Bayesian Models for Sequences

- the world is dynamic
  - old information becomes obsolete
  - new information is available
  - the decisions an agent takes need to reflect these changes
- the dynamics of the world can be captured by means of state-based models

Bayesian Models for Sequences

- changes in the world are modelled as transitions between subsequent states
- state transitions can be
  - clocked, e.g.
    - speech: every 10 ms
    - vision: every 40 ms
    - stock market trends: every 24 hours
  - triggered by external events
    - language: every other word
    - travel planning: potential transfer points

Bayesian Models for Sequences

- main purpose:
  - predicting the probability of the next event
  - computing the probability of a (sub-)sequence
- important application areas:
  - speech and language processing, genome analysis, time series predictions (stock market, natural desasters, ...)

Markov chain

- Markov chain: special sort of belief network for sequential observations

$$S_0 \longrightarrow S_1 \longrightarrow S_2 \longrightarrow S_3 \longrightarrow S_4$$

- Thus, $P(S_{t+1}|S_0, \ldots, S_t) = P(S_{t+1}|S_t)$.
- Intuitively S_t conveys all of the information about the history that can affect the future states.
- "The past is independent of the future given the present."

Stationary Markov chain

- A stationary Markov chain is when for all $t > 0$, $t' > 0$, $P(S_{t+1}|S_t) = P(S_{t'+1}|S_{t'})$.
- Under this condition the network consists of two different kinds of slices
  - for the initial state without previous nodes (parents) we specify $P(S_0)$
  - for all the following states we specify $P(S_t|S_{t-1})$
- Simple model, easy to specify
- Often a highly natural model

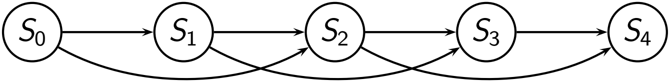
Stationary Markov chain

- The network can be extended indefinitely:
  - it is "rolled out" over the full length of the observation sequence
- rolling out the network can be done on demand (incrementally)
  - the length of the observation sequence need not be known in advance

Higher-order Markov Models

- modelling dependencies of various lengths
- bigrams

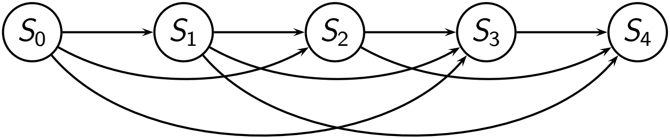
- trigrams



  - three different time slices have to modelled
    - for S_0: $P(S_0)$
    - for S_1: $P(S_1|S_0)$
    - for all others: $P(S_i)|S_{i-2}S_{i-1})$

Higher-order Markov Models

- quadrograms: $P(S_i | S_{i-3} S_{i-2} S_{i-1})$



- four different kinds of time slices required

Markov Models

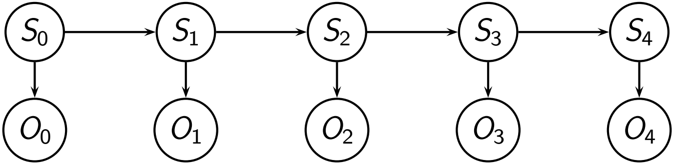
- examples of Markov chains for German letter sequences
- unigrams:
  aiobnin*tarsfneonlpiitdregedcoa*ds*e*dbieastnreleeucdkeaitb*
  dnurlarsls*omn*keu**svdleeoieei* ...
- bigrams:
  er*agepteprteiningeit*gerelen*re*unk*ves*mterone*hin*d*an*
  nzerurbom* ...
- trigrams:
  billunten*zugen*die*hin*se*sch*wel*war*gen*man*
  nicheleblant*diertunderstim* ...
- quadrograms:
  eist*des*nich*in*den*plassen*kann*tragen*was*wiese*
  zufahr* ...

Hidden Markov Model

- Often the observation does not deterministically depend on the state of the model
- This can be captured by a Hidden Markov Model (HMM)
- ... even if the state transitions are not directly observable

Hidden Markov Model

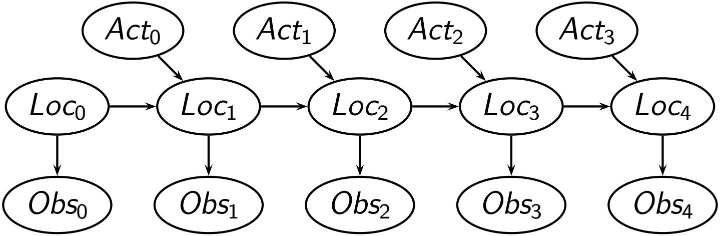
- A HMM is a belief network where states and observations are separated



- $P(S_0)$ specifies initial conditions
- $P(S_{t+1}|S_t)$ specifies the dynamics
- $P(O_t|S_t)$ specifies the sensor model

