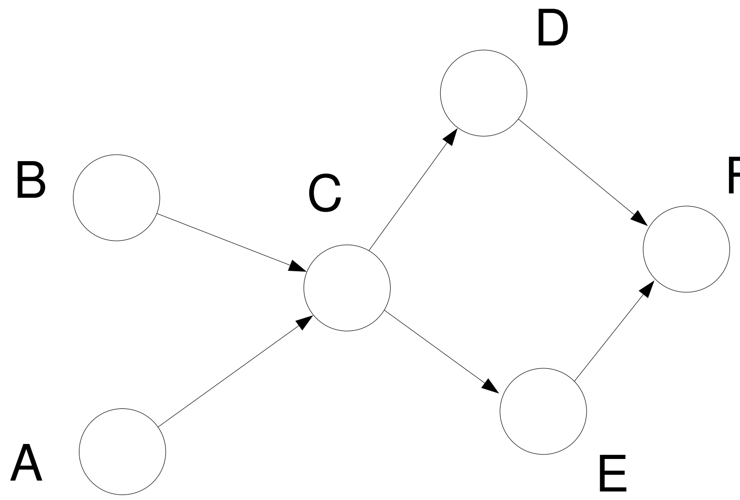
GRAPHICAL MODELS

- Graphical models are a marriage between graph theory and probability theory
- They are similar in spirit to neural networks, but are "more probabilistic"
  - (every node has a probabilistic interpretation)
- There are two kinds of graphical models; those based on *undirected* graphs and those based on *directed* graphs. Our focus will be directed graphs.

- Alternative names for graphical models: *belief networks, Bayesian networks, probabilistic independence networks, Markov random fields, loglinear models, influence diagrams*

Directed graphical models—basics

Consider an arbitrary directed (acyclic) graph, where each node in the graph corresponds to a random variable (scalar or vector):



Just as in a neural network, we can imagine some of these nodes to be "inputs," others to be "outputs," and others to be "hidden" (more on this later)

We want to associate a probability distribution $P(A, B, C, D, E, F)$ with this graph, and we want all of our calculations to respect this distribution

e.g.,

$$P(F|A, B) = \frac{\Sigma_C \Sigma_D \Sigma_E P(A, B, C, D, E, F)}{\Sigma_C \Sigma_D \Sigma_E \Sigma_F P(A, B, C, D, E, F)}$$

Some ways to use a graphical model

- *Prediction*
  - calculate the probability of *downstream* variables given the values of *upstream* variables
- *Diagnosis*
  - calculate the probability of *upstream* variables given the values of *downstream* variables
- *Value of information*
  - find the node that has the most effect on the value of a probability of interest
- *Cleaning up missing values*
  - fill in a node whose value is unknown with its most probable value
- *Learning*
  - how can we improve a model? how to improve parameter values? what additional nodes or links should be added to get a better model?

Specification of a graphical model

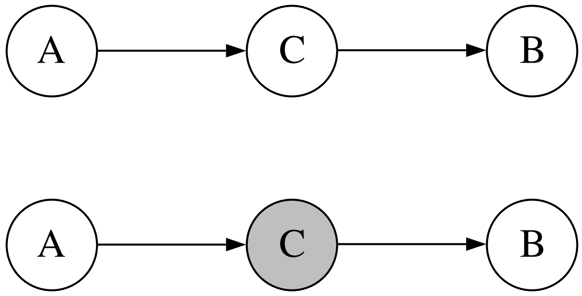
There are two components to any graphical model:

- the *qualitative* specification
- the *quantitative* specification

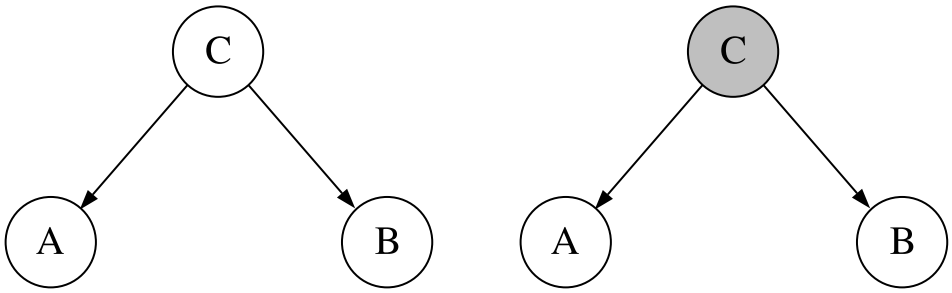
Where do these come from in real life?

