Graphical models, neural networks and variational approximations

Michael Jordan

Center for Biological and Computational Learning
Massachusetts Institute of Technology
E25-229
Cambridge, MA 02139

In collaboration with:

Zoubin Ghahramani, Tommi Jaakkola and Lawrence Saul

June, 1996

Overview

- Graphical models
- Mixture models
- Hidden Markov models
- Hierarchical mixtures of experts
- Exact calculations for general graphical models
  - Junction tree approach
  - Statistical mechanical approach
- Variational approximation
- Sigmoid belief networks
- Noisy-OR networks
- Exploiting tractable substructures
- Factorial hidden Markov models

Importance of probability theory

Probability theory provides an surefooted guide to the slippery world of complex network models:

- intuitively satisfying
- provides a relationship between inference and learning
- justifies and clarifies approximations
- allows qualitative information to be utilized
- allows prior knowledge to be incorporated naturally
- etc.

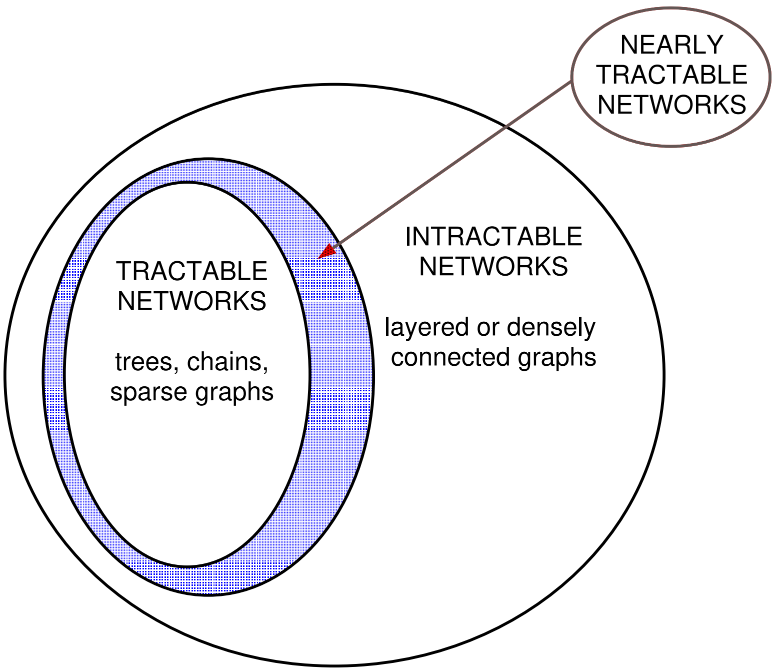
Moreover,

- The tractability barrier is not what it used to be (as we will see in this talk)

Tractable vs. intractable networks

There are many efficient probabilistic network models, including mixture models, Hidden Markov models, Kalman filters, and decision trees

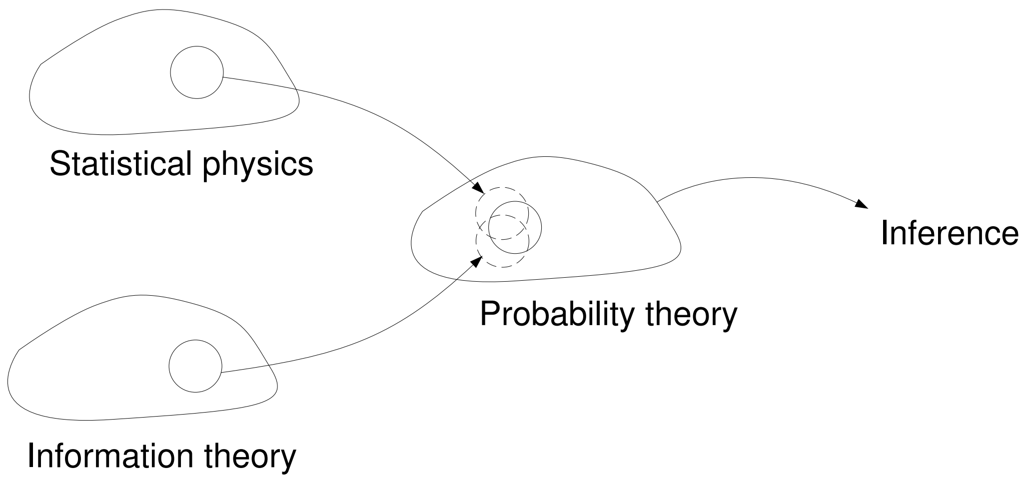
- In general, provably efficient algorithms exist for trees and chains
- Sampling methods are generally applicable
- There seem to be a reasonably large class of models for which efficient approximations can be developed



Formalisms

There is substantial agreement that probability theory is the best framework available in which to develop learning algorithms.

That doesn't mean that we always have to work with probabilities directly. One can work with energies (statistical physics) or codes (information theory) and convert the results to probabilities using standard theorems. Sometimes energies or codes are easier to work with.



(circles: those models that are relatively easy to specify and understand)

Graphical models

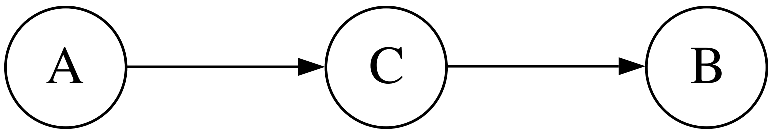
- The graphical model framework provides a general formalism that links networks and probabilities
- There are two broad classes of graphical model—those based on *directed* graphs and those based on *undirected* graphs
- Undirected graphical models are also known as Markov random fields
- I will focus on directed graphs in this talk

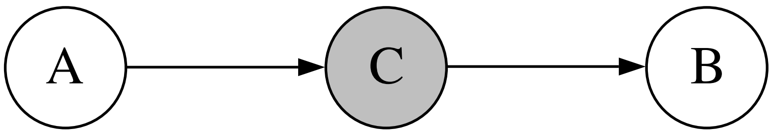
There are two components to any graphical model

- the qualitative specification
- the quantitative specification

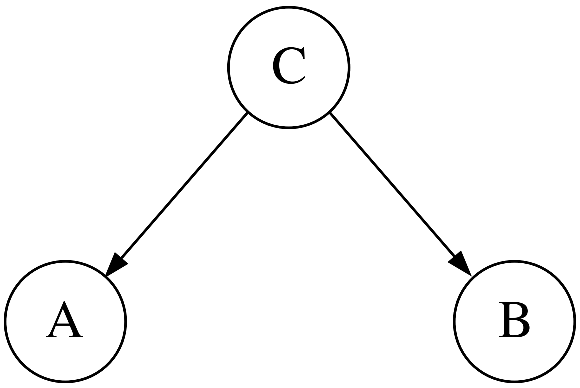
Qualitative specification of graphical models

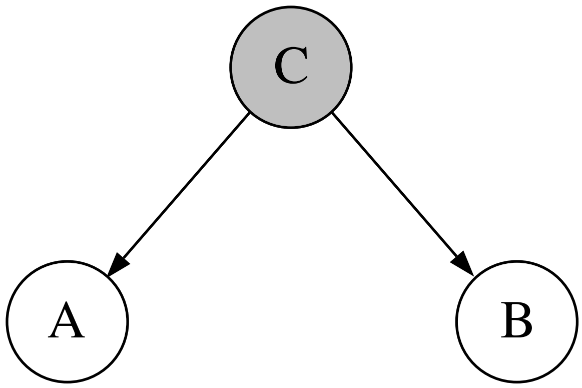
(Frydenberg, Lauritzen, Pearl, others)





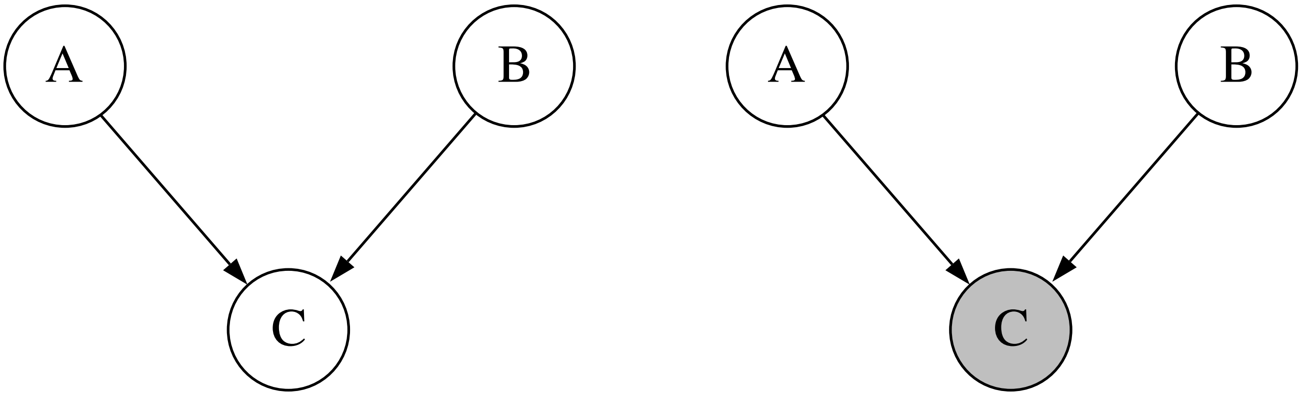
- A and B are marginally *dependent*
- A and B are conditionally *independent*