Example (1): robot localization

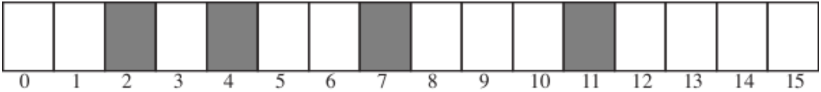
- Suppose a robot wants to determine its location based on its actions and its sensor readings: Localization
- This can be represented by the augmented HMM:



- Combining two kinds of uncertainty:
  - The location depends probabilistically on the robot's action
  - The sensor data are noisy

Example localization domain

- Circular corridor, with 16 locations:



- Doors at positions: 2, 4, 7, 11.
- Robot starts at an unknown location and must determine where it is.

Example Sensor Model

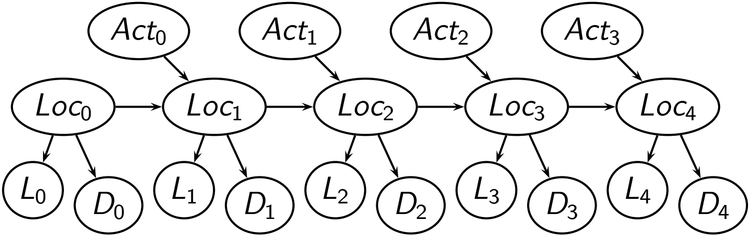
- $P(Observe\ Door \mid At\ Door) = 0.8$
- $P(Observe\ Door \mid Not\ At\ Door) = 0.1$

Example Dynamics Model

- $P(loc_t = L | action_{t-1} = goRight \wedge loc_{t-1} = L) = 0.1$
- $P(loc_t = L + 1 | action_{t-1} = goRight \wedge loc_{t-1} = L) = 0.8$
- $P(loc_t = L + 2 | action_{t-1} = goRight \wedge loc_{t-1} = L) = 0.074$
- $P(loc_t = L' | action_{t-1} = goRight \wedge loc_{t-1} = L) = 0.002$ for any other location L'.
  - All location arithmetic is modulo 16.
  - The action *goLeft* works the same but to the left.

Combining sensor information

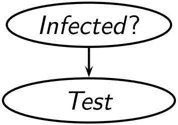
- the robot can have many (noisy) sensors for signals from the environment
- e.g. a light sensor in addition to the door sensor
- Sensor Fusion: combining information from different sources



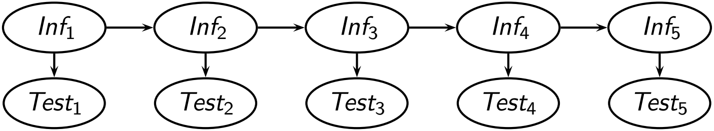
D_t door sensor value at time t
L_t light sensor value at time t

Hidden Markov Models

- Example (2): medical diagnosis (milk infection test) (Jensen and Nielsen 2007)
- the probability of
  the test outcome depends on the cow being infected or not

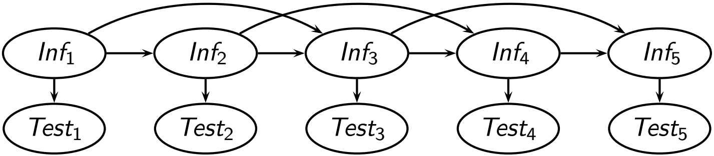


- the probability of the cow being infected depends on the cow being infected on the previous day
  - first order Markov model



Hidden Markov Models

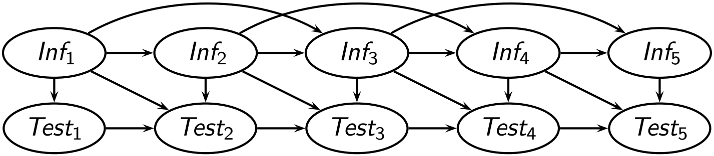
- the probability of the cow being infected depends on the cow being infected on the two previous days
  - incubation and infection periods of more than one day
  - second order Markov model



  - assumes only random test errors
- weaker independence assumptions
  - more powerful model
  - more data required for training

Hidden Markov Models

- the probability of the test outcome also depends on the cow's health and the test outcome on the previous day
  - can also capture systematic test errors
  - second order Markov model for the infection
  - first order Markov model for the test results



Hidden Markov Models

- Example (3): Tagging for Natural Language Processing
- annotating the word forms in a sentence with

