,b,c=C,d,e=E,f)}$$

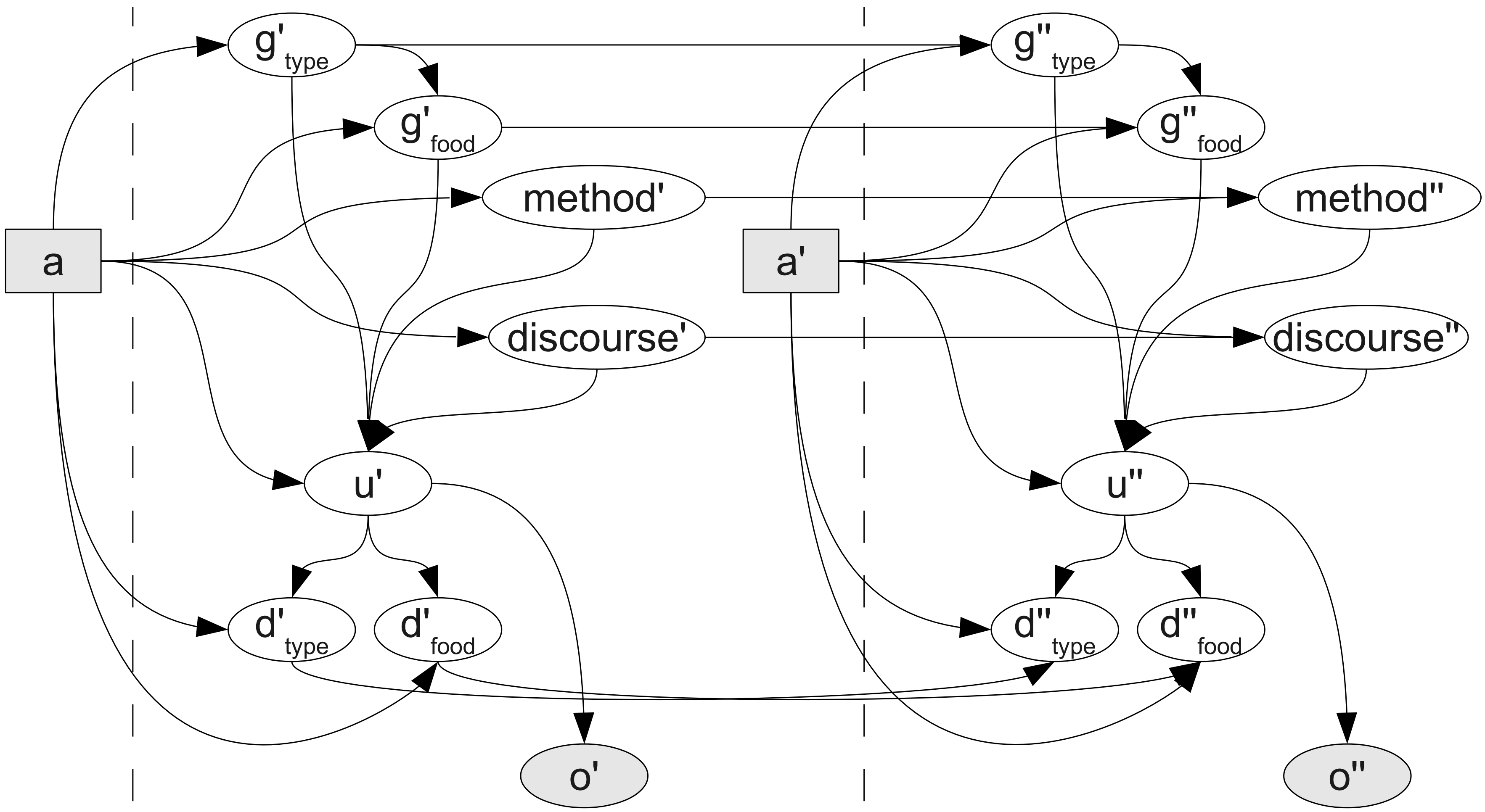
- The problem is not in the posterior itself
- The problem is in computing the normalisation constant

Dynamic Bayesian Networks

- Like a Bayesian network
- However, it can grow.