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

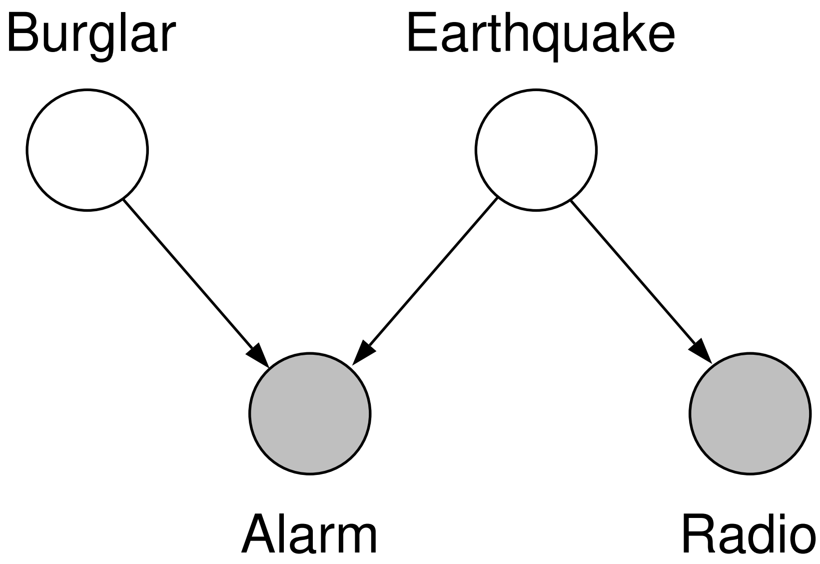
Semantics of graphical models (cont)



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

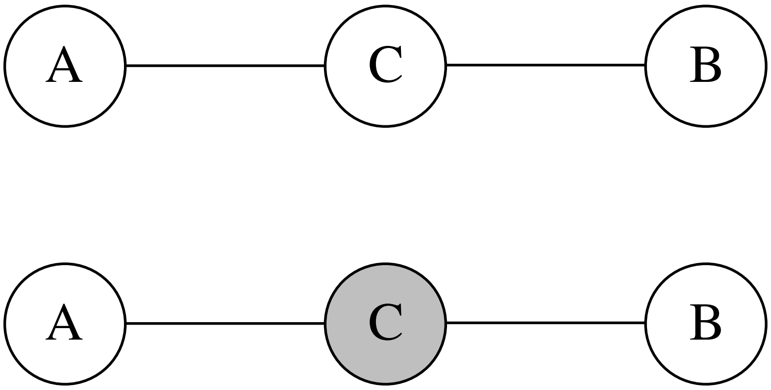
- This is the interesting case...

“Explaining away”



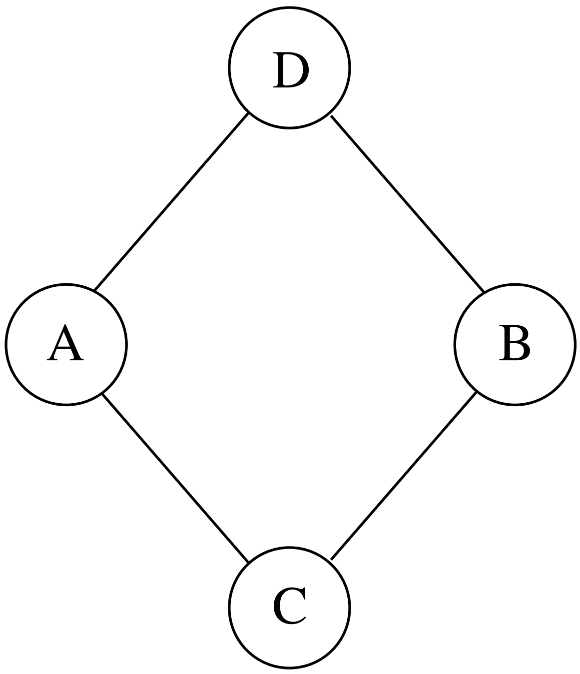
- All connections (in both directions) are “excitatory”
- But an increase in “activation” of Earthquake leads to a decrease in “activation” of Burglar
- It’s hard to handle this in the standard neural network formalism without hacking
- We need the concept of “Markov blanket” and we need algorithms that take into account the Markov blanket

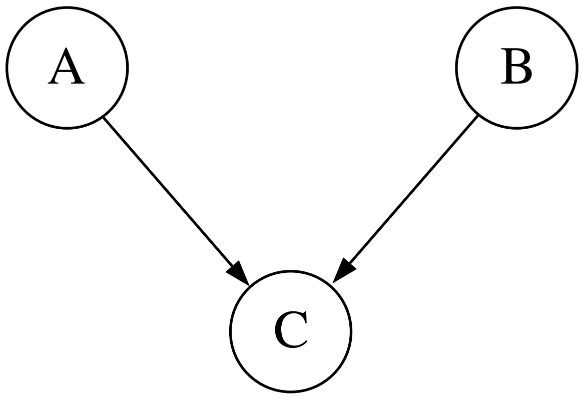
Semantics of undirected graphs



- 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 graphical models

- 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

Example: Gaussian mixture models

- A Gaussian mixture model is a popular clustering model
- It provides a probabilistic approach to "competitive learning"

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

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

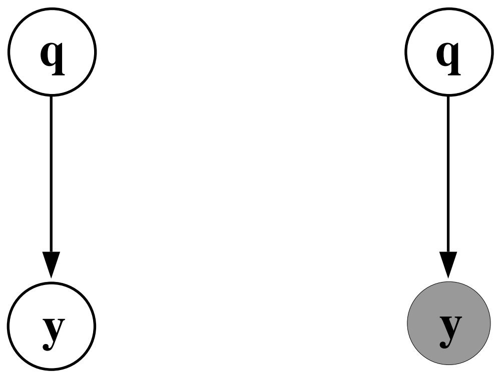
- output probabilities:

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

- log likelihood for a single data point:

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

Gaussian mixture models as a graphical model

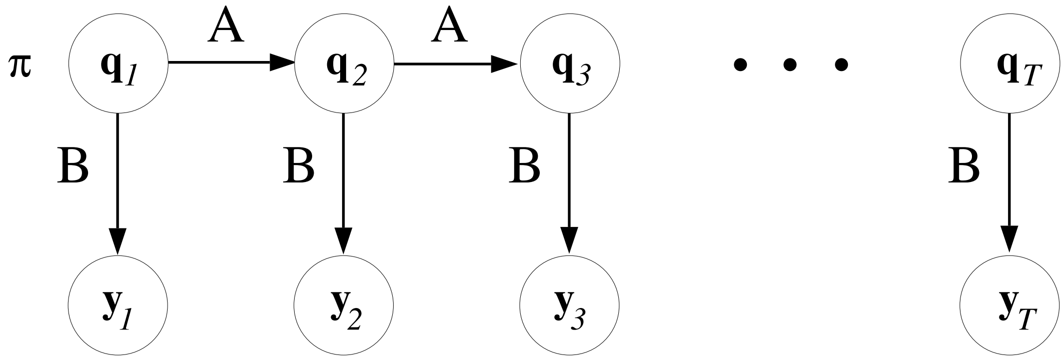


- The key calculation is to calculate the posterior probabilities:

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

- This can be viewed as a normalized "distance"
- The calculation can also be viewed as "reversing the arrow" in the graphical model

Example: Hidden Markov models



- 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

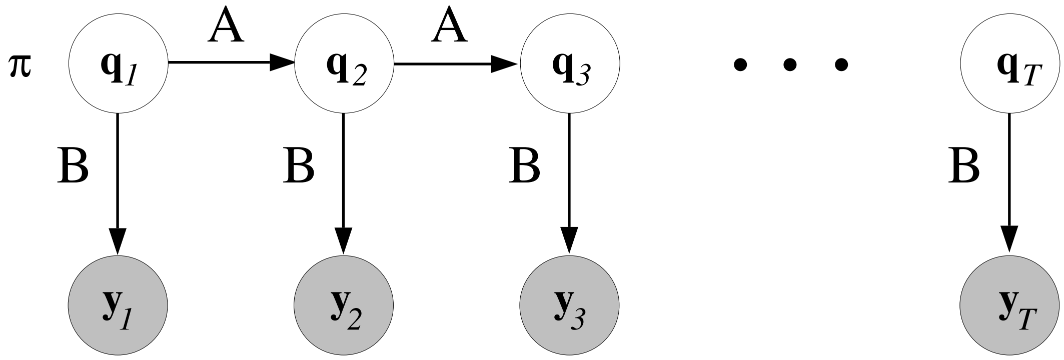
- Each node has a probability distribution associated with it.
- The graph of the HMM makes conditional independence statements.

  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 *observed*.
- Calculate probabilities of *unobserved* hidden states $\mathbf{q}_t$ via the forward-backward algorithm.

The forward-backward algorithm involves:

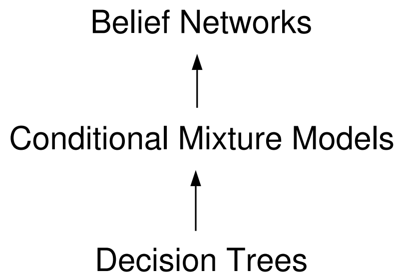
- multiplication (conditioning)
- summation (marginalization) in the lattice.

This algorithm is efficient, because of the lattice structure.

It is a special case of a general exact graphical algorithm.

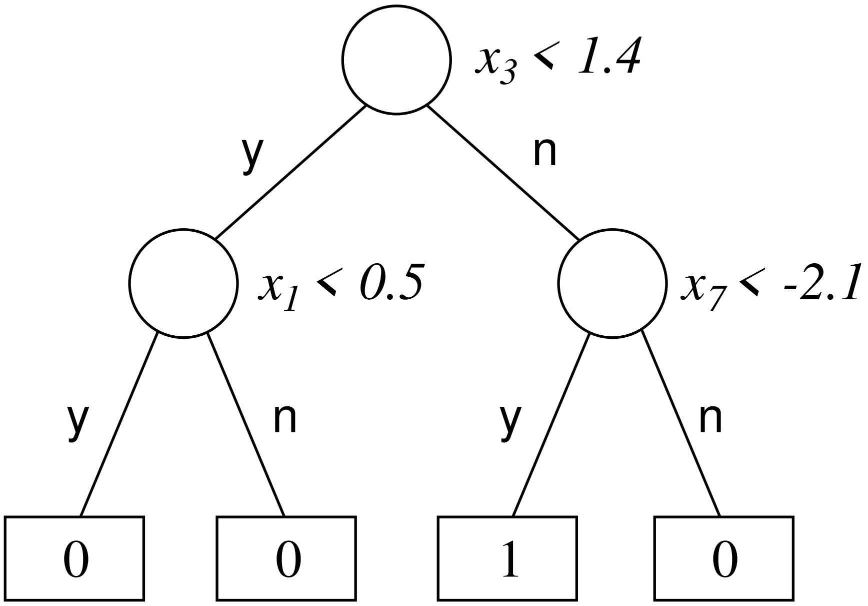
Example: Probabilistic decision trees

(Jordan and Jacobs, 1994)



- A method for probabilizing decision trees that yields an efficient graphical algorithm.
- The algorithm is to trees what Baum-Welch (HMM's) is to chains.
- Convert decision tree problems into (conditional) mixture problems. Mixtures are instances of general graphical models.