- prior knowledge of causal relationships
- prior knowledge of modular relationships
- assessment from experts
- learning from data

Qualitative specification of graphical models

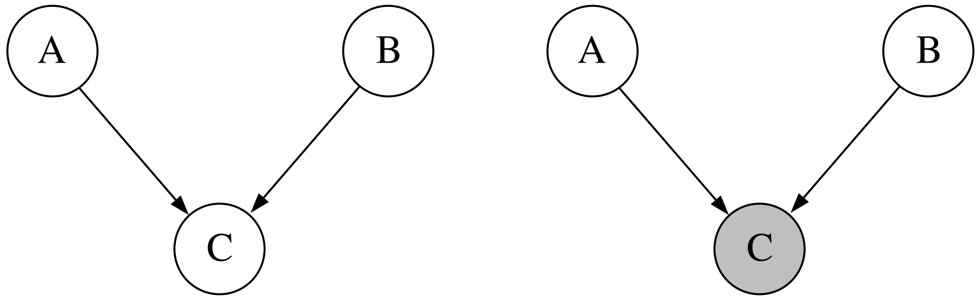


- A and B are marginally *dependent*
- A and B are conditionally *independent*



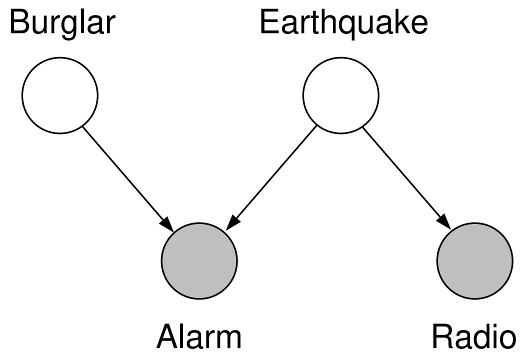
- A and B are marginally *dependent*
- A and B are conditionally *independent*

Semantics of graphical models (cont)



- A and B are marginally *independent*
- A and B are conditionally *dependent*
- This is the interesting case...

"Explaining away"

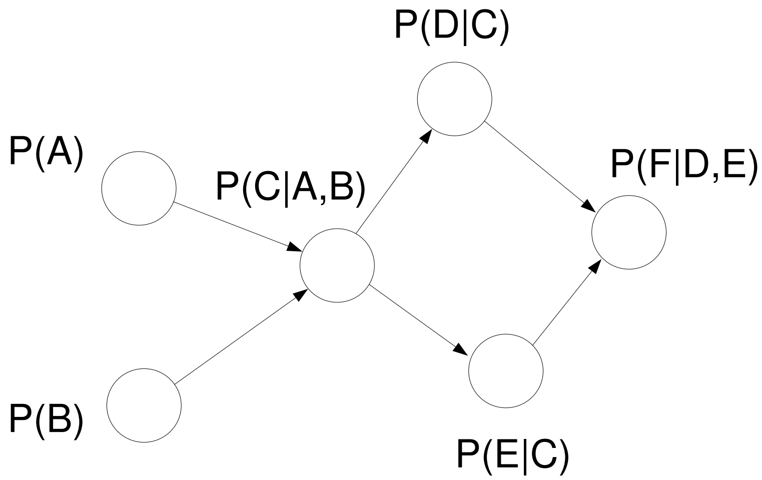


- All connections (in both directions) are "excitatory"
- But an increase in "activation" of Earthquake leads to a decrease in "activation" of Burglar
- Where does the "inhibition" come from?

Quantitative specification of directed graphical models

Question: how do we specify a joint distribution over the nodes in the graph?