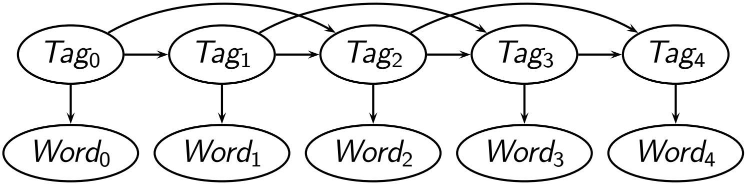
part-of-speech information *Yesterday*$_{RB}$ *the*$_{DT}$ *school*$_{NNS}$ *was*$_{VBD}$ *closed*$_{VBN}$
topic areas: *He did some field work.* *field*$_{military}$, *field*$_{agriculture}$, *field*$_{physics}$, *field*$_{social\ sci.}$, *field*$_{optics}$, ...
semantic roles *The winner*$_{Beneficiary}$ *received the trophy*$_{Theme}$ *at the town hall*$_{Location}$

Hidden Markov Models

- sequence labelling problem
  - the label depends on the current state and the most recent history
- one-to-one correspondence between states, tags, and word forms

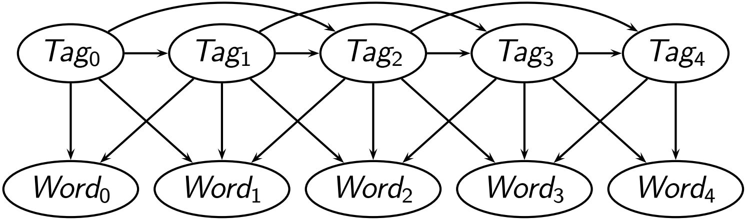
Hidden Markov Models

- causal (generative) model of the sentence generation process
  - tags are assigned to states
  - the underlying state (tag) sequence produces the observations (word forms)
- typical independence assumptions
  - trigram probabilities for the state transitions
  - word form probabilities depend only on the current state



Hidden Markov Model

- weaker independence assumption (stronger model):
  - the probability of a word form also depends on the previous and subsequent state



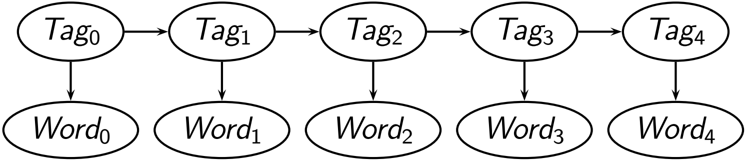
Two alternative graphical representations

- influence diagrams, belief networks, Bayesian networks, causal networks, graphical models, ...
- state transition diagrams (probabilistic finite state machines)

	Bayesian networks	State transition diagrams
state nodes	variables with states as values	states
edges into state nodes	causal influence and their probabilities	possible state transitions
# state nodes	# model states	length of the observation sequence
observation nodes	variables with observations as values	observation values
edges into observ. nodes	conditional probability tables	conditional probabilities

Two alternative graphical representations

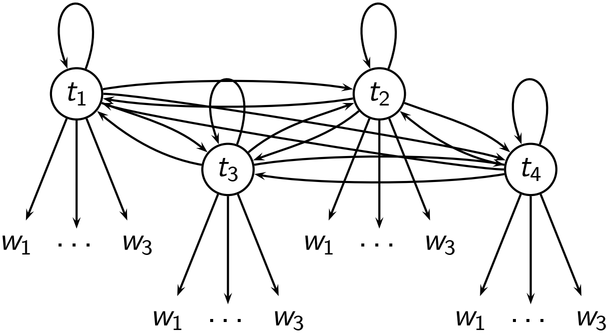
- Tagging as a Bayesian network



- possible state transitions are not directly visible
  - indirectly encoded in the conditional probability tables
- sometimes state transition diagrams are better suited to illustrate the model topology

Two alternative graphical representations

- Tagging as a state transition diagram (possible only for bigram models)



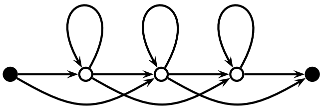
- ergodic model: full connectivity between all states

Hidden Markov Models

- Example (4): Speech Recognition, Swype gesture recognition
- observation subsequences of unknown length are mapped to one label
  $\rightarrow$ alignment problem
- full connectivity is not desired
- a phone/syllable/word realization cannot be reversed

Hidden Markov Models

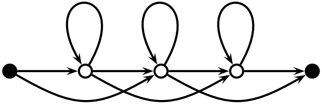
- possible model topologies for phones (only transitions depicted)



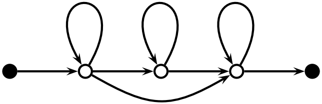
$P(1\|0)$	$P(1\|1)$	0	0	0
$P(2\|0)$	$P(2\|1)$	$P(2\|2)$	0	0
0	$P(3\|1)$	$P(3\|2)$	$P(3\|3)$	0
0	0	$P(4\|2)$	$P(4\|3)$	0

Hidden Markov Models

- possible model topologies for phones (only transitions depicted)