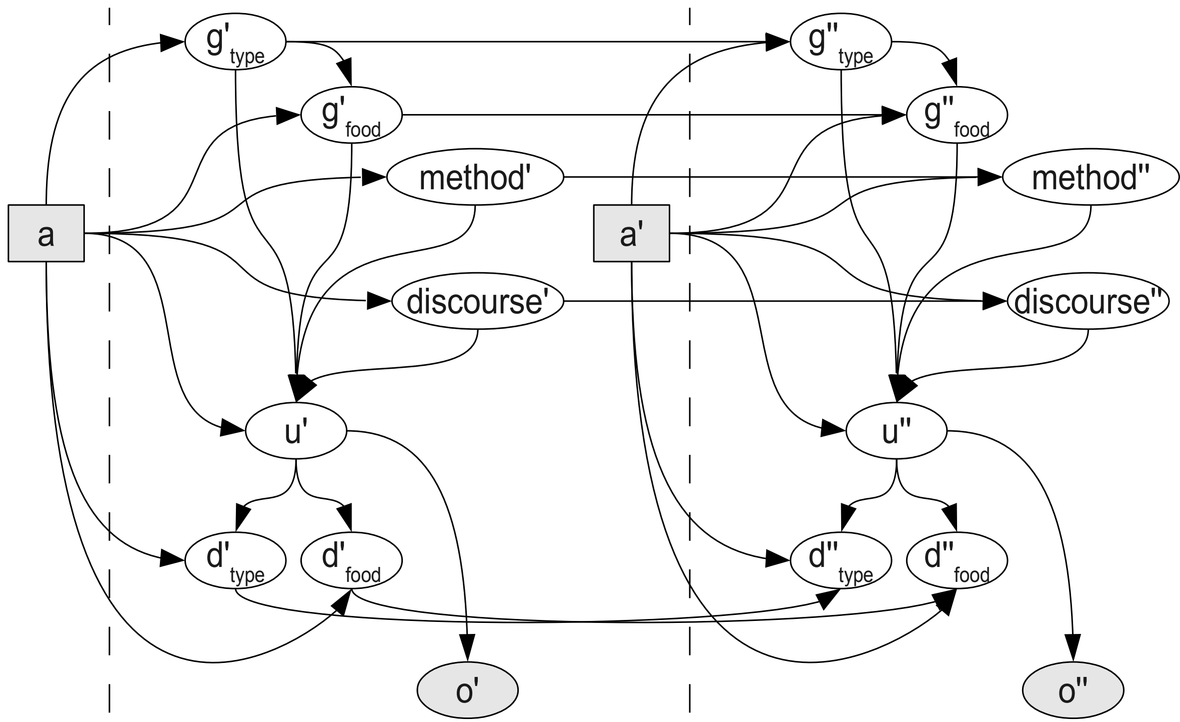


Belief monitoring as DBN



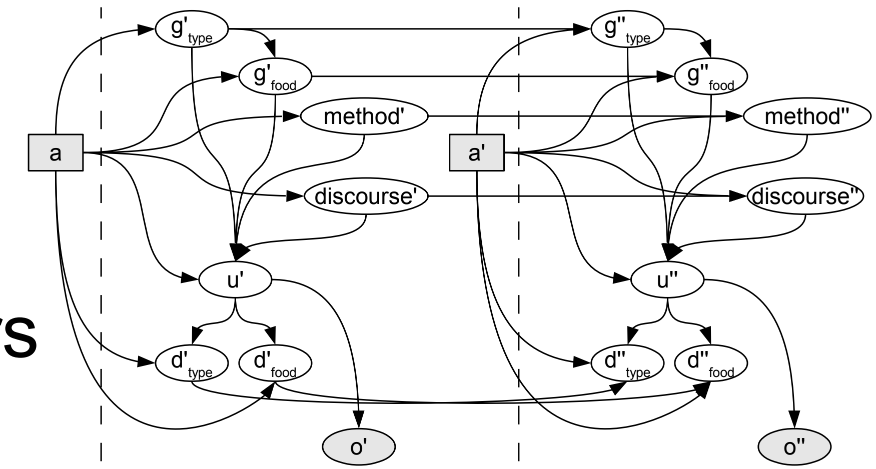
Inference in SDS

- Most of the time, we are interested marginal distributions, e.g.:
  - p(g"_type | a', …)
  - p(g"_food | a', …)
  - p(d"_type | a', …)
  - …



Inference in SDS

- Exact inference is intractable
  - Approximation techniques are necessary
- Loopy belief propagation
  - Infers the marginal distribution for the nodes
- Expectation propagation
  - Also infers parameters
  - Maximise the likelihood of the dialogue model parameters



Inference in Bayesian networks

- Simple marginalisation is inefficient
- Use dynamic programming

- Belief propagation
  - using dynamic programming
  - exact on trees
  - equivalent to Forward-Bacward algorithm for HMMs

- If used on networks with cycles then it is inexact
  - can be used iteratively → Loopy Belief Propagation
  - it converges to some local optimum
  - most of the time it works

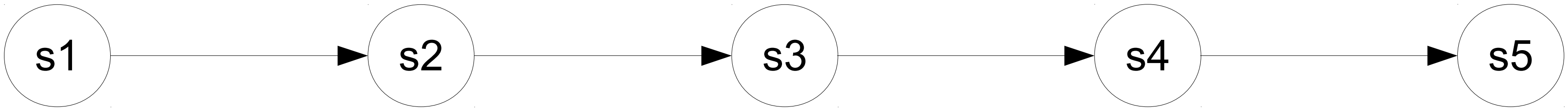
Belief propagation on a chain

- Compute p(s5) from p(s1,s2,s3,s4,s5)

$$p(s_5) = \sum_{s_1, s_2, s_3, s_4} p(s_1, s_2, s_3, s_4, s_5)$$

$$= \sum_{s_4} p(s_5 | s_4) \sum_{s_3} p(s_4 | s_3) \sum_{s_2} p(s_3 | s_2) \sum_{s_1} p(s_2 | s_1)\, p(s_1)$$

- Use dynamic programming
  - aka message passing algorithm



Forward message passing

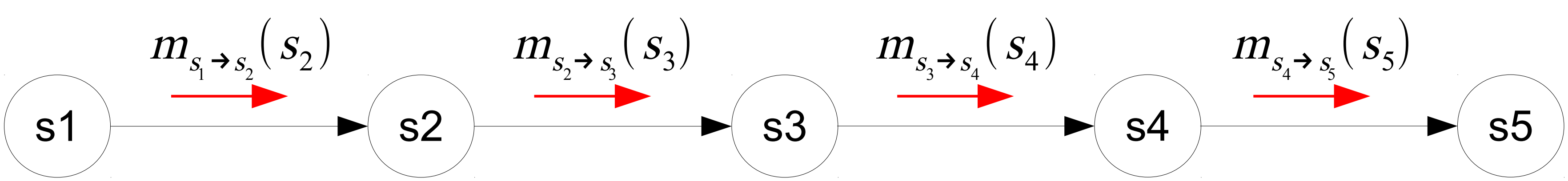
$$p(s_5) = \sum_{s_4} p(s_5|s_4) \sum_{s_3} p(s_4|s_3) \sum_{s_2} p(s_3|s_2) \sum_{s_1} p(s_2|s_1)\, p(s_1)$$