Answer: associate a conditional probability with each node:



and take the product of the *local* probabilities to yield the *global* probabilities

Justification

In general, let $\{S\}$ represent the set of random variables corresponding to the N nodes of the graph

For any node S_i, let $\mathrm{pa}(S_i)$ represent the set of parents of node S_i

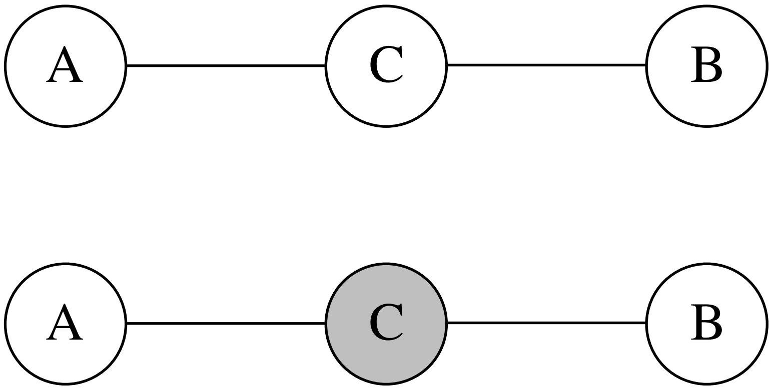
Then

$$\begin{aligned} P(S) &= P(S_1)P(S_2|S_1)\cdots P(S_N|S_{N-1},\ldots,S_1) \\ &= \prod_i P(S_i|S_{i-1},\ldots,S_1) \\ &= \prod_i P(S_i|\mathrm{pa}(S_i)) \end{aligned}$$

where the last line is by assumption

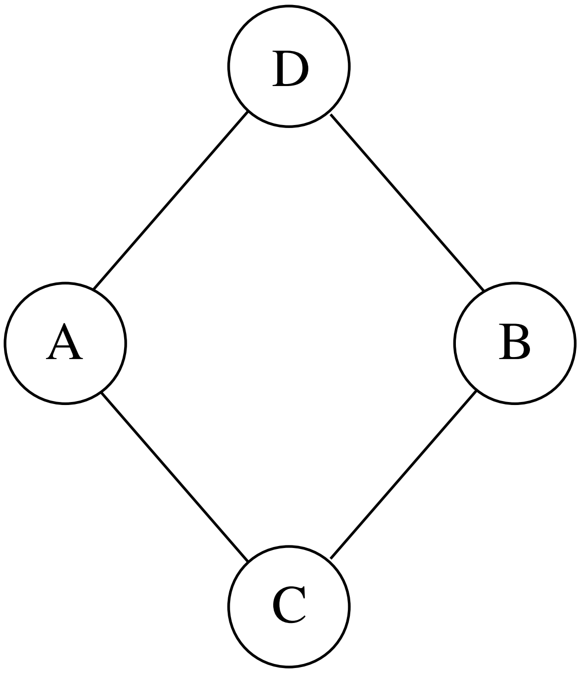
It is possible to prove a theorem that states that if arbitrary probability distributions are utilized for $P(S_i|\mathrm{pa}(S_i))$ *in the formula above, then the family of probability distributions obtained is exactly that set which respects the qualitative specification (the conditional independence relations) described earlier*