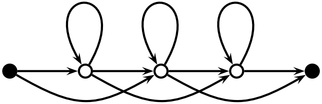
$$\begin{matrix} P(1|0) & P(1|1) & 0 & 0 & 0 \\ P(2|0) & P(2|1) & P(2|2) & 0 & 0 \\ 0 & P(3|1) & P(3|2) & P(3|3) & 0 \\ 0 & 0 & P(4|2) & P(4|3) & 0 \end{matrix}$$



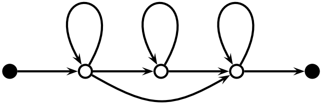
$$\begin{matrix} P(1|0) & P(1|1) & 0 & 0 & 0 \\ 0 & P(2|1) & P(2|2) & 0 & 0 \\ 0 & P(3|1) & P(3|2) & P(3|3) & 0 \\ 0 & 0 & 0 & P(4|3) & 0 \end{matrix}$$

Hidden Markov Models

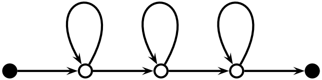
- possible model topologies for phones (only transitions depicted)



$$\begin{matrix} P(1|0) & P(1|1) & 0 & 0 & 0 \\ P(2|0) & P(2|1) & P(2|2) & 0 & 0 \\ 0 & P(3|1) & P(3|2) & P(3|3) & 0 \\ 0 & 0 & P(4|2) & P(4|3) & 0 \end{matrix}$$



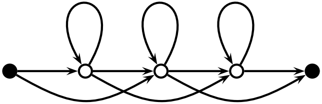
$$\begin{matrix} P(1|0) & P(1|1) & 0 & 0 & 0 \\ 0 & P(2|1) & P(2|2) & 0 & 0 \\ 0 & P(3|1) & P(3|2) & P(3|3) & 0 \\ 0 & 0 & 0 & P(4|3) & 0 \end{matrix}$$



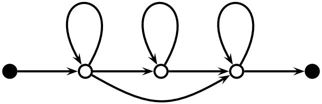
$$\begin{matrix} P(1|0) & P(1|1) & 0 & 0 & 0 \\ 0 & P(2|1) & P(2|2) & 0 & 0 \\ 0 & 0 & P(3|2) & P(3|3) & 0 \\ 0 & 0 & 0 & P(4|3) & 0 \end{matrix}$$

Hidden Markov Models

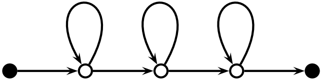
- possible model topologies for phones (only transitions depicted)



$$\begin{matrix} P(1|0) & P(1|1) & 0 & 0 & 0 \\ P(2|0) & P(2|1) & P(2|2) & 0 & 0 \\ 0 & P(3|1) & P(3|2) & P(3|3) & 0 \\ 0 & 0 & P(4|2) & P(4|3) & 0 \end{matrix}$$



$$\begin{matrix} P(1|0) & P(1|1) & 0 & 0 & 0 \\ 0 & P(2|1) & P(2|2) & 0 & 0 \\ 0 & P(3|1) & P(3|2) & P(3|3) & 0 \\ 0 & 0 & 0 & P(4|3) & 0 \end{matrix}$$

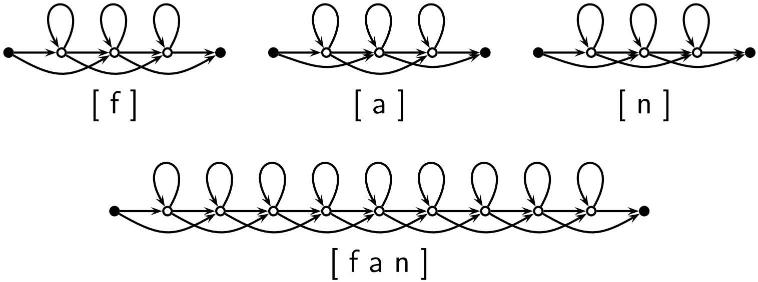


$$\begin{matrix} P(1|0) & P(1|1) & 0 & 0 & 0 \\ 0 & P(2|1) & P(2|2) & 0 & 0 \\ 0 & 0 & P(3|2) & P(3|3) & 0 \\ 0 & 0 & 0 & P(4|3) & 0 \end{matrix}$$

- the more data available the more sophisticated (and powerful) models can be trained

Hidden Markov Models

- composition of submodels on multiple levels
  - phone models have to be concatenated into word models
  - word models are concatenated into utterance models

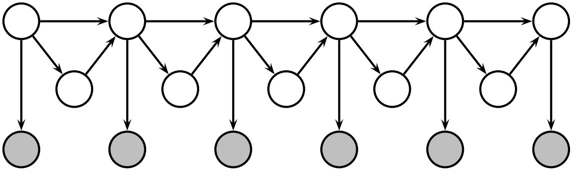


Dynamic Bayesian Networks

- using complex state descriptions, encoded by means of features
    - model can be in "different states" at the same time
- more efficient implementation of state transitions
- modelling of transitions between sub-models
- factoring out different influences on the outcome
    - interplay of several actuators (muscles, motors, ...)
- modelling partly asynchronized processes
    - coordinated movement of different body parts (e.g. sign language)
    - synchronization between speech sounds and lip movements
    - synchronization between speech and gesture
    - ...