$$p(s_5) = \sum_{s_4} p(s_5|s_4) \sum_{s_3} p(s_4|s_3) \sum_{s_2} p(s_3|s_2)\, m_{s_1 \to s_2}(s_2)$$

$$p(s_5) = \sum_{s_4} p(s_5|s_4) \sum_{s_3} p(s_4|s_3)\, m_{s_2 \to s_3}(s_3)$$

$$p(s_5) = \sum_{s_4} p(s_5|s_4)\, m_{s_3 \to s_4}(s_4)$$

$$p(s_5) = m_{s_4 \to s_5}(s_5)$$



Backward message passing

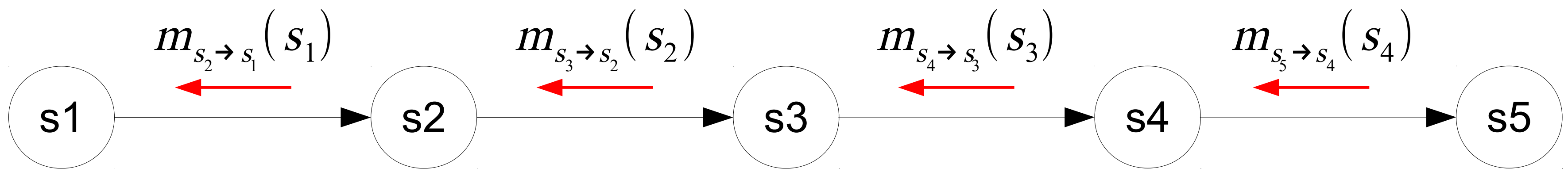
$$p(s_1)=\sum_{s_2} p(s_2|s_1)\,p(s_1)\sum_{s_3} p(s_3|s_2)\sum_{s_4} p(s_4|s_3)\sum_{s_5} p(s_5|s_4)$$

$$p(s_1)=\sum_{s_2} p(s_2|s_1)\,p(s_1)\sum_{s_3} p(s_3|s_2)\sum_{s_4} p(s_4|s_3)\,m_{s_5\rightarrow s_4}(s_4)$$

$$p(s_1)=\sum_{s_2} p(s_2|s_1)\,p(s_1)\sum_{s_3} p(s_3|s_2)\,m_{s_4\rightarrow s_3}(s_3)$$

$$p(s_1)=\sum_{s_2} p(s_2|s_1)\,p(s_1)\,m_{s_3\rightarrow s_2}(s_2)$$

$$p(s_1)=m_{s_2\rightarrow s_1}(s_1)$$



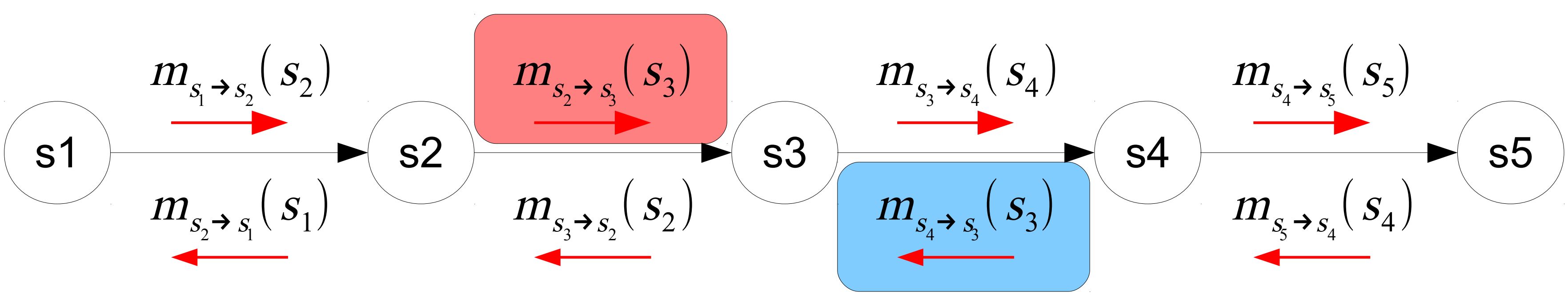
Message passing

$$p(s_3=S_3)=\sum_{s_5} p(s_5|s_4)\sum_{s_4} p(s_4|s_3=S_3)\sum_{s_2} p(s_3=S_3|s_2)\sum_{s_1} p(s_2|s_1)\,p(s_1)$$

$$p(s_3=S_3)=\sum_{s_5} p(s_5|s_4)\sum_{s_4} p(s_4|s_3=S_3)\,m_{s_2\to s_3}(s_3=S_3)$$

$$p(s_3=S_3)=m_{s_2\to s_3}(s_3=S_3)\sum_{s_4} p(s_4|s_3=S_3)\sum_{s_5} p(s_5|s_4)$$