Classical decision trees
(CART, C4.5)



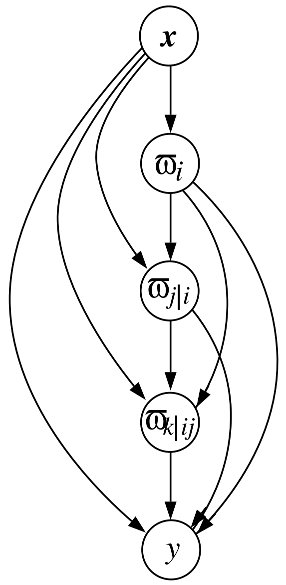
- drop the data set down the tree
- at each node, try to find a split of the input space (a half-plane) that yields the largest gain in "purity" on left and right
- build a large tree and prune backward to create a nested sequence of trees
- pick the best tree from the sequence using cross-validation

Probabilistic decision trees

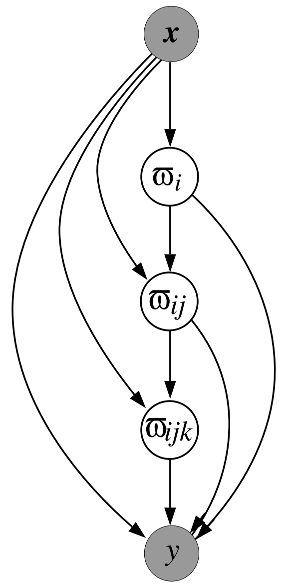
- drop inputs down the tree and use probabilistic models for decisions (splits)
- at leaves of trees use probabilistic models to generate outputs from inputs
- use a Bayes' rule recursion to compute posterior credit for nonterminals in the tree

The basic idea is to convert the decision tree into a conditional mixture model

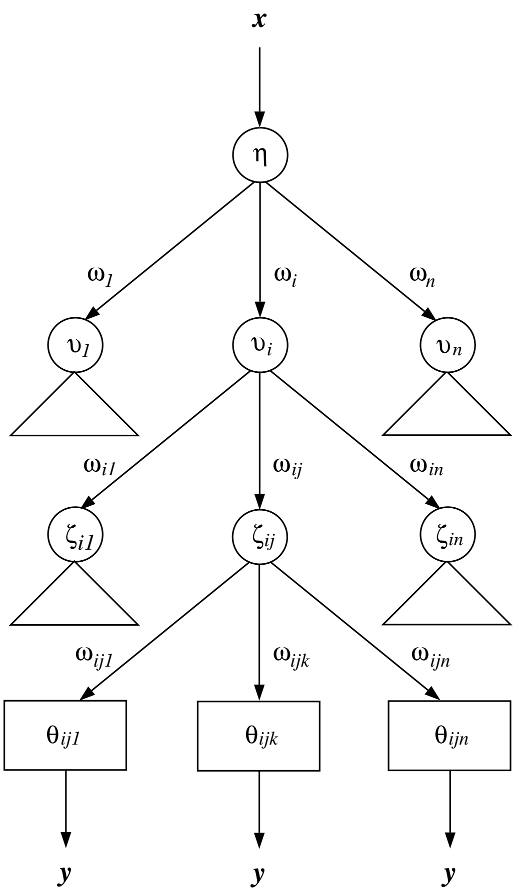
Representation as a graphical model



For learning and inference, the nodes for **x** and **y** are observed (shaded)

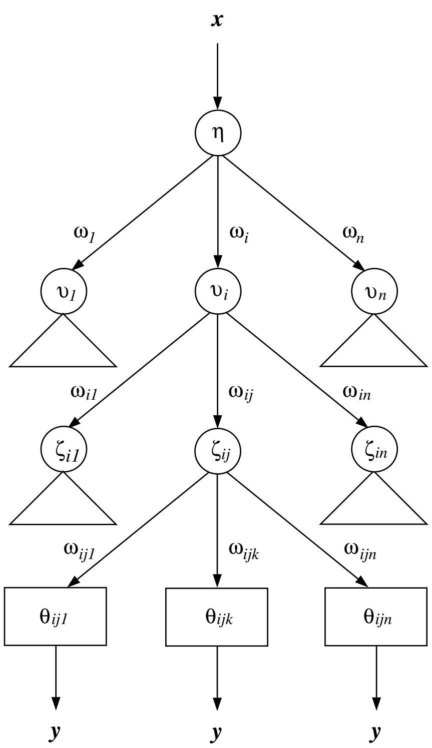


We need to calculate probabilities of unshaded nodes



Model the splits in the decision tree using categorical probability models

- let ω_i, ω_{ij}, $\omega_{ijk}, \ldots$ represent multinomial split variables at the nonterminals
- these variables will be treated as "hidden" variables (cf. states of an HMM)
- each path down the tree defines a component of a conditional mixture



Splitting models at the nonterminals:

$$
\begin{aligned}
g_i &\equiv P(\omega_i|\mathbf{x}) \\
g_{j|i} &\equiv P(\omega_{ij}|\mathbf{x}, \omega_i) \\
g_{k|ij} &\equiv P(\omega_{ijk}|\mathbf{x}, \omega_i, \omega_{ij})
\end{aligned}
$$

Output models at the leaves:

$$P(\mathbf{y}|\mathbf{x}, \omega_i, \omega_{ij}, \omega_{ijk} \ldots)$$

Likelihood of the tree is a mixture

The total probability of an output $\mathbf{y}$ given an input $\mathbf{x}$ is given by the sum across all paths from the root to the leaves:

$$\begin{aligned} P(\mathbf{y}|\mathbf{x},\Theta) \;=\; & \textstyle\sum_i P(\omega_i|\mathbf{x}) \sum_j P(\omega_{ij}|\mathbf{x},\omega_i) \\ & \textstyle\sum_k P(\omega_{ijk}|\mathbf{x},\omega_i,\omega_{ij}) \cdots \\ & P(\mathbf{y}|\mathbf{x},\omega_i,\omega_{ij},\omega_{ijk}\ldots) \end{aligned}$$

This is a (conditional) mixture model:

$$P(\mathbf{y}|\mathbf{x},\Theta) \;=\; \textstyle\sum_{ijk\ldots} P(\Omega_{ijk\ldots}|\mathbf{x}) P(\mathbf{y}|\mathbf{x},\Omega_{ijk\ldots})$$

Component Models

Split models (nonterminals)

- $P(\omega_i|\mathbf{x},\eta)$ is a classification model
- any parametric classification model is appropriate—we use a multinomial logit ("softmax") model

  this yields "soft" linear discriminants—soft version of a CART/C4.5 tree

Leaf models

- we use simple generalized linear models:

  Regression—linear regression

  Binary classification—logistic regression

  Multiway classification—multinomial logit model

  (can also handle count estimates, failure estimates, etc.)

Probability calculation in the decision tree

- The hidden variables are the unknown values of the decisions (splits) in the decision tree.

  Define indicator variables z_i, $z_{j|i}$, $z_{k|ij}, \ldots$
- Calculate the expected values of these indicator variables given the inputs and outputs:

$$E(z_i^{(p)}|\mathbf{x}^{(p)}, \mathbf{y}^{(p)}) = P(\omega_i^{(p)}|\mathbf{x}^{(p)}, \mathbf{y}^{(p)})$$

Posterior splitting probabilities

$$\begin{aligned} h_i &\equiv P(\omega_i|\mathbf{x}, \mathbf{y}) \\ h_{j|i} &\equiv P(\omega_{ij}|\mathbf{x}, \mathbf{y}, \omega_i) \\ h_{k|ij} &\equiv P(\omega_{ijk}|\mathbf{x}, \mathbf{y}, \omega_i, \omega_{ij}) \\ &\ldots \end{aligned}$$

(cf. *prior* splitting probabilities)

$$\begin{aligned} g_i &\equiv P(\omega_i|\mathbf{x}) \\ g_{j|i} &\equiv P(\omega_{ij}|\mathbf{x}, \omega_i) \\ g_{k|ij} &\equiv P(\omega_{ijk}|\mathbf{x}, \omega_i, \omega_{ij}) \\ &\ldots \end{aligned}$$

Bayes' rule yields:

$$h_i = \frac{g_i \sum_j g_{j|i} \sum_k g_{k|ij} P_{ijk}(\mathbf{y}|\mathbf{x})}{\sum_i g_i \sum_j \sum_j g_{j|i} \sum_k g_{k|ij} P_{ijk}(\mathbf{y}|\mathbf{x})}$$