Dynamic Bayesian Networks

- problem: state-transition probability tables are sparse
  - contain a large number of zero probabilities
- alternative model structure: separation of state and transition variables



deterministic state variables

stochastic transition variables

observation variables

- causal links can be stochastic *or* deterministic
  - stochastic: conditional probabilities to be estimated
  - deterministic: to be specified manually (decision trees)

Dynamic Bayesian Networks

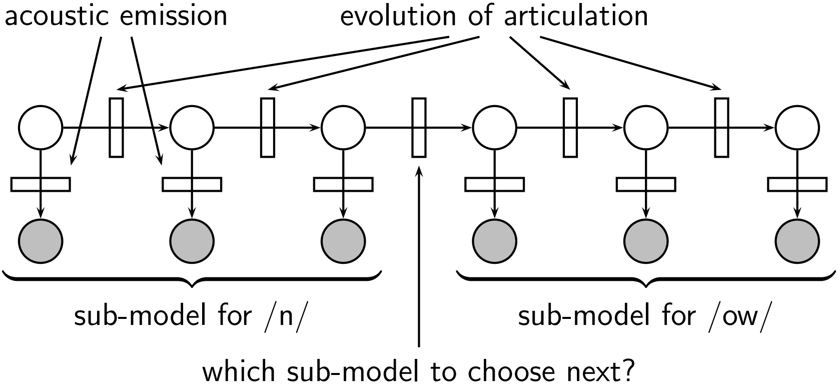
- state variables
  - distinct values for each state of the corresponding HMM
  - value at slice $t+1$ is a deterministic function of the state and the transition of slice t
- transition variables
  - probability distribution
  - which arc to take to leave a state of the corresponding HMM
  - number of values is the outdegree of the corresponding state in an HMM
- use of transition variables is more efficient than using stochastic state variables with zero probabilities for the impossible state transitions

Dynamic Bayesian Networks

- composite models: some applications require the model to be composed out of sub-models
  - speech: phones $\rightarrow$ syllables $\rightarrow$ words $\rightarrow$ utterances
  - vision: sub-parts $\rightarrow$ parts $\rightarrow$ composites
  - genomics: nucleotides $\rightarrow$ amino acids $\rightarrow$ proteins

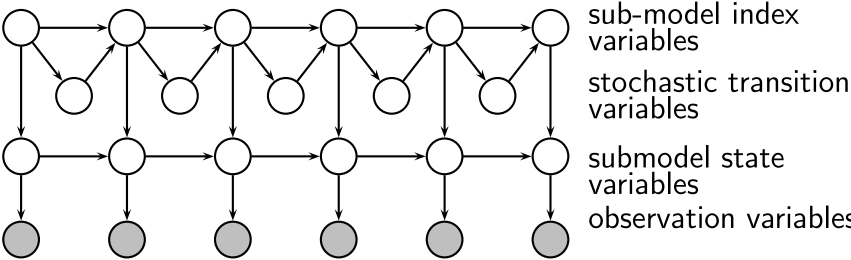
Dynamic Bayesian Networks

- composite models:
  - length of the sub-segments is not kown in advance
  - naive concatenation would require to generate all possible segmentations of the input sequence



Dynamic Bayesian Networks

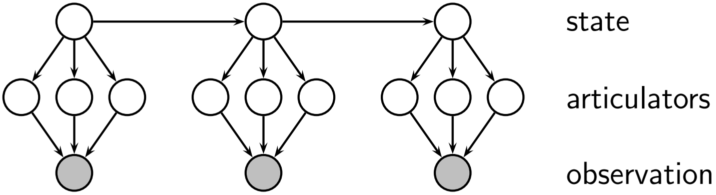
- additional sub-model variables select the next sub-model to choose



- sub-model index variables: which submodel to use at each point in time
- sub-model index and transition variables model legal sequences of sub-models (control layer)
- several control layers can be combined

Dynamic Bayesian Networks

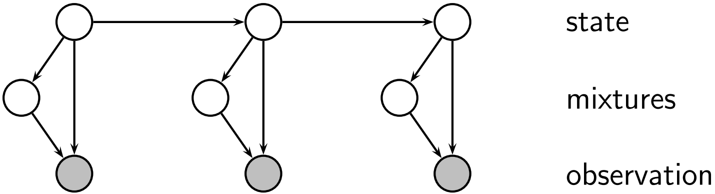
- factored models (1): factoring out different influences on the observation
- e.g. articulation:
  - asynchroneous movement of articulators (lips, tongue, jaw, ...)