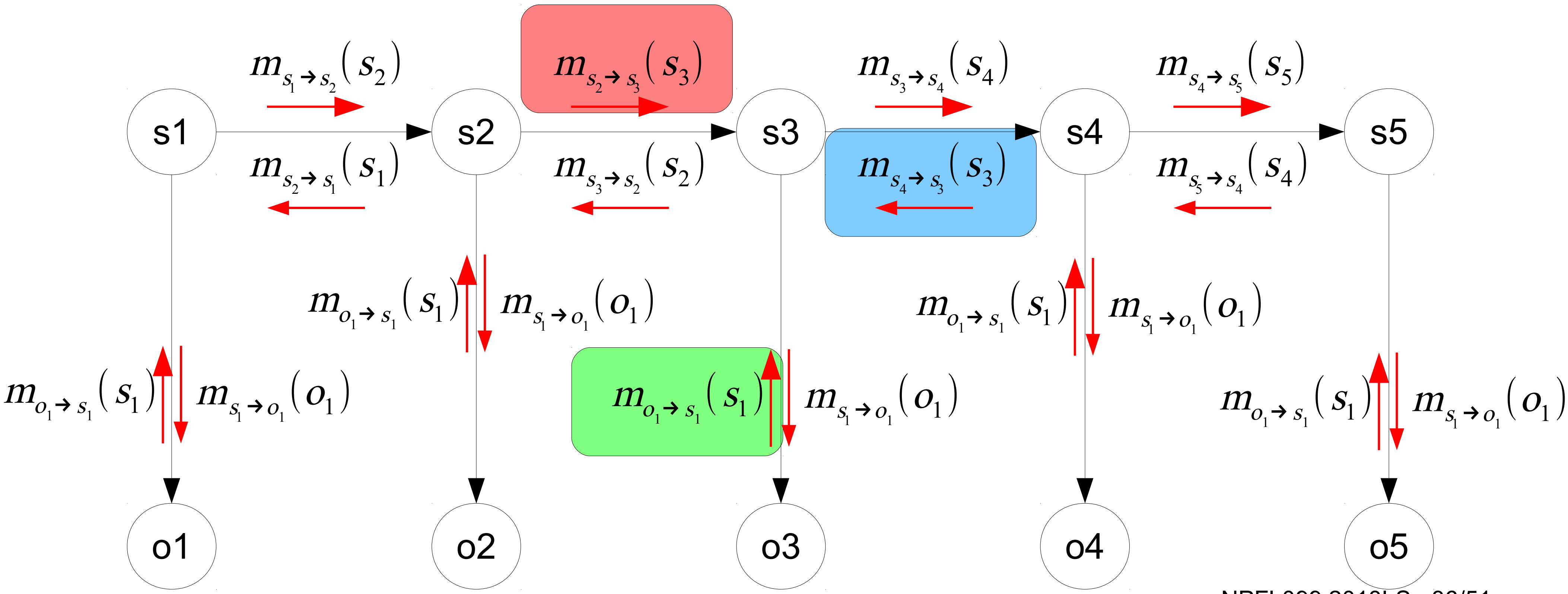
$$p(s_3=S_3)=m_{s_2\to s_3}(s_3=S_3)\,m_{s_4\to s_3}(s_3=S_3)$$



Belief propagation on a tree

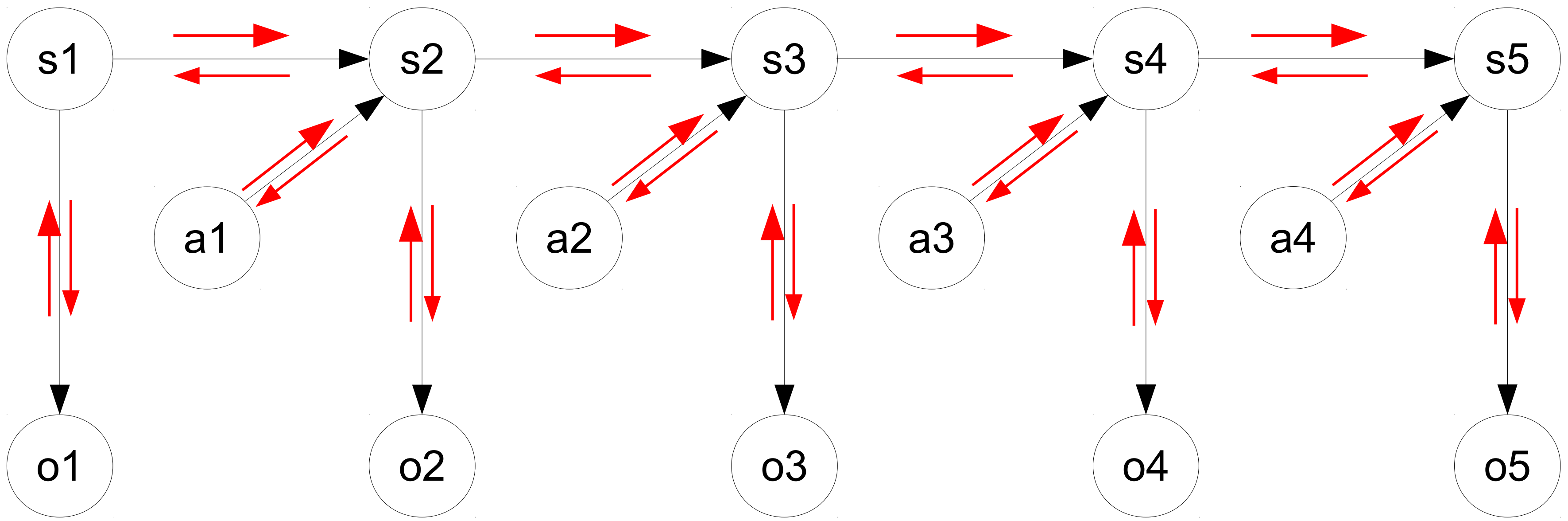
- We are interested in p(s3 = S3)

$$p(s_3 = S_3) = m_{s_2 \to s_3}(s_3 = S_3)\, m_{o_3 \to s_3}(s_3 = S_3)\, m_{s_4 \to s_3}(s_3 = S_3)$$