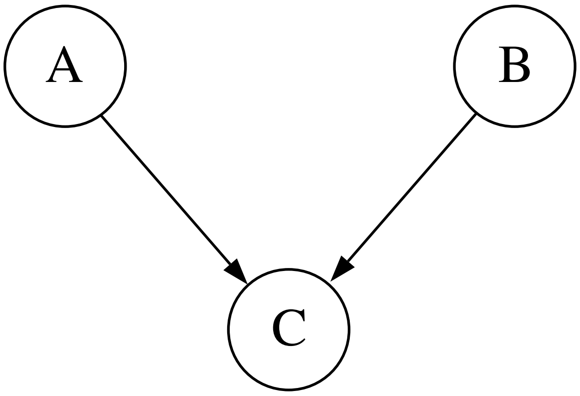
Semantics of undirected graphs



- A and B are marginally *dependent*
- A and B are conditionally *independent*

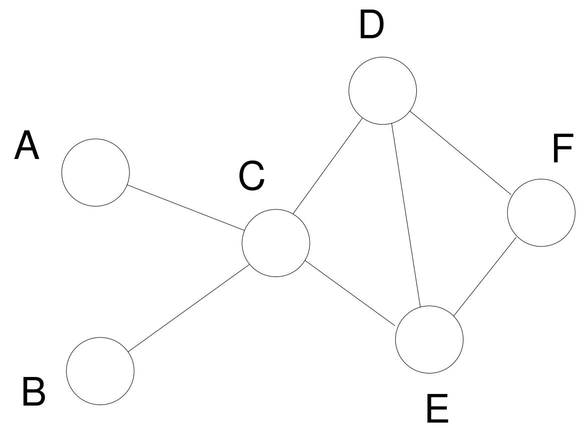
Comparative semantics



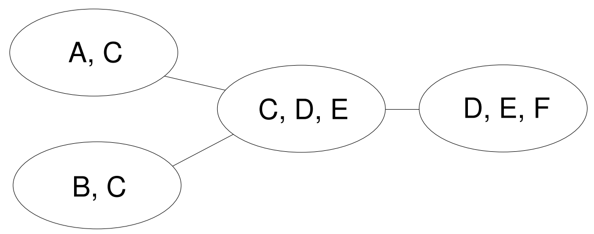


- The graph on the left yields conditional independencies that a directed graph can't represent
- The graph on the right yields marginal independencies that an undirected graph can't represent

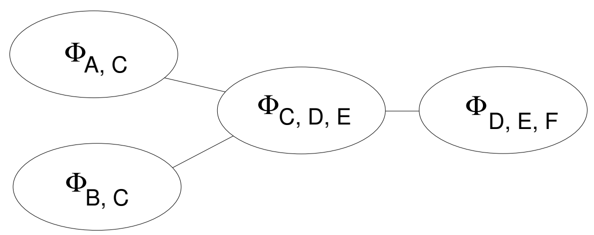
Quantitative specification of undirected graphical models



- identify the *cliques* in the graph:



- define a *configuration* of a clique as a specification of values for each node in the clique
- define a *potential* of a clique as a function that associates a real number with each configuration of the clique



Quantitative specification (cont.)

- finally, define the probability of a *global* configuration of the nodes as the product of the *local* potentials on the cliques:

$$P(A, B, C, D, E, F) = \phi_{(A,B)}\phi_{(B,C)}\phi_{(C,D,E)}\phi_{(D,E,F)}$$

It is then possible to prove a theorem that states that if arbitrary potentials are utilized in the product formula for probabilities, then the family of probability distributions obtained is exactly that set which respects the qualitative specification (the conditional independence relations) described earlier

(This theorem is known as the Hammersley-Clifford theorem)

Evidence and Inference

- "Absorbing evidence" means observing the values of certain of the nodes
- Absorbing evidence divides the units of the network into two groups:

visible units $\{V\}$	those for which we have instantiated values ("evidence nodes").
hidden units $\{H\}$	those for which we do not have instantiated values.

- "Inference" means calculating the *conditional distribution*

$$P(H|V) = \frac{P(H,V)}{\Sigma_{\{H\}}\, P(H,V)}$$

  – *prediction* and *diagnosis* are special cases

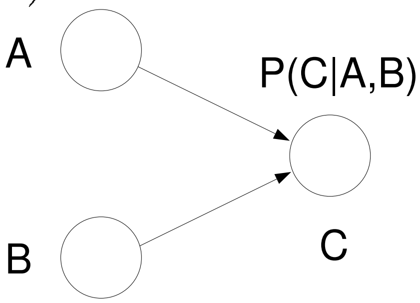
Inference algorithms for directed graphs

- There are several inference algorithms; some of which operate directly on the directed graph
- The most popular inference algorithm, known as the *junction tree* algorithm (which we'll discuss here), operates on an undirected graph
- It also has the advantage of clarifying some of the relationships between the various algorithms