- if the data is drawn from a factored source, full DBNs are superior to the special case of HMMs

Dynamic Bayesian Networks

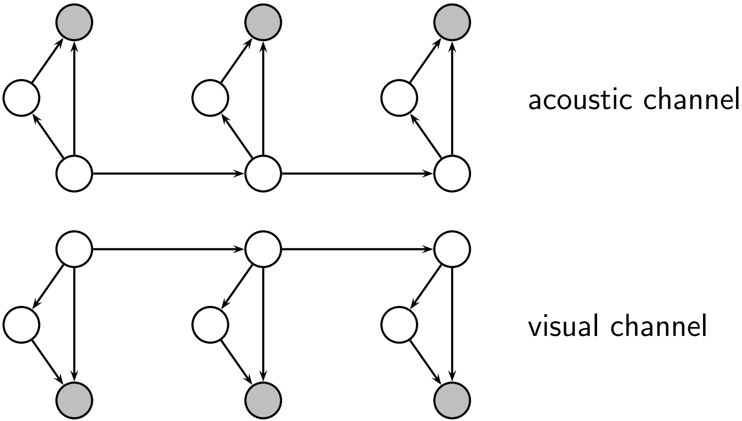
- factored models (2): coupling of different input channels
  - e.g. acoustic and visual information in speech processing
- naïve approach (1): data level fusion



- too strong synchronisation constraints

Dynamic Bayesian Networks

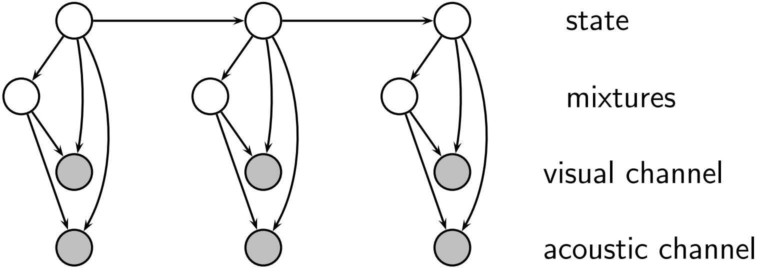
- naïve approach(2): independent input streams



- no synchronisation at all

Dynamic Bayesian Networks

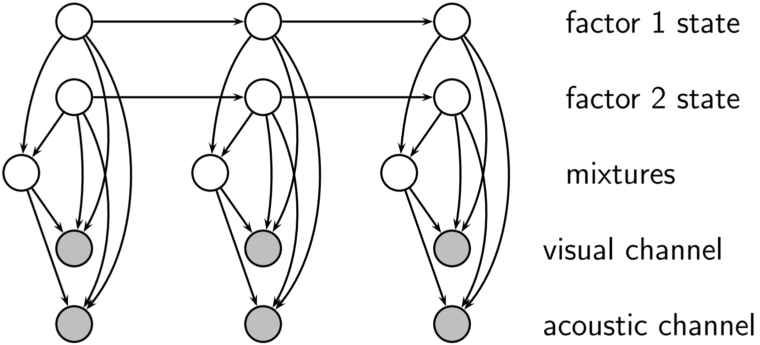
- product model



- state values are taken from the cross product of acoustic and visual states
- large probability distributions have to be trained

Dynamic Bayesian Networks

- factorial model (Nefian et al. 2002)



- independent (hidden) states
- indirect influence by means of the "explaining away" effect
- loose coupling of input channels

Dynamic Bayesian Networks

- inference is extremely expensive
  - nodes are connected across slides
  - domains are not locally restricted
  - cliques become intractably large
- but: joint distribution usually need not be computed
  - only maximum detection required
  - finding the optimal path through a lattice
  - dynamic programming can be applied (Viterbi algorithm)

Learning of Bayesian Networks

- estimating the probabilities for a given structure
  - for complete data:
    - maximum likelihood estimation
    - Bayesian estimation
  - for incomplete data
    - expectation maximization
    - gradient descent methods
- learning the network structure

Parameter estimation

- complete data
  - maximum likelihood estimation
  - Bayesian estimation
- incomplete data
  - expectation maximization
  - (gradient descent techniques)

Maximum Likelihood Estimation

- likelihood of the model M given the (training) data $\mathcal{D}$

$$L(M|\mathcal{D}) = \prod_{d \in \mathcal{D}} P(d|M)$$

- log-likelihood

$$LL(M|\mathcal{D}) = \sum_{d \in \mathcal{D}} log_2 P(d|M)$$

- choose among several possible models for describing the data according to the principle of maximum likelihood

$$\hat{\Theta} = \arg\max_{\Theta} L(M_{\Theta}|\mathcal{D}) = \arg\max_{\Theta} LL(M_{\Theta}|\mathcal{D})$$