Belief propagation on a tree

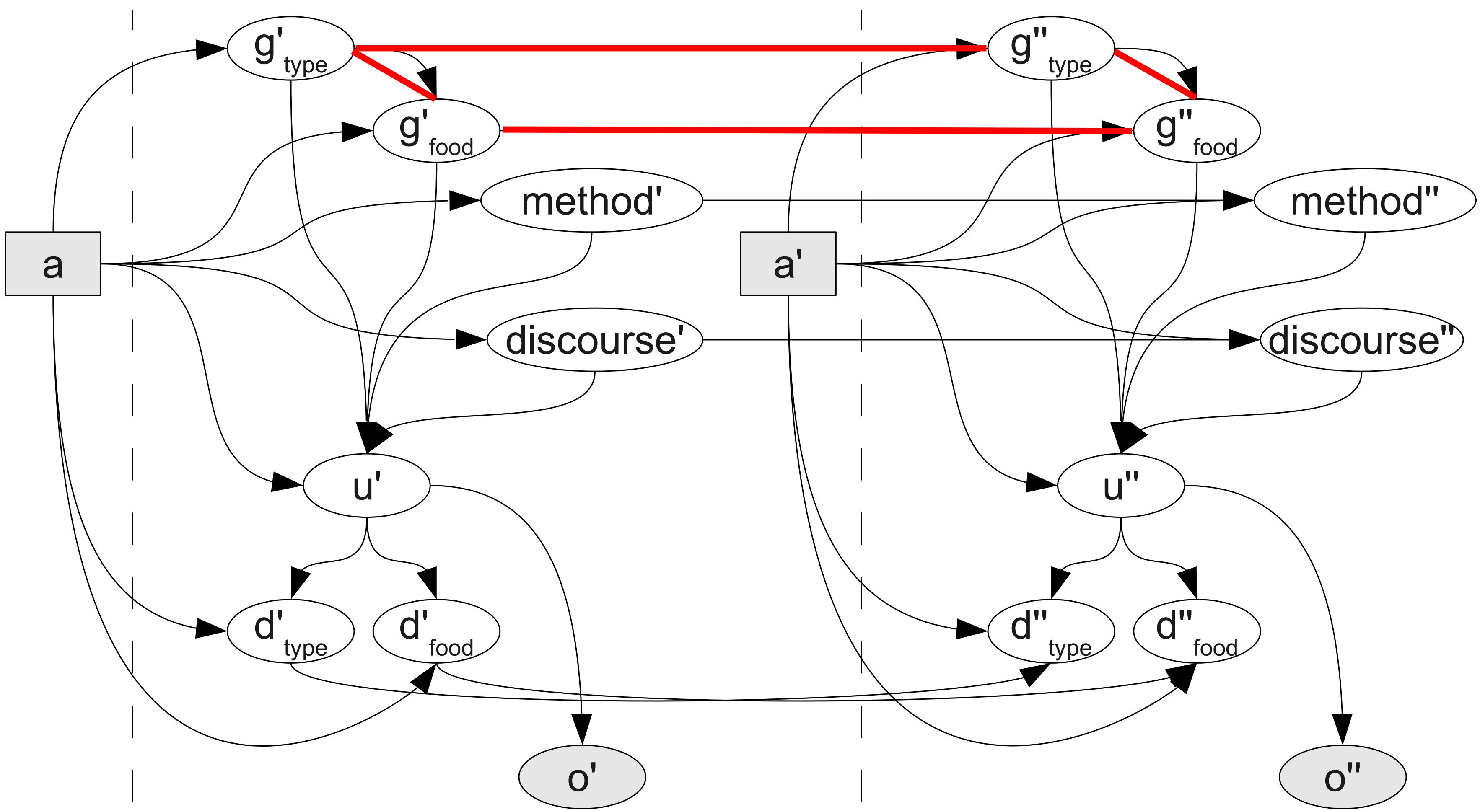
- The same algorithm scales to an arbitrary tree



- To compute marginals, compute messages first
- After one forward and backward sweep, all marginals can be computed at once