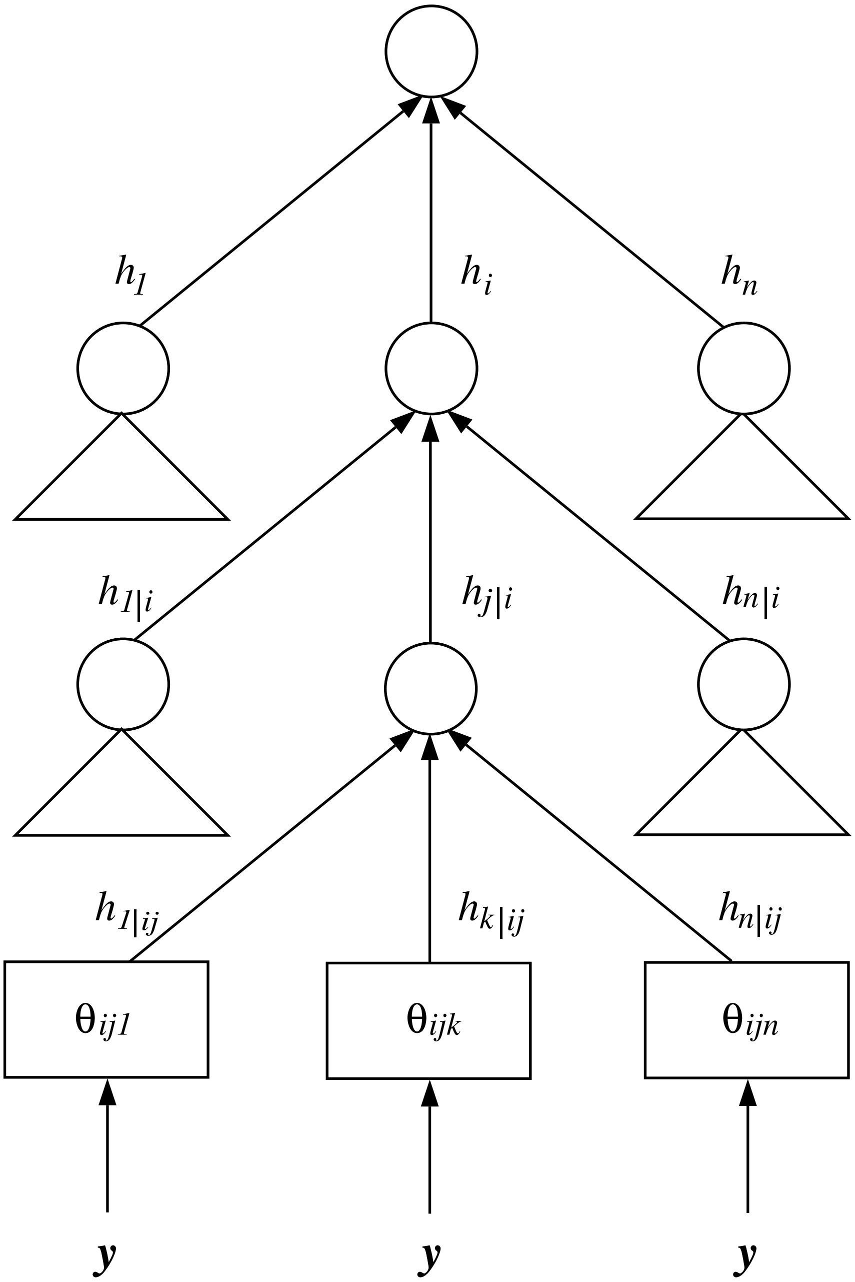
$$h_{j|i} = \frac{g_{j|i} \sum_k g_{k|ij} P_{ijk}(\mathbf{y}|\mathbf{x})}{\sum_j g_{j|i} \sum_k g_{k|ij} P_{ijk}(\mathbf{y}|\mathbf{x})}$$

$$h_{k|ij} = \frac{g_{k|ij} P_{ijk}(\mathbf{y}|\mathbf{x})}{\sum_k g_{k|ij} P_{ijk}(\mathbf{y}|\mathbf{x})}$$

$$\ldots$$

Note that the denominator at each level appears in the numerator at the next level up. We can organize the computation as a recursion.

Posterior propagation



This algorithm is efficient, because of the tree structure.

It is a special case of a general exact graphical algorithm.

Generic algorithm for exact computation

(Lauritzen, Spiegelhalter, Jensen, others)

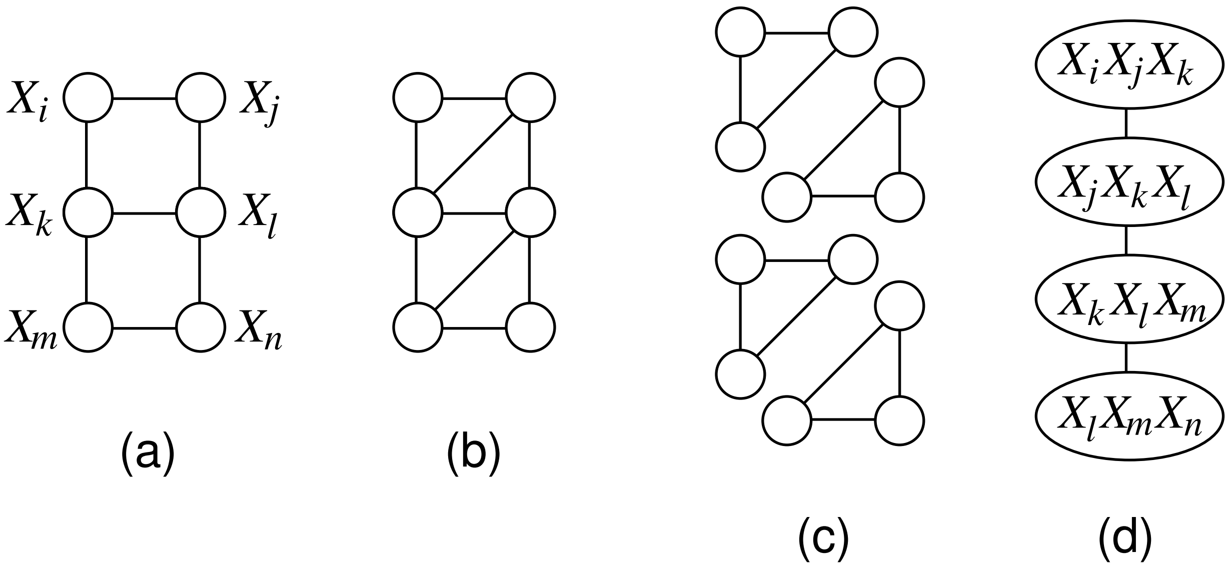
There is a generic algorithm (the junction tree algorithm) for computing probabilities in graphical models

Many interesting algorithms are special cases, e.g.:

- HMM's
- Kalman filters

The junction tree algorithm involves:

- multiplication (conditioning)
- summation (marginalization)



Convert from directed graphs to undirected graphs

- Marry the parents of all nodes ("moral graph")

Compile the undirected graph into a junction tree

- Triangulate the graph (b)
- The cliques of a triangulated graph can be arranged to form a tree with the running intersection property (d)
- Form probability potentials on the cliques of the triangulated graph
- The joint probability of the nodes in the original graph is the product of the potentials in the junction tree

Now probabilities can be calculated with a local message-passing algorithm.

One page primer on statistical mechanics

- N random variables: $\{S_i\}$
- *configuration* α: $\{S_i = s_i^\alpha\}$
- *energy* of configuration α: $E_\alpha \equiv E(\{s_i^\alpha\})$
- *Boltzmann factor*: $\exp\{-E_\alpha\}$
- *partition function*: $Z \equiv \Sigma_\alpha \exp\{-E_\alpha\}$
- *probability* of configuration α:

$$P_\alpha = \frac{\exp\{-E_\alpha\}}{Z}$$

- *free energy*:

$$F \equiv \Sigma_\alpha P_\alpha E_\alpha + \Sigma_\alpha P_\alpha \log P_\alpha$$

- Fundamental Theorem:

$$F = -\log Z$$

How is the energy defined?

- (Hopfield, 1982):

$$E_\alpha = -\sum_{i<j} J_{ij} S_i^\alpha S_j^\alpha$$

- Alternatively, suppose that we have a directed graphical model in which conditional probabilities have been defined (cf. HMM, HME):

$$P(S_i^\alpha | \mathrm{pa}(S_i^\alpha))$$

  The joint probability on the graph is a simple product:

$$P_\alpha = \Pi_i P(S_i^\alpha | \mathrm{pa}(S_i^\alpha))$$

  We can compile these into energies as follows:

$$E_\alpha = -\log P_\alpha$$

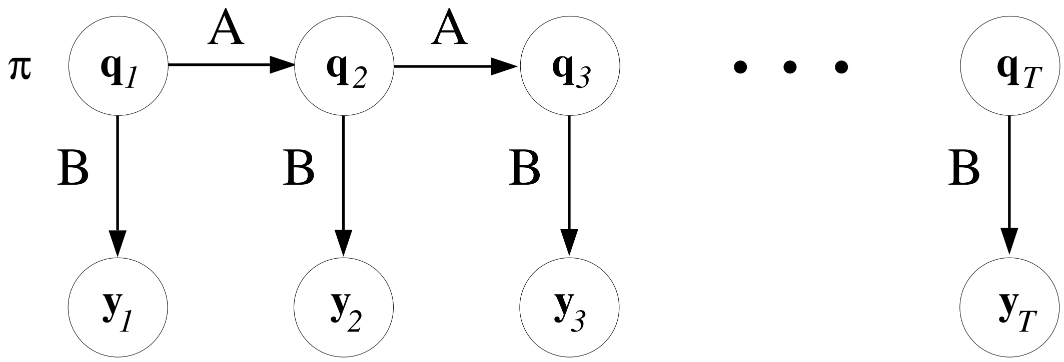
These energies are defined on the cliques:

$$
\begin{aligned}
E_\alpha &\equiv -\log P_\alpha \\
&= -\log\left[\Pi_i P(S_i^\alpha|\mathrm{pa}(S_i^\alpha))\right] \\
&= \Sigma_i\left[-\log P(S_i^\alpha|\mathrm{pa}(S_i^\alpha))\right]
\end{aligned}
$$

Moreover the partition function Z is unity:

$$Z = \Sigma_\alpha \exp\{-E_\alpha\} = \Sigma_\alpha P_\alpha = 1$$

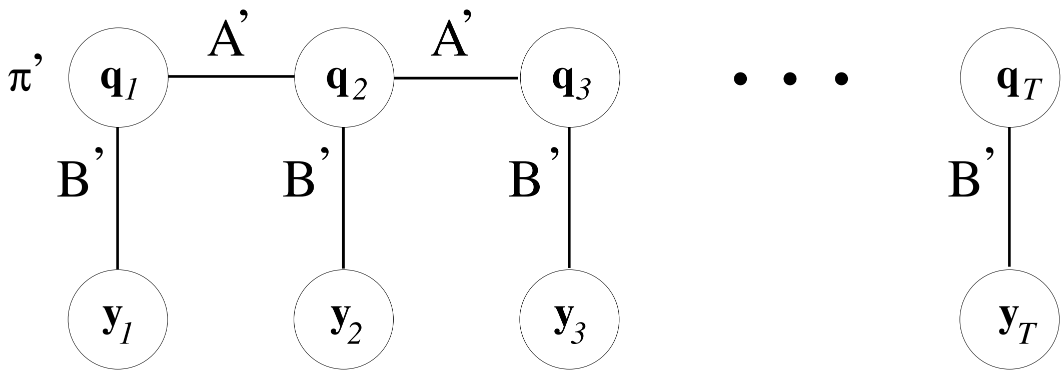
Example: Hidden Markov models



joint pdf:

$$P(\{q_t, y_t\}) = \pi_{q_1} a_{q_1 q_2} \dots a_{q_{T-1} q_T} b_{q_1 y_1} \dots b_{q_T y_T}$$