- the models only differ in the set of parameters Θ

Maximum Likelihood Estimation

- complete data: estimating the parameters by counting

$$P(A = a) = \frac{N(A = a)}{\sum_{v \in dom(A)} N(A = v)}$$

$$P(A = a | B = b, C = c) = \frac{N(A = a, B = b, C = c)}{N(B = b, C = c)}$$

Rare events

- sparse data results in pessimistic estimations for unseen events
  - if the count for an event in the data base is 0, the event is considered impossible by the model
  - in many applications most events will never be observed, irrespective of the sample size

Rare events

- Bayesian estimation: using an estimate of the prior probability as starting point for the counting
  - estimation of maximum a posteriori parameters
  - no zero counts can occur
  - if nothing else available use an even distribution as prior
  - Bayesian estimate in the binary case with an even distribution

    $$P(yes) = \frac{n+1}{n+m+2}$$

    n: counts for yes, m: counts for no
  - effectively adding virtual counts to the estimate

Rare events

- alternative: smoothing as a post processing step
- remove probability mass from the frequent observations ...
- ... and distribute it to the not observed ones
  - floor method
  - discounting
  - ...

Incomplete Data

- missing at random:
  - probability that a value is missing depends only on the observed value
  - e.g. confirmation measurement: values are available only if the preceding measurement was positive/negative
- missing completely at random
  - probability that a value is missing is also independent of the value
  - e.g. stochastic failures of the measurement equipment
  - e.g. hidden/latent variables (mixture coefficients of a Gaussian mixture distribution)
- nonignorable:
  - neither MAR or MCAR
  - probability depends on the unseen values, e.g. exit polls for extremist parties

Expectation Maximization

- estimating the underlying distribution of not directly observable variables
- expectation:
  - "complete" the data set using the current estimation $h = \Theta$ to calculate expectations for the missing values
  - applies the model to be learned (Bayesian inference)
- maximization:
  - use the "completed" data set to find a new maximum likelihood estimation $h' = \Theta'$

Expectation Maximization

- full data consists of tuples $\langle x_{i1}, \ldots, x_{ik}, z_{i1}, \ldots, z_{il} \rangle$
  only x_i can be observed
- training data: $X = \{\vec{x}_1, \ldots, \vec{x}_m\}$
- hidden information: $Z = \{\vec{z}_1, \ldots, \vec{z}_m\}$
- parameters of the distribution to be estimated: Θ
- Z can be treated as random variable with $p(Z) = f(\Theta, X)$
- full data: $Y = \{\vec{y} \mid \vec{y} = \vec{x}_i || \vec{z}_i\}$
- hypothesis: h of Θ, needs to be revised into h'

Expectation Maximization

- goal of EM: $h' = \arg\max E(\log_2 p(Y|h'))$
- define a function $Q(h'|h) = E(\log_2 p(Y|h')|h, X)$
- Estimation (E) step:
  Calculate $Q(h'|h)$ using the current hypothesis h and the observed data X to estimate the probability distribution over Y

$$Q(h'|h) \leftarrow E(\log_2 p(Y|h')|h, X)$$

- Maximization (M) step
  Replace hypothesis h by h' that maximizes the function Q

$$h \leftarrow \arg\max_{h'} Q(h'|h)$$

Expectation Maximization

- expectation step requires applying the model to be learned
  - Bayesian inference
- gradient ascent search
  - converges to the next local optimum
  - global optimum is not guaranteed

Expectation Maximization

$$\begin{tikzcd}[column sep=huge]
\boxed{h} \arrow[r, bend left, "Q(h'|h) \leftarrow E(\ln p(Y|h')|h, X)"] & \boxed{Q(h'|h)} \arrow[l, bend left, "h \leftarrow \arg\max_{h'} Q(h'|h)"]
\end{tikzcd}$$

- If Q is continuous, EM converges to the local maximum of the likelihood function $P(Y|h')$

Learning the Network Structure

- learning the network structure
- space of possible networks is extremely large ($> \mathcal{O}(2^n)$)
- a Bayesian network over a complete graph is always a possible answer, but not an interesting one (no modelling of independencies)
- problem of overfitting
- two approaches
  - constraint-based learning
  - (score-based learning)

Constraint-based Structure Learning

- estimate the pairwise degree of independence using conditional mutual information
- determine the direction of the arcs between non-independent nodes

Estimating Independence

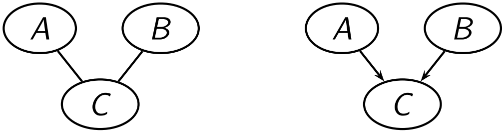
- conditional mutual information

$$CMI(A,B|\mathcal{X}) = \sum_{\mathcal{X}} \widehat{P}(\mathcal{X}) \sum_{A,B} \widehat{P}(A,B|\mathcal{X}) log_2 \frac{\widehat{P}(A,B|\mathcal{X})}{\widehat{P}(A|\mathcal{X})\widehat{P}(B|\mathcal{X})}$$