BP on a factored dialogue state

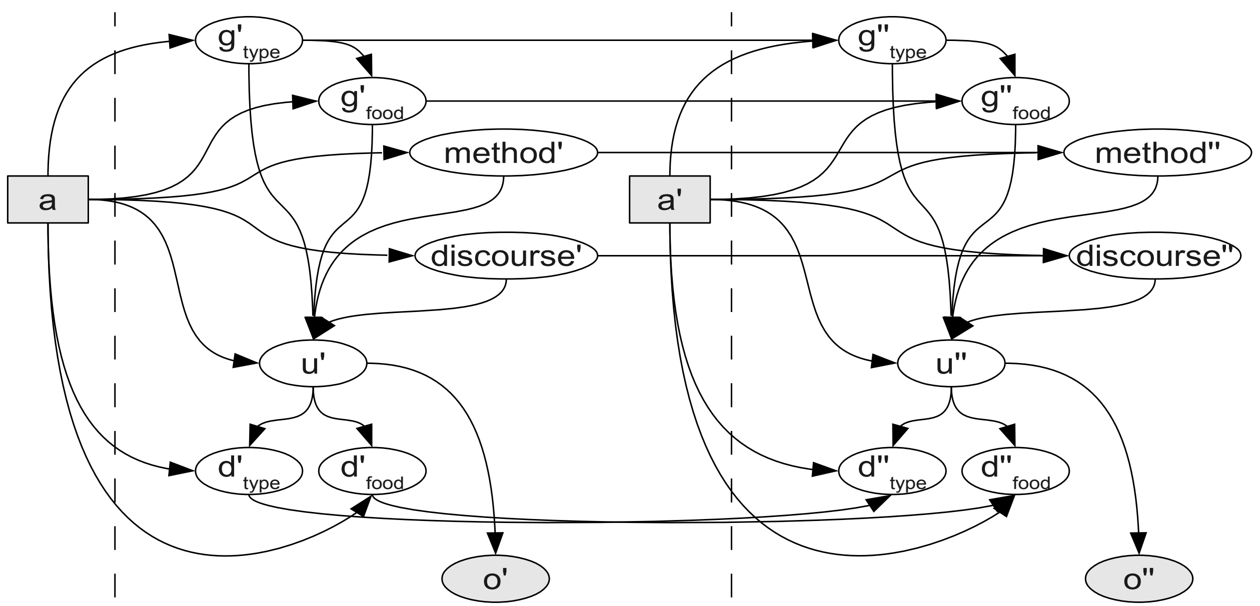
- It is not a tree any more



BP on a factored dialogue state

- Although not exact, perform belief propagation
- Iterate until convergence
  - there are multiple ways how the iterate

→ Loopy belief propagation



Thank you!

Filip Jurčíček

Institute of Formal and Applied Linguistics
Charles University in Prague
Czech Republic

Home page: http://ufal.mff.cuni.cz/~jurcicek

Approximation

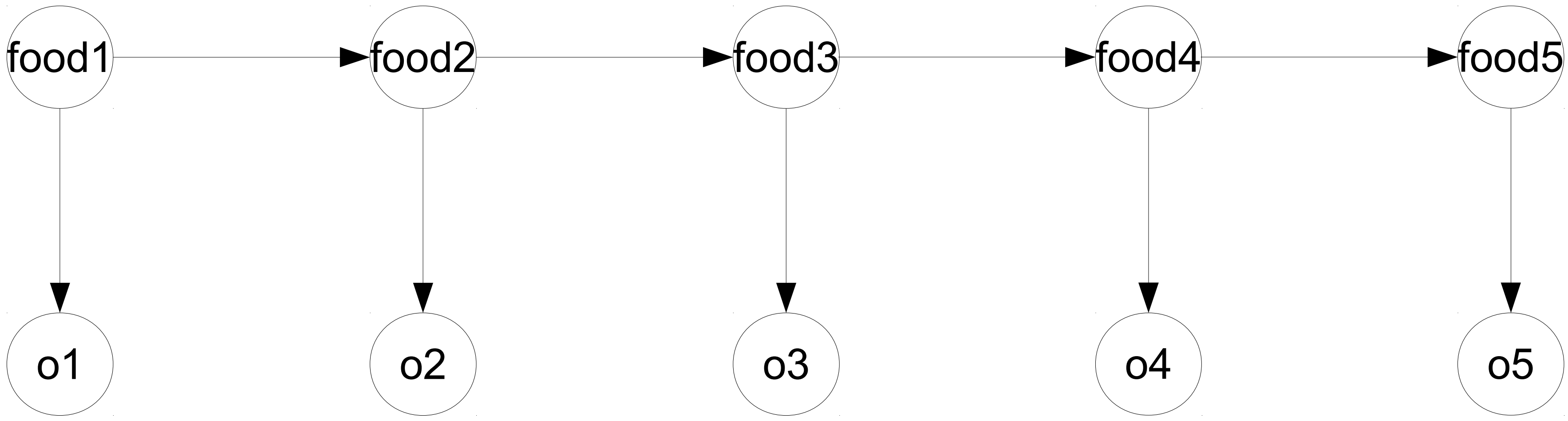
- Although BP or LBP significantly reduces the computational complexity, it is not enough
- Approximations
  - Grouped belief propagation
    - Enumerate only the values supported by the observations
  - Constant change transition probabilities
    - Some probabilities can be computed as a complement of others

Belief propagation example

- Assume a simple dialogue model only with one node: **food**
  - Having N values: Italian, Chinese, English, ...
  - Transition probability for the food node