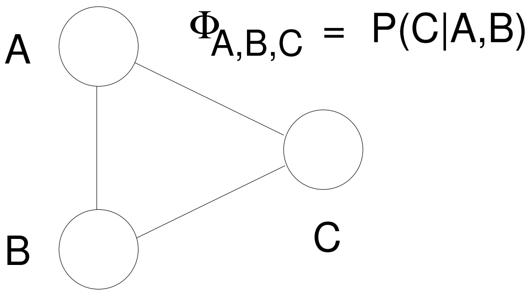
To understand the junction tree algorithm, we need to understand how to "compile" a directed graph into an undirected graph

Moral graphs

- Note that for both directed graphs and undirected graphs, the joint probability is in a product form
- So let's convert local conditional probabilities into potentials; then the products of potentials will give the right answer
- Indeed we can think of a conditional probability, e.g., $P(C|A,B)$ as a function of the three variables A, B, and C (we get a real number for each configuration):



- Problem: A node and its parents are not generally in the same clique
- Solution: Marry the parents to obtain the "moral graph"



Moral graphs (cont.)

- Define the potential on a clique as the product over all conditional probabilities contained within the clique
- *Now* the products of potentials gives the right answer:

$$
\begin{aligned}
& P(A, B, C, D, E, F) \\
& \quad = P(A)P(B)P(C|A, B)P(D|C)P(E|C)P(F|D, E) \\
& \quad = \phi_{(A,B,C)}\phi_{(C,D,E)}\phi_{(D,E,F)}
\end{aligned}
$$

where

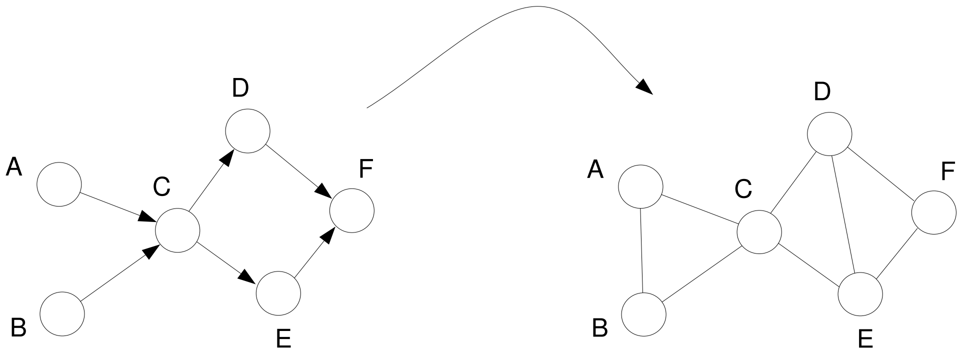
$$\phi_{(A,B,C)} = P(A)P(B)P(C|A, B)$$

and

$$\phi_{(C,D,E)} = P(D|C)P(E|C)$$

and

$$\phi_{(D,E,F)} = P(F|D, E)$$

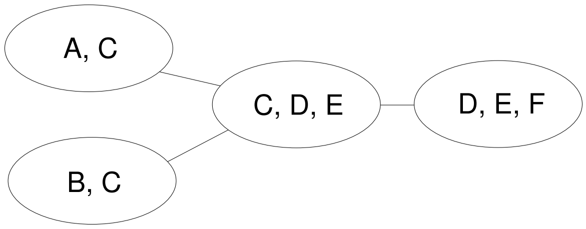


Propagation of probabilities

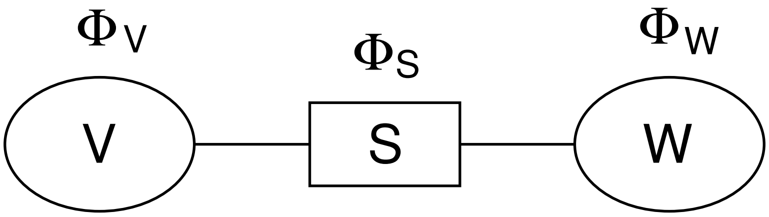
Now suppose that some evidence has been absorbed. How do we propagate this effect to the rest of the graph?