- two nodes are independent if $CMI(A,B|\mathcal{X}) = 0$
- choose all pairs of nodes as non-independent, where the significance of a χ^2-test on the hypothesis $CMI(A,B|\mathcal{X}) = 0$ is above a certain user-defined threshold
- high minimal significance level: more links are established
- result is a skeleton of possible candidates for causal influence

Determining Causal Influence

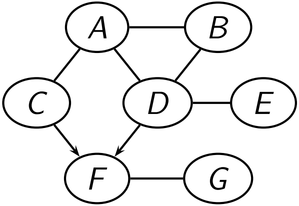
- Rule 1 (introduction of v-structures): $A - C$ and $B - C$ but not $A - B$ introduce a v-structure $A \rightarrow C \leftarrow B$ if there exists a set of nodes $\mathcal{X}$ so that A is d-separated from B given $\mathcal{X}$



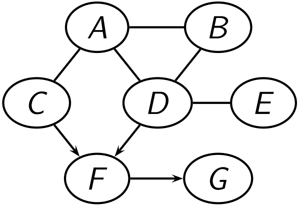
Determining Causal Influence

- Rule 2 (avoid new v-structures): When Rule 1 has been exhausted and there is a structure $A \rightarrow C - B$ but not $A - B$ then direct $C \rightarrow B$
- Rule 3 (avoid cycles): If $A \rightarrow B$ introduces a cycle in the graph do $A \leftarrow B$
- Rule 4 (choose randomly): If no other rule can be applied to the graph, choose an undirected link and give it an arbitrary direction

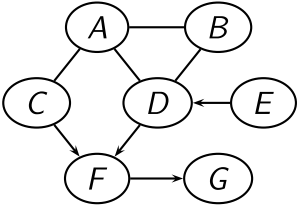
Determining Causal Influence



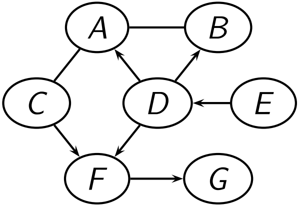
Rule 1



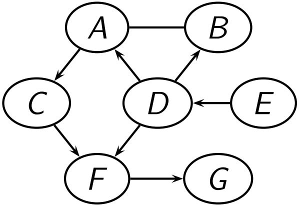
Rule 2



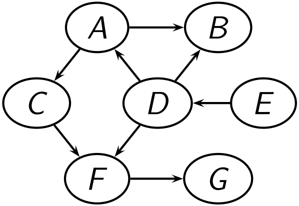
Rule 4



Rule 2



Rule 2



Rule 4

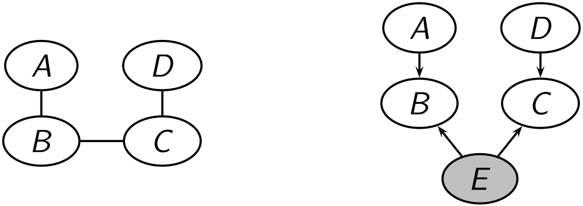
Determining Causal Influence

- independence/non-independence candidates might contradict each other
- $\neg I(A, B), \neg I(A, C), \neg I(B, C)$, but $I(A, B|C), I(A, C|B)$ and $I(B, C|A)$
  - remove a link and build a chain out of the remaining ones

  - uncertain region: different heuristics might lead to different structures

Determining Causal Influence

- $I(A, C), I(A, D), I(B, D)$



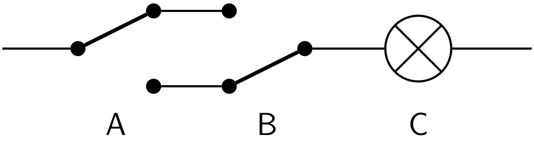
  - problem might be caused by a hidden variable $E \rightarrow B$ $E \rightarrow C$ $A \rightarrow B$ $D \rightarrow C$

Constraint-based Structure Learning

- useful results can only be expected, if
  - the data is complete
  - no (unrecognized) hidden variables obscure the induced influence links
  - the observations are a faithful sample of an underlying Bayesian network
    - the distribution of cases in $\mathcal{D}$ reflects the distribution determined by the underlying network
    - the estimated probability distribution is very close to the underlying one
  - the underlying distribution is recoverable from the observations

Constraint-based Structure Learning

- example of an unrecoverable distribution:
  - two switches: $P(A = up) = P(B = up) = 0.5$
  - $P(C = on) = 1$ if $val(A) = val(B)$
  - $\rightarrow I(A, C), I(B, C)$



- problem: independence decisions are taken on individual links (CMI), not on complete link configurations

$$P(C|A, B) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$