$$p(food_{t+1} | food_t) = p_{food_{t+1}, food_t}$$

- We need N*N parameters

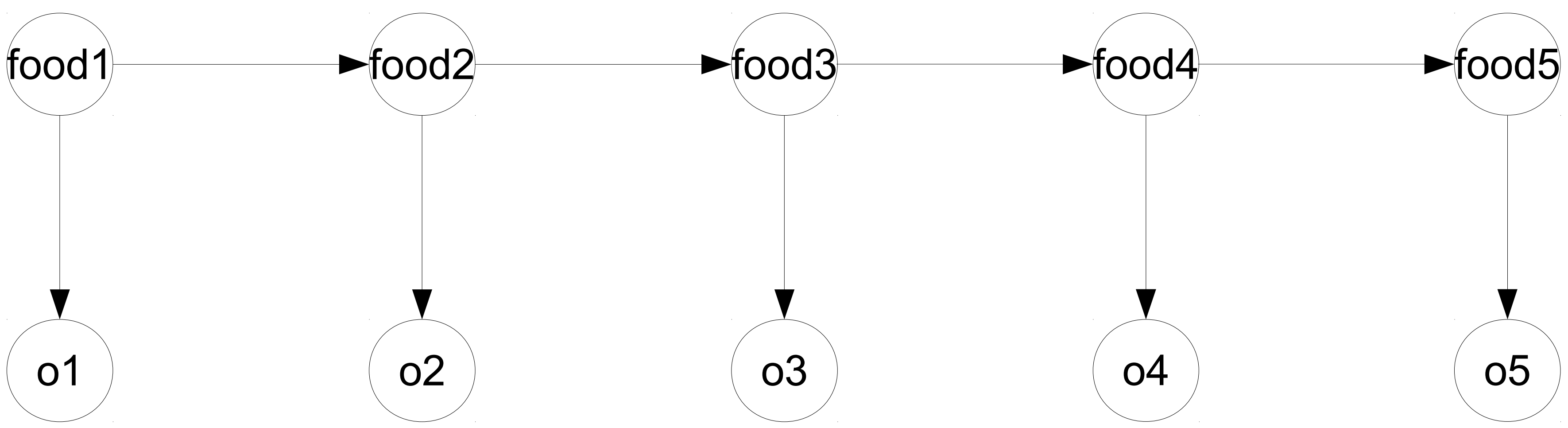


Grouped belief propagation

- Message to the food_{t+1} node is

$$m_{food_t \to food_{t+1}}(food_{t+1}) = \sum_{food_t} p(food_{t+1} | food_t)$$
$$m_{food_{t-1} \to food_t}(food_t)$$
$$m_{o_t \to food_t}(food_t)$$

- For every value of the food node, we have to sum over N values



Grouped belief propagation

- This can be greatly simplified

- At the beginning, we do not have evidence that probabilities of some values differ

$$m_{food_t \to food_{t+1}}(food_{t+1}) = \sum_{food_t} p(food_{t+1} | food_t)$$

$$m_{food_{t-1} \to food_t}(food_t) \quad m_{o_t \to food_t}(food_t) = const.$$

or equal to 0
in some models

$$m_{food_t \to food_{t+1}}(food_{t+1}) = m_{food_{t-1} \to food_t}(food_t) \; m_{o_t \to food_t}(food_t) \sum_{food_t} p(food_{t+1} | food_t)$$

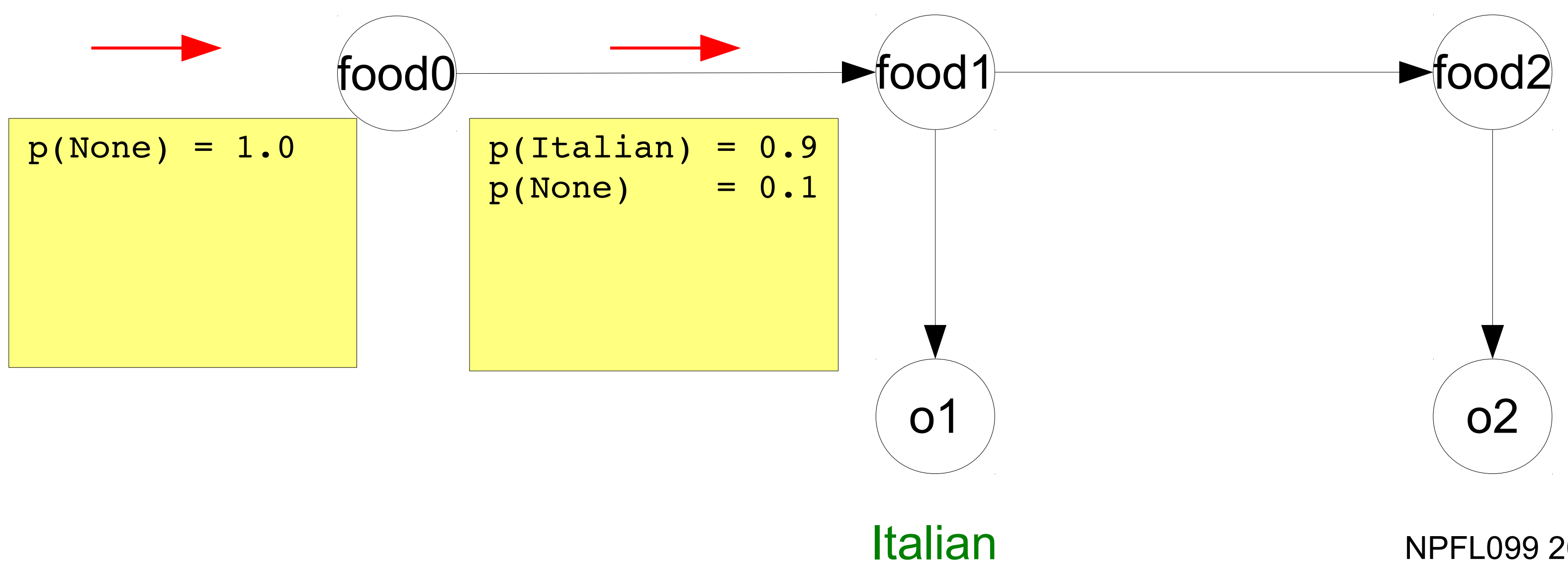
Grouped belief propagation

- To implement this
  - We have special node value N = None
  - We do not enumerate values with 0 prob.