Undirected Hidden Markov models



Parameterization:

$$A'_{ij} = \log A_{ij}; \qquad B'_{ij} = \log B_{ij}; \qquad \pi'_i = \log \pi_i$$

Energy function:

$$H(\{q_t, y_t\}) = -\pi'_{q_1} - \Sigma_t A'_{q_1 q_2} - \Sigma_t B'_{q_1 y_2}$$

Then the same joint pdf is obtained via the Boltzmann distribution:

$$\begin{aligned} P(\{q_t, y_t\}) &= \frac{e^{-H}}{Z} \\ &= \pi_{q_1} a_{q_1 q_2} \ldots a_{q_{T-1} q_T} b_{q_1 y_1} \ldots b_{q_T y_T} \end{aligned}$$

and the partition function is unity

Calculations under the Boltzmann distribution

How do we calculate probabilities of various configurations of a network?

- In the Boltzmann machine, this was done by Gibbs sampling
- I.e., the probability of a node given its neighbors was calculated and a sample taken from that distribution (“noisy perceptron” rule)
- But there is an alternative, exact approach that doesn’t require sampling

Exact calculations—decimation

(cf. Saul and Jordan, 1995)

- There are graphical operations that prune nodes while preserving the Boltzmann distribution
- The idea is much like reducing an electrical circuit; we reduce the circuit until a very simple circuit remains
- There is a general form of decimation that is equivalent to the junction tree calculations

Approximations

For complex networks, approximations must be used.

There are two major methodologies:

- sampling methods (e.g., Gibbs sampling)
- variational methods

We will focus on variational methods in the remainder of the talk.

Let's begin with two examples of networks for which variational methods have been developed...

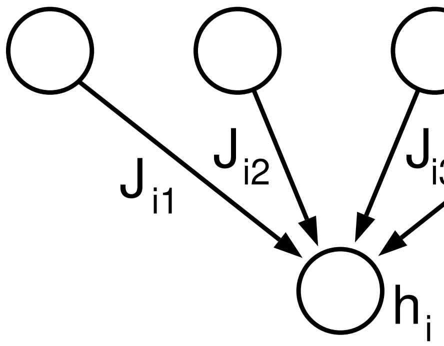
Sigmoid belief networks

(Neal; Saul, Jaakkola & Jordan)

- Units represent binary random variables:

$$\{S_i\}_{i=1}^N = \{0,1\}^N.$$

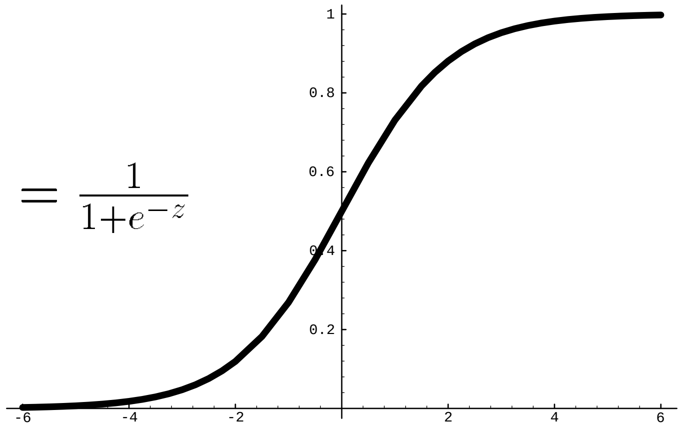
- Weights J_{ij} are attached to each link in the directed graph.



- A sigmoid function is used to represent conditional probabilities:

$$P(S_i = 1|\Pi_{S_i}) = \sigma\left(\sum_j J_{ij}S_j + h_i\right).$$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$



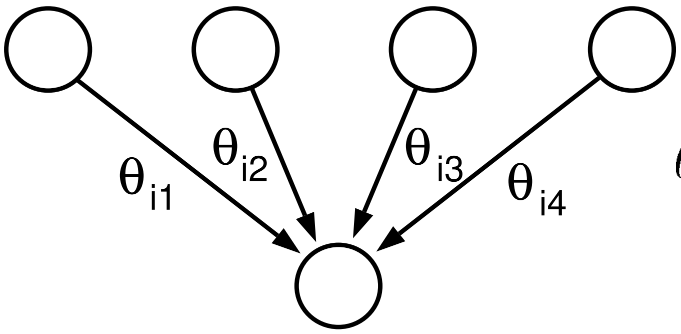
Noisy-OR networks

(Pearl)

- Units are activated by independent causal events:

$$P(S_i = 1 | \Pi_{S_i}) = 1 - \prod_j (1 - p_{ij})^{S_j}.$$

- Weights θ_{ij} are attached to each link in the directed graph.

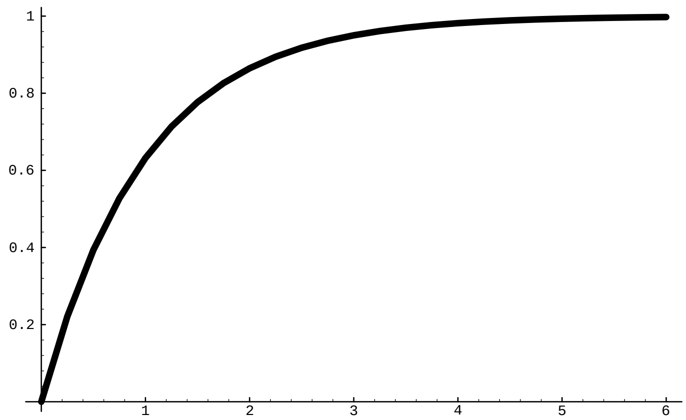


$$\theta_{ij} = -\ln(1 - p_{ij})$$

- Noisy-OR gates are represented by the activation function:

$$P(S_i = 1 | \Pi_{S_i}) = \rho\left(\sum_j \theta_{ij} S_j\right).$$

$$\rho(z) = 1 - e^{-z}$$



*	**sigmoid**	**noisy-OR**
units	$\{S_i\}_{i=1}^N = \{0,1\}^N$	$\{S_i\}_{i=1}^N = \{0,1\}^N$
weights	J_{ij}, h_i parameters in logistic regression	$\theta_{ij} = -\ln(1-p_{ij})$ constrained by probabilities
activation function	$\sigma(z) = \frac{1}{1+e^{-z}}$ monotone increasing	$\rho(z) = 1 - e^{-z}$ monotone increasing

- Sigmoid and noisy-OR gates have **similar, but not identical**, probabilistic semantics.

- Both provide **compact representations** of the conditional probabilities used to propagate beliefs.

Inference

- 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.

- Inferences are made under the **conditional distribution**

$$P\{H|V\} = \frac{P\{H,V\}}{Z_V},$$

where Z_V is the **likelihood** of the observed evidence:

$$Z_V = P\{V\} = \Sigma_{\{H\}} P\{H,V\}.$$

Variational approximation

(see, e.g., Saul, Jaakkola & Jordan, 1996)

- Variational approximations are a tool from physics
  that have been widely used in
  undirected networks.

- The **extension to directed graphs** requires new machinery but can be made for many networks.

- A variational approach for sigmoid belief networks has two main virtues:

  1. It provides a tractable approximation, $Q\{H|V\} \approx P\{H|V\}$, to the conditional distributions required for **inference**.

  2. It provides a rigorous lower bound on the likelihoods Z_V required for **learning**.
  3. It is also possible to develop upper bounds on the likelihoods, although we won't discuss upper bounds here
     (see Jaakkola & Jordan, UAI'96).

Bounding the likelihood

- Suppose we have an approximating distribution, $Q\{H|V\}$.

- A bound on Z_V follows from **Jensen's inequality**:

$$
\begin{aligned}
\ln Z_V &= \ln \Sigma_{\{H\}} \, P\{H,V\} \\
&= \ln \Sigma_{\{H\}} \, Q\{H|V\} \cdot \frac{P\{H,V\}}{Q\{H|V\}} \\
&\geq \Sigma_{\{H\}} \, Q\{H|V\} \ln \left[\frac{P\{H,V\}}{Q\{H|V\}} \right].
\end{aligned}
$$

- The difference between the left and right hand side is the **Kullback-Leibler divergence**:

$$
KL(Q||P) = \sum_{\{H\}} Q\{H|V\} \ln \left[\frac{Q\{H|V\}}{P\{H|V\}} \right].
$$

- Note: better approximations give better bounds.

Division of labor

- The bound has two basic components:

$$\begin{pmatrix} \textbf{mean field} \\ \textbf{entropy} \end{pmatrix} = -\Sigma_{\{H\}}\, Q\{H|V\} \ln Q\{H|V\}.$$

$$\begin{pmatrix} \textbf{mean field} \\ \textbf{energy} \end{pmatrix} = -\Sigma_{\{H\}}\, Q\{H|V\} \ln P\{H,V\}.$$

- What we need is the difference:

$$\ln Z \geq \begin{pmatrix} \textbf{mean field} \\ \textbf{entropy} \end{pmatrix} - \begin{pmatrix} \textbf{mean field} \\ \textbf{energy} \end{pmatrix}.$$

Variational principle

- The simplest variational approximation, which we will refer to as a **mean field** approximation considers the **factorized approximation**:

$$Q\{H|V\} = \Pi_i \; \mu_i^{S_i}(1-\mu_i)^{1-S_i}.$$

- The parameters μ_i are the mean values of S_i under $Q\{H|V\}$.