Clique trees

A *clique tree* is an (undirected) tree of cliques



Consider cases in which two neighboring cliques V and W have an overlap S (e.g., (A, C) overlaps with (C, D, E)).



- the cliques need to "agree" on the probability of nodes in the overlap; this is achieved by *marginalizing* and *rescaling*:

$$\phi_S^* = \sum_{V \setminus S} \phi_V$$

$$\phi_W^* = \phi_W \frac{\phi_S^*}{\phi_S}$$

- this occurs in parallel, distributed fashion throughout the clique tree

Clique trees (cont.)

This simple local message-passing algorithm on a clique tree defines the general probability propagation algorithm for directed graphs!

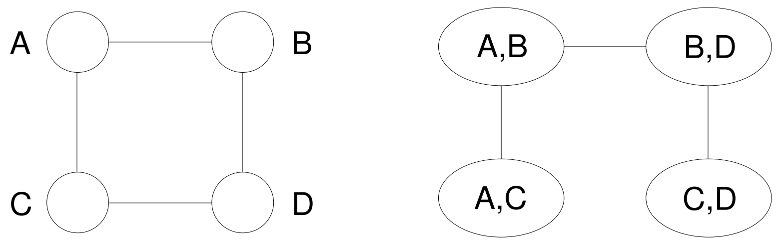
Many interesting algorithms are special cases:

- calculation of posterior probabilities in mixture models
- Baum-Welch algorithm for hidden Markov models
- posterior propagation for probabilistic decision trees
- Kalman filter updates

The algorithm seems reasonable. Is it correct?

A problem

Consider the following graph and a corresponding clique tree:



Note that C appears in two non-neighboring cliques.

Question: What guarantee do we have that the probability associated with C in these two cliques will be the same?

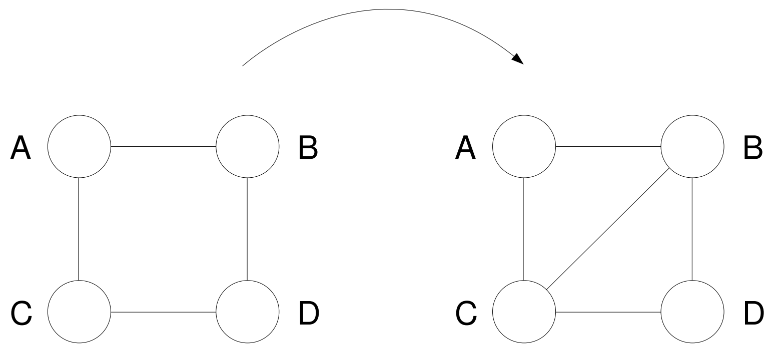
Answer: Nothing. In fact this is a problem with the algorithm as described so far. It is not true that in general *local* consistency implies *global* consistency.

What else do we need to get such a guarantee?

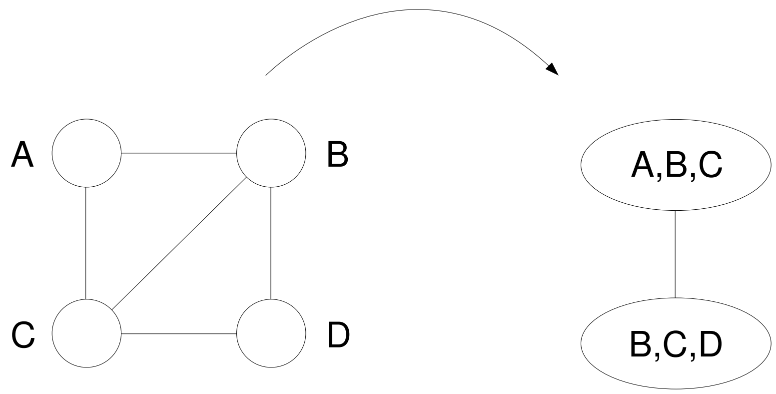
Triangulation (last idea, hang in there)

A triangulated graph is one in which *no cycles with four or more nodes* exist in which there is *no chord*

We *triangulate* a graph by adding chords:



Now we no longer have our problem:



A clique tree for a triangulated graph has the *running intersection property*: if a node appears in two cliques, it appears everywhere on the path between the cliques

Thus *local consistency implies global consistency* for such clique trees

Junction trees

A clique tree for a triangulated graph is referred to as a *junction tree*

In junction trees, local consistency implies global consistency. Thus the local message-passing algorithm is (provably) correct.