Grouped belief propagation

$$m_{food_0 \rightarrow food_1}(food_1 = Italian) = p(food_1 = Italian | food_0 = N)\, m(food_0 = N)$$

$$m_{food_0 \rightarrow food_1}(food_1 = N) = 1 - m_{food_0 \rightarrow food_1}(food_1 = Italian)$$

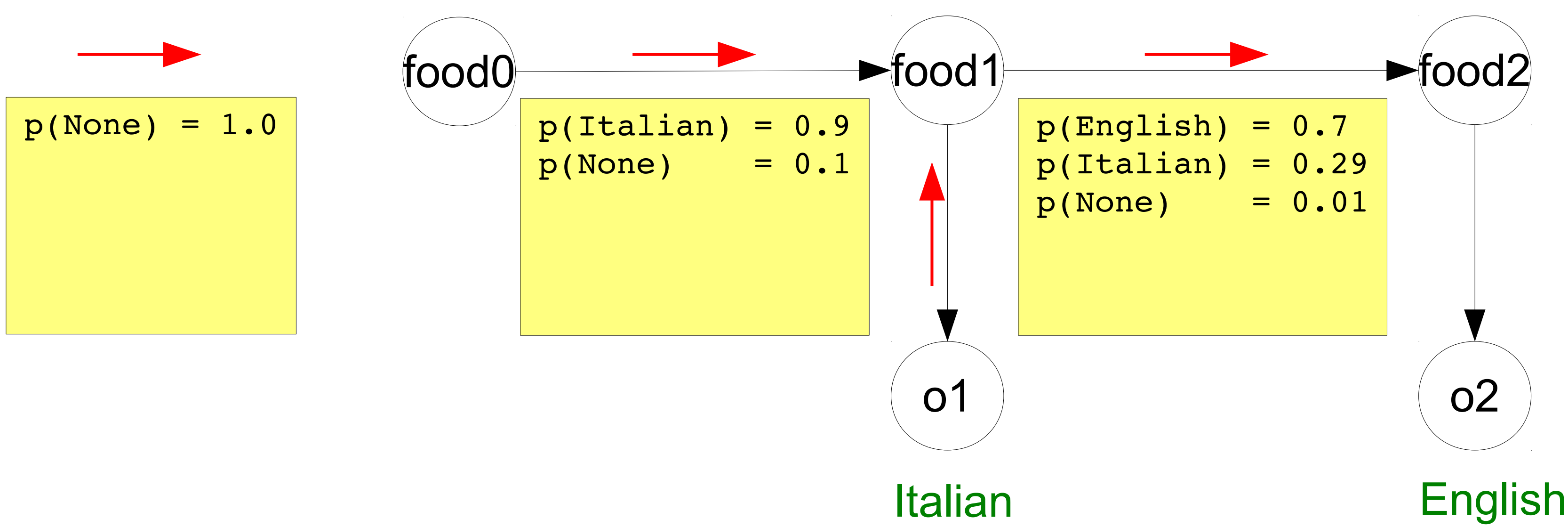


Grouped belief propagation

$$m_{food_1 \to food_2}\, p(\, food_2 = Italian) = \sum_{food = Italian\,, English\,, N} p(\, food_2 = Italian |\, food_2 = food)\; m_{\neg food_2}(\, food_1 = food)$$

$$m_{food_1 \to food_2}(\, food_2 = English) = \sum_{food = Italian\,, English,\, N} p(\, food_2 = English |\, food_1 = food)\; m_{\neg\, food_2}(\, food_1 = food)$$

$$m_{food_1 \to food_2}(\, food_2 = N) = 1 - \sum_{food = Itaian\,, English} m_{food_0 \to food_1}(\, food_2 = food)$$



Constant change transition probs

- Assume this prob. distribution for the following example

$$p(food_{t+1} | food_t) = \theta_1 \; \forall \; food_{t+1} = food_t$$

$$p(food_{t+1} | food_t) = \theta_2 \; \forall \; food_{t+1} \neq food_t$$

- Then the following can be simplified

$$m_{food_1 \rightarrow food_2} p(food_2 = Italian) = \sum_{food = Italian, English, N} p(food_2 = Italian | food_2 = food) \; m_{\neg food_2}(food_1 = food)$$

Constant change transition probs

$$m_{food_1 \rightarrow food_2} p(food_2 = Italian) = \sum_{food=Italian, English, N} p(food_2 = Italian | food_2 = food) \, m_{\neg food_2}(food_1 = food)$$

- Expand the sum

$$m_{food_1 \rightarrow food_2} p(food_2 = Italian) = . \; p(food_2 = Italian | food_2 = Italian) \, m_{\neg food_2}(food_1 = Italian) \; \sum_{food=English, N} p(food_2 = Italian | food_2 = food) \, m_{\neg food_2}(food_1 = food)$$

- Factor out the constant change transition prob.

$$m_{food_1 \rightarrow food_2} p(food_2 = Italian) = . \; p(food_2 = Italian | food_2 = Italian) \, m_{\neg food_2}(food_1 = Italian) \; \theta_2 \sum_{food=English, N} m_{\neg food_2}(food_1 = food)$$

Constant change transition probs

$$m_{food_1 \rightarrow food_2} p(food_2 = Italian) = .$$

$$p(food_2 = Italian | food_2 = Italian)\, m_{\neg food_2}(food_1 = Italian)$$

$$\theta_2 \sum_{food = English, N} m_{\neg food_2}(food_1 = food)$$

- Replace the sum by a complement

$$m_{food_1 \rightarrow food_2} p(food_2 = Italian) = .$$

$$p(food_2 = Italian | food_2 = Italian)\, m_{\neg food_2}(food_1 = Italian)$$

$$\theta_2 (1 - m_{\neg food_2}(food_1 = Italian))$$

Summary

- Talked about
  - Grouping similar states (HIS)
  - Factoring the dialogue states (BUDS)
- Loopy Belief Propagation on general graphs
- Grouping values in nodes
- Constant change transition probabilities