- We find the best approximation of this form by varying $\{\mu_i\}$ to minimize $KL(Q||P)$.

- This amounts to maximizing the lower bound:

$$\ln Z \geq \left(\begin{array}{c} \textbf{mean field} \\ \textbf{entropy} \end{array} \right) - \left(\begin{array}{c} \textbf{mean field} \\ \textbf{energy} \end{array} \right).$$

- Abuse of notation: we define **clamped** values (0 or 1) for the instantiated nodes,

$$\mu_i = S_i \text{ for } i \in V.$$

Mean field entropy

- The mean field entropy is:

$$-\Sigma_{\{H\}}\ Q\{H|V\} \ln Q\{H|V\}.$$

- Our **factorized approximation** is:

$$Q\{H|V\} = \Pi_i\ \mu_i^{S_i}(1-\mu_i)^{1-S_i},$$

- The **joint entropy** is the sum of the individual unit entropies:

$$-\Sigma_i\ \left[\mu_i \ln \mu_i + (1-\mu_i)\ln(1-\mu_i)\right].$$

Mean field energy

- The mean field energy is:

$$-\Sigma_{\{H\}}\, Q\{H|V\} \ln P\{H,V\}.$$

- In sigmoid belief networks:

$$\begin{aligned} \ln P\{H,V\} &= -\sum_{ij} J_{ij} S_i S_j - \sum_i h_i S_i \\ &\quad + \sum_i \ln\left[1 + e^{\Sigma_j J_{ij} S_j + h_i}\right]. \end{aligned}$$

- The first two terms are common to undirected networks. **The last term is not**.

- Averaging over $Q\{H|V\}$ gives:

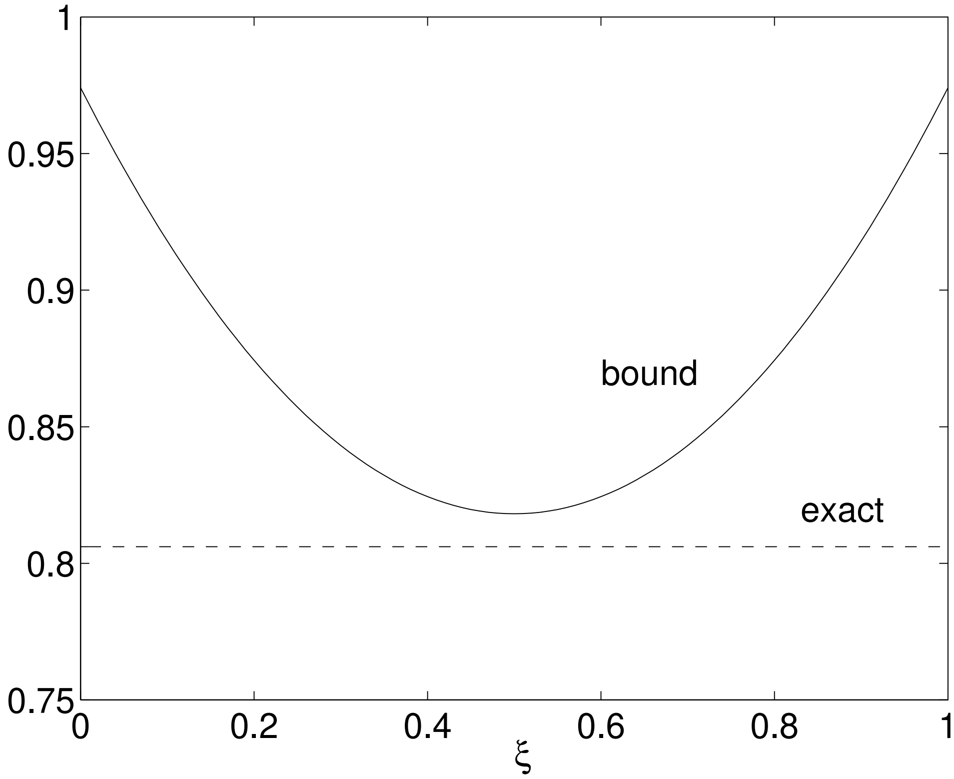
$$-\sum_{ij} J_{ij}\mu_i\mu_j - \sum_i h_i\mu_i + \sum_i \left\langle \ln\left[1 + e^{\Sigma_j J_{ij} S_j + h_i}\right]\right\rangle$$

- The last term is intractable, so (again) we use Jensen's inequality to obtain an upper bound...

- **Lemma** (Seung): for any random variable z, and
any real number ξ:

$$\langle \ln[1+e^z] \rangle \leq \xi\langle z\rangle + \ln\left\langle e^{-\xi z} + e^{(1-\xi)z}\right\rangle .$$

- **Ex**: z is Gaussian with zero mean and unit variance.



- **Ex**: z is the sum of weighted inputs to S_i;

$$z = \sum_j J_{ij}S_j + h_i.$$

The lemma provides an upper bound on the intractable terms in the mean field energy:

$$\sum_i \left\langle \ln\left[1 + e^{\sum_j J_{ij}S_j + h_i}\right]\right\rangle$$

Mean field equations

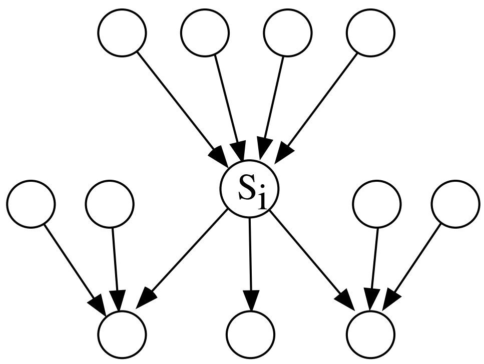
- The bound on the log-likelihood,

$$\ln Z \geq \begin{pmatrix} \textbf{mean field} \\ \textbf{entropy} \end{pmatrix} - \begin{pmatrix} \textbf{mean field} \\ \textbf{energy} \end{pmatrix},$$

  is valid for any setting of the mean field parameters, $\{\mu_i\}$.

- The optimal $\{\mu_i\}$ are found by solving the mean field equations:

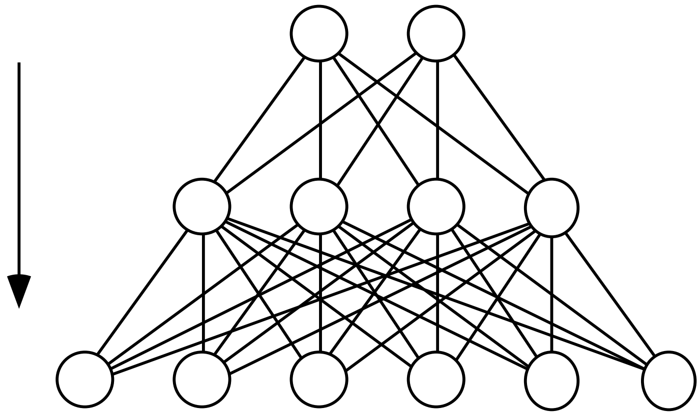
$$\mu_i = \sigma\left(h_i + \sum_j \left[J_{ij}\mu_j + J_{ji}(\mu_j - \xi_j) - K_{ji}\right]\right).$$



- The effective input to S_i is composed of terms from its **Markov blanket**.

Numerical experiments

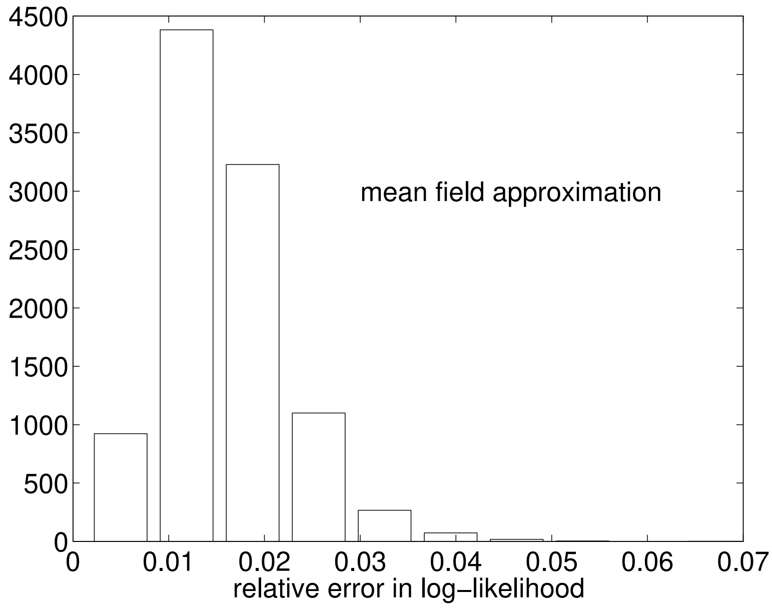
- For small networks, the mean field bound can be compared to the true likelihood obtained by exact enumeration.



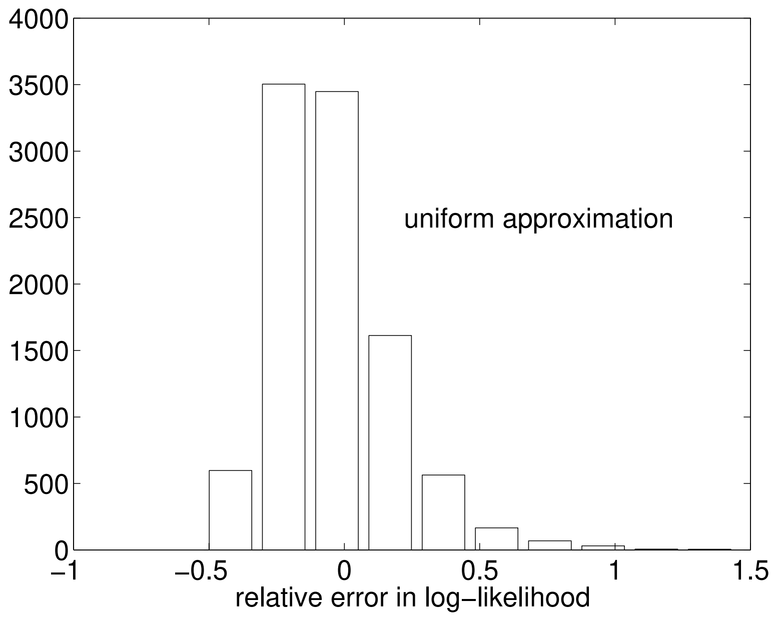
- We considered the event that all the units in the bottom layer were inactive.