It's also possible to show that *only* triangulated graphs have the property that their clique trees are junction trees. Thus, we *must* triangulate.

Summary of the junction tree algorithm

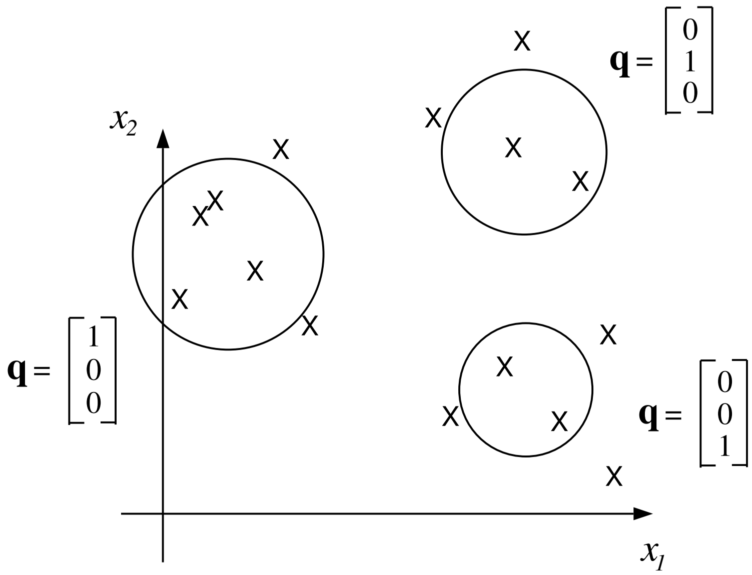
---

1. *Moralize* the graph
2. *Triangulate* the graph
3. *Propagate* by local message-passing in the junction tree

- note that the first two steps are “off-line”
- the algorithm is easily implemented in object-oriented code

Example: Gaussian mixture models

- A Gaussian mixture model is a popular clustering model
- We'll discuss in fuller detail tomorrow



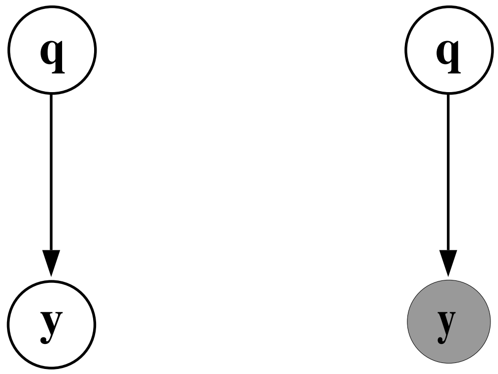
- hidden state $\mathbf{q}$ is a multinomial RV
- output $\mathbf{x}$ is a Gaussian RV
- prior probabilities on hidden states:

$$\pi_i = P(q_i = 1)$$

- class-conditional probabilities:

$$P(\mathbf{x}|q_i = 1) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i)\}$$

Gaussian mixture models as a graphical model



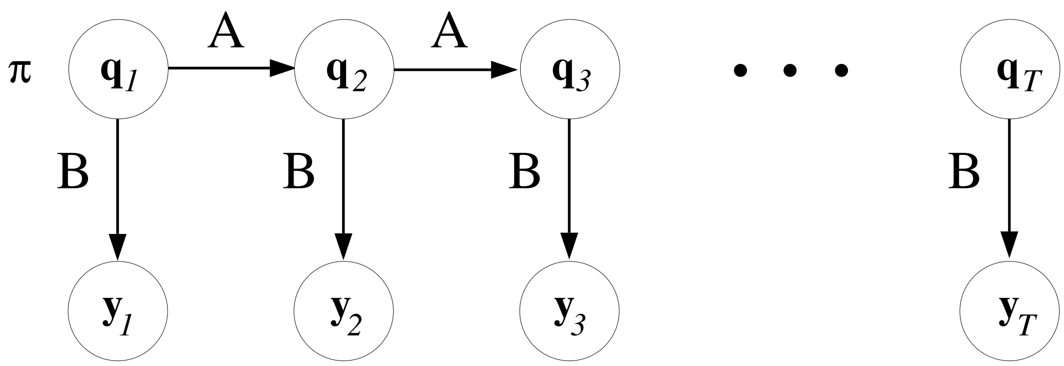
- The inference problem is to calculate the posterior probabilities:

$$
\begin{aligned}
P(q_i = 1|\mathbf{x}) &= \frac{P(\mathbf{x}|q_i = 1)P(q_i = 1)}{P(\mathbf{x})} \\
&= \frac{\frac{\pi_i}{|\Sigma_i|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i)\}}{\sum_j \frac{\pi_j}{|\Sigma_j|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \mu_j)^T \Sigma_j^{-1} (\mathbf{x} - \mu_j)\}}
\end{aligned}
$$

- This can be viewed as a normalized "distance"
- Tomorrow we'll refer to this calculation as the computation of "responsibilities" or the "E step of the EM algorithm"

Example: Hidden Markov models

- A hidden Markov model is a popular time series model
- It is a "mixture model with dynamics"
- Again, we'll discuss in fuller detail tomorrow



- T time steps
- M states ($\mathbf{q}_t$ is a multinomial RV)
- N outputs ($\mathbf{y}_t$ is a multinomial RV)
- state transition probability matrix A:

$$A = P(\mathbf{q}_{t+1}|\mathbf{q}_t)$$

- emission matrix B:

$$B = P(\mathbf{y}_t|\mathbf{q}_t)$$

- initial state probabilities π:

$$\pi = P(\mathbf{q}_1)$$

HMM as a graphical model

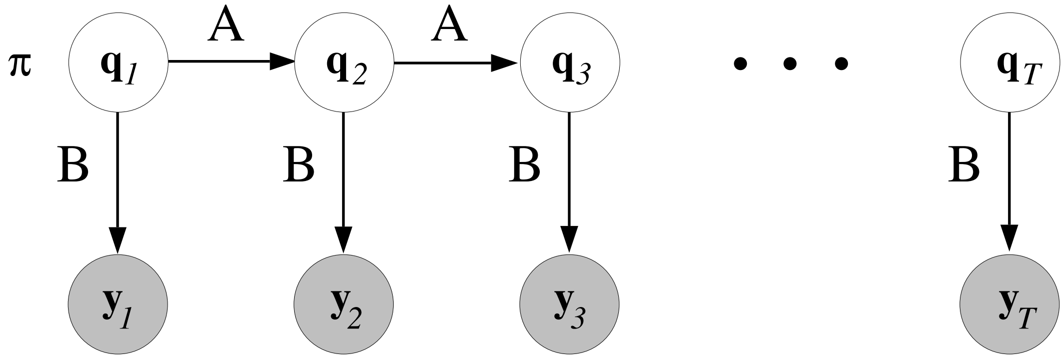
- Each node has a probability distribution associated with it.
- The graph of the HMM makes conditional independence statements.

  For example,

  $$P(\mathbf{q}_{t+1}|\mathbf{q}_t, \mathbf{q}_{t-1}) = P(\mathbf{q}_{t+1}|\mathbf{q}_t)$$

  can be read off the graph as a separation property.

HMM probability calculations



- The time series of $\mathbf{y}_t$ values is the *evidence*
- The *inference* calculation involves calculating the probabilities of the hidden states $\mathbf{q}_t$ given the evidence
- The classical algorithm for doing this calculation is the *forward-backward* algorithm

The forward-backward algorithm involves:

- multiplication (conditioning)
- summation (marginalization) in the lattice.

This algorithm is efficient, because of the lattice structure.

It is a special case of the junction tree algorithm.

Summary of graphical models

- Qualitative specification of probabilistic models on graphs
- Quantitative specification of probabilistic models on graphs
- Ease of incorporation of prior (causal) knowledge
- Junction tree algorithm