- This was done for 10000 random networks whose weights and biases were uniformly distributed between -1 and 1.

- **Mean field approximation:**

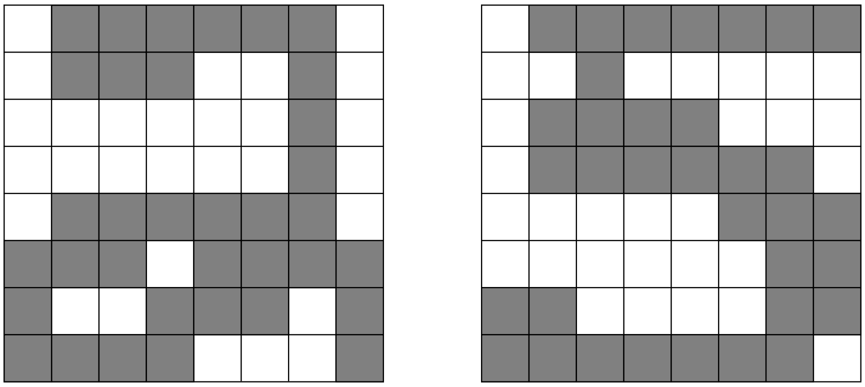


- **Uniform approximation:**



Digit recognition

- Images:



- Confusion matrix:

	0	1	2	3	4	5	6	7	8	9
0	388	2	2	0	1	3	0	0	4	0
1	0	393	0	0	0	1	0	0	6	0
2	1	2	376	1	3	0	4	0	13	0
3	0	2	4	373	0	12	0	0	6	3
4	0	0	2	0	383	0	1	2	2	10
5	0	2	1	13	0	377	2	0	4	1
6	1	4	2	0	1	6	386	0	0	0
7	0	1	0	0	0	0	0	388	3	8
8	1	9	1	7	0	7	1	1	369	4
9	0	4	0	0	0	0	0	8	5	383

classification error: less than 5%

Extensions

- discrete-valued units

$$S_i \in \{1, 2, \ldots, n\}$$

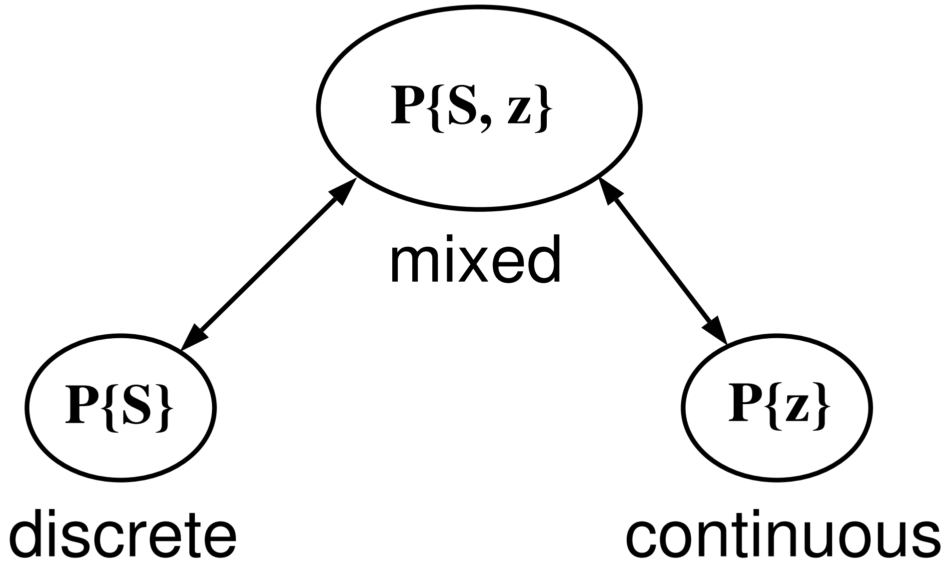
- other activation functions

$$\textbf{Ex:}\;\; P(S_i = 1|z) = \frac{1}{\sqrt{\pi}} \int^z dx\; e^{-x^2}$$

- continuous variables

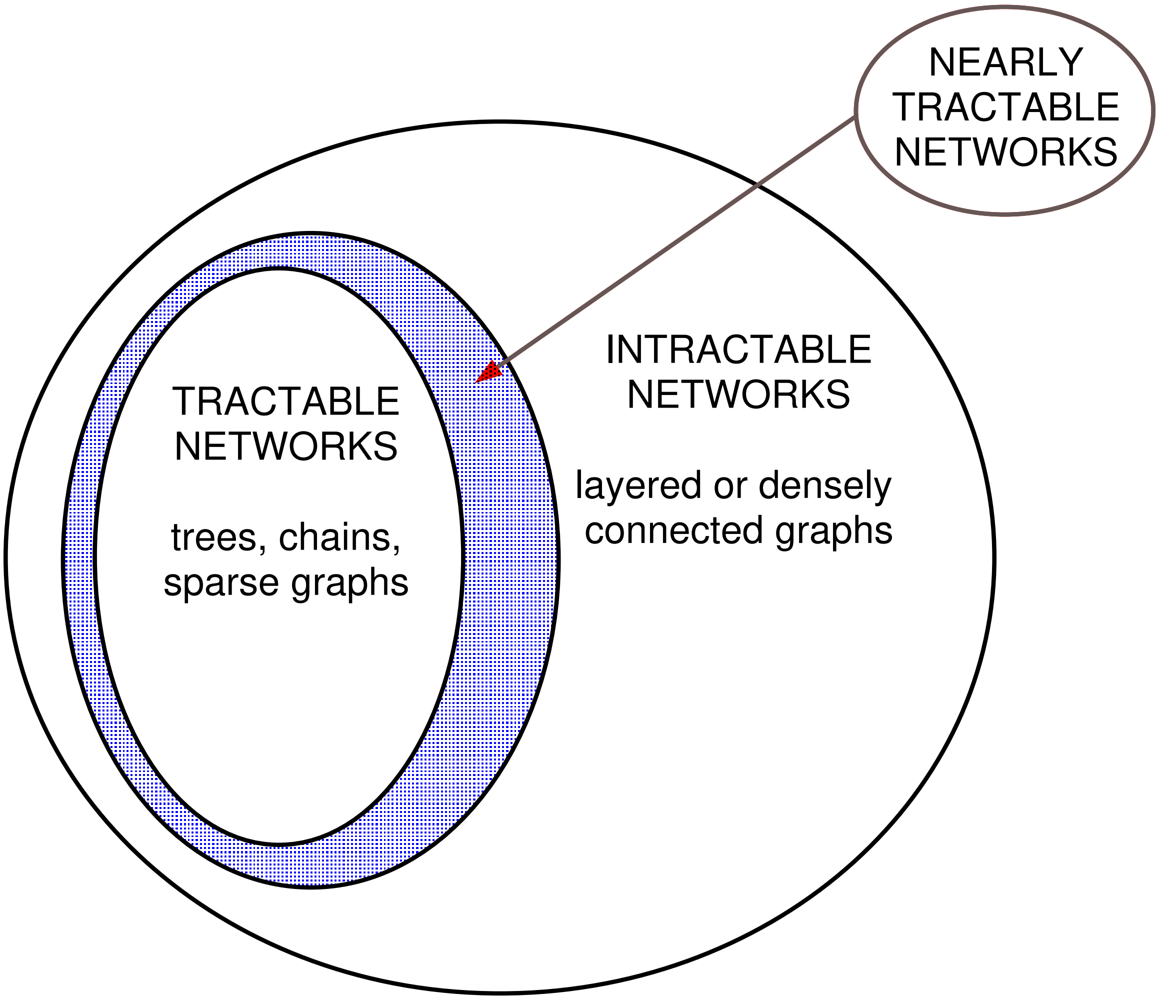
$$\textbf{Ex:}\;\; P\left(z_i | \{z_{j<i}\}\right) \text{ is non-gaussian}$$

- duality



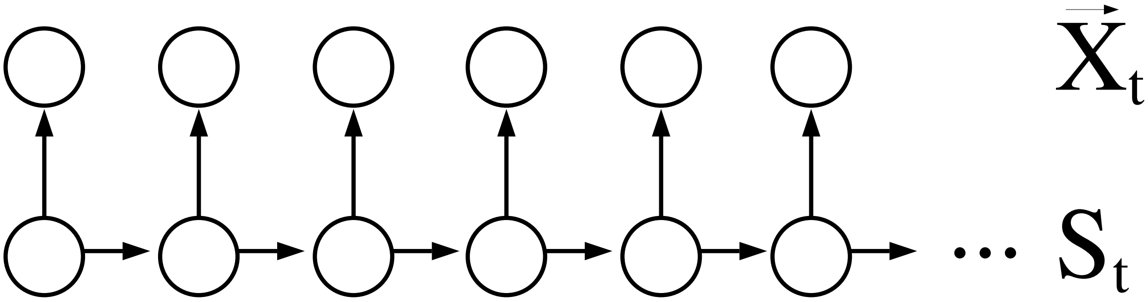
Exploiting tractable substructures

(Saul & Jordan, 1995)

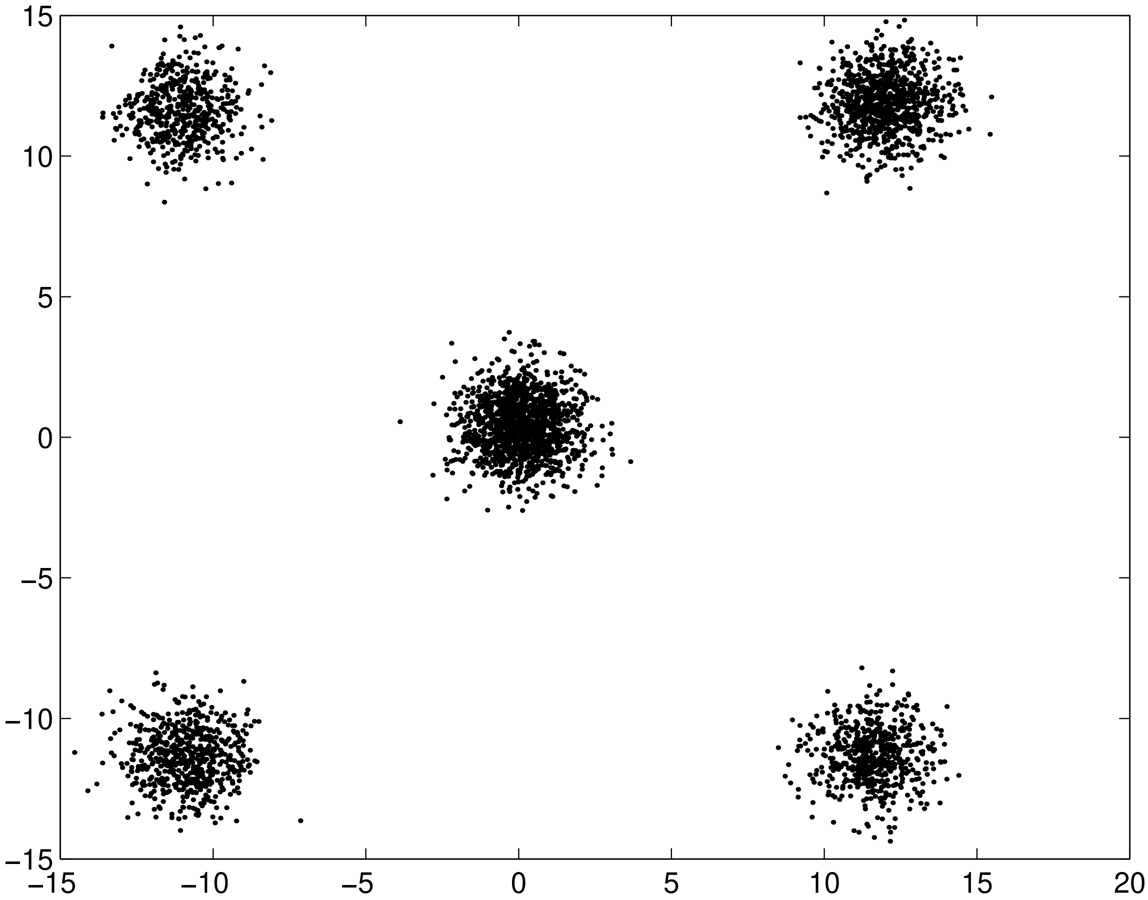


- For networks with tractable substructures, we can use a **refined** variational approximation...

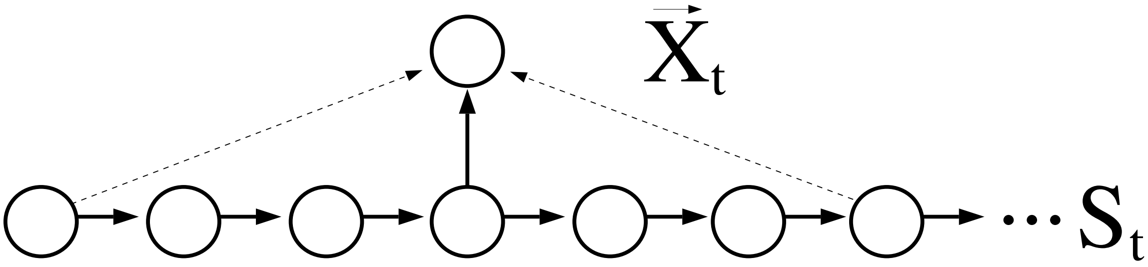
Hidden Markov model



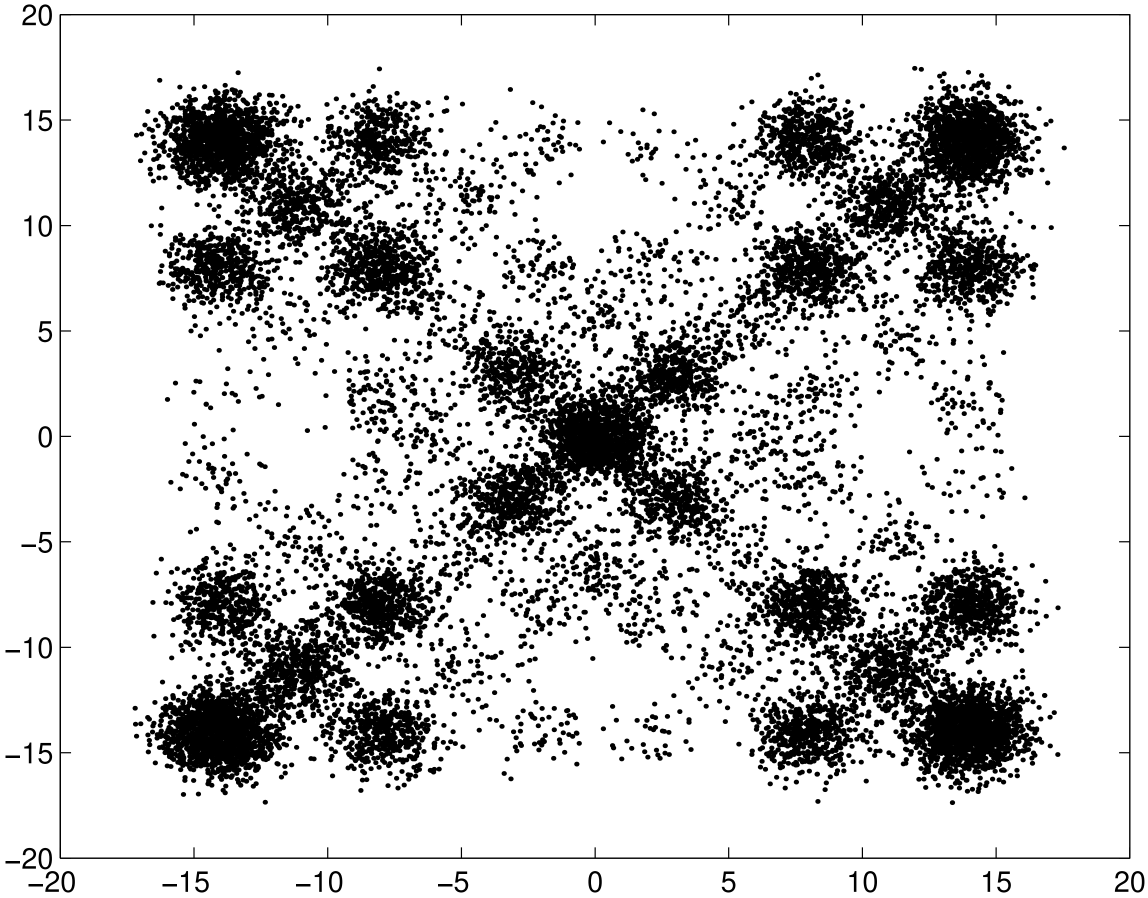
- The observation $\vec{X}_t$ depends on the hidden state S_t at time t.



HMM with context

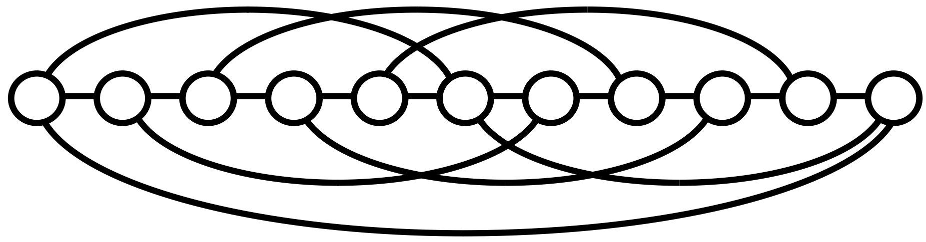


- The observation $\vec{X}_t$ depends on the hidden states at times t and $t \pm \Delta$.



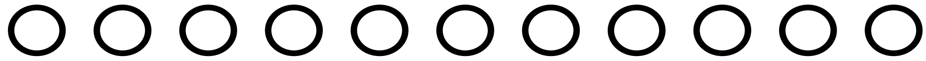
Exploiting tractable substructures

- **undirected graph** for context HMM



$$P\{S|\vec{X}\} = \frac{1}{Z}\exp\left\{\Sigma_t\left[JS_tS_{t+1} + h_tS_t + KS_tS_{t+\Delta} + \ldots\right]\right\}$$

- **naive** variational approximation



$$Q\{S|\vec{X}\} = \Pi_i\ \mu_i^{S_i}(1-\mu_i)^{1-S_i}$$

- **refined** variational approximation

$$Q\{S|\vec{X}\} = \frac{1}{Z_Q}\exp\left\{\Sigma_t\left[JS_tS_{t+1} + H_tS_t\right]\right\}$$

Other HMM variations

- Factorial HMM (Ghahramani & Jordan, 1995)
- HMM with long-range connections (Saul & Jordan, 1995)
- Hidden Markov decision trees (Jordan, Ghahramani & Saul, 1996)

Can all be tackled with variational algorithms...

Summary

- **Approximations** are required for inference and learning in intractable networks.

- **Variational methods** provide:
  1. lower and upper bounds on the posterior distributions required for **inference**.
  2. lower bounds on the likelihoods required for **learning**.

- Variational approximations can be used in **directed graphs** with discrete and/or continuous variables.

- **Refined** variational approximations can exploit the presence of tractable substructures.

Final conclusions

- probabilistic networks with graphical model semantics
- HMM's and HME's are special cases (chains and trees)
- EM as a unifying framework for parameter estimation
  - exact methods
  - approximate methods
- statistical physics tools

http

Home Page:

$\boxed{\text{http://www.ai.mit.edu/projects/jordan.html}}$