Machine Learning

CSCI 5622 Fall 2020

Prof. Claire Monteleoni

Today

- Intro. to Generative Learning / probabilistic models
  - Intro to Bayesian Networks
  - Naïve Bayes Model
  - Hidden Markov Model

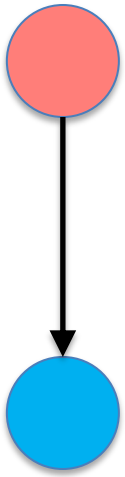
with much credit to S. Dasgupta and T. Jaakkola

Intro. to Probabilistic Graphical Models

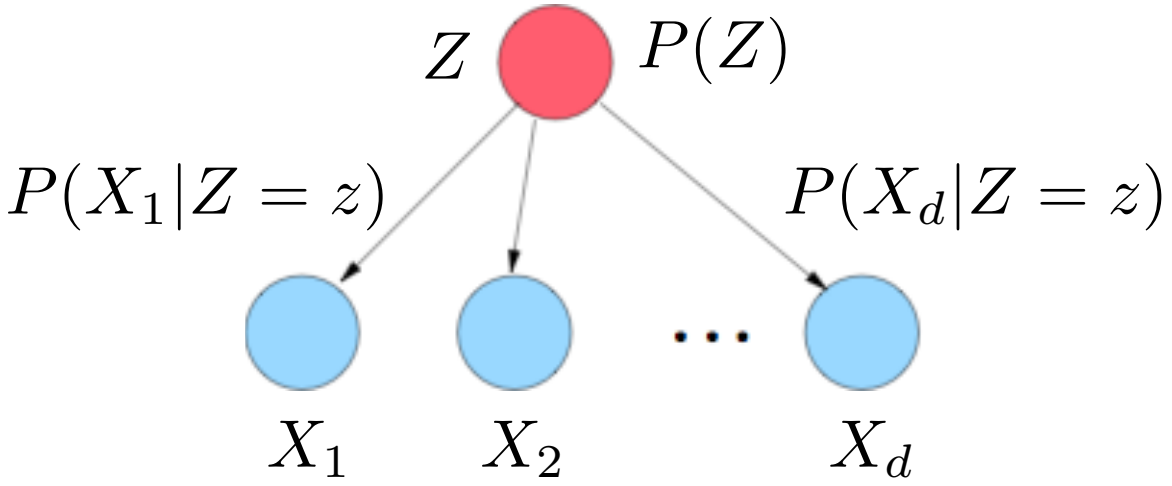
- A.k.a. Bayesian networks

We have already seen one, we just hadn't drawn its graph yet.

Mixture Model:



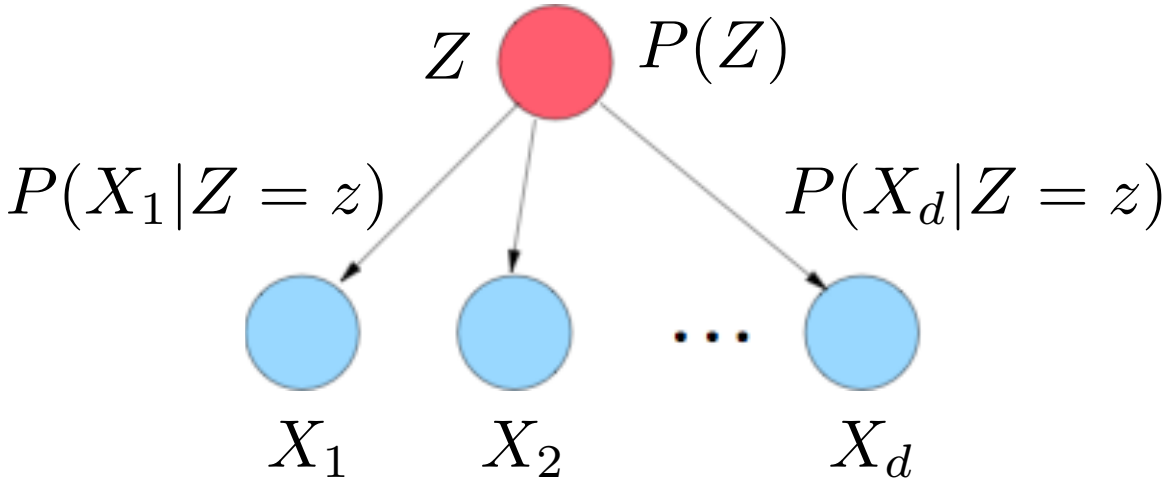
Graphical models / Bayes Nets



This is an example of a Graphical Model (a.k.a. Bayesian Network, Bayes Net, Probabilistic Graphical Model)

- Nodes are random variables
- Edges are conditional probability distributions

Graphical models / Bayes Nets

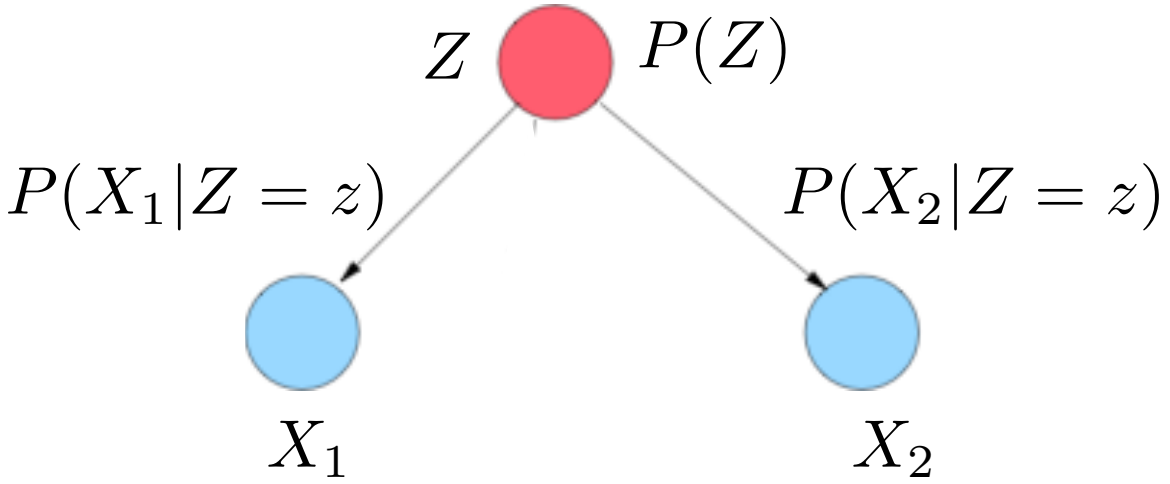


A graphical model factors the joint probability distribution (over all random variables)

→ Much fewer table entries are needed than in the full joint!

This is subject to the (conditional) independence assumptions specified by the graph.

Graphical models / Bayes Nets



Example: Binary random variables: $z, x_i, x_2 \in \{0, 1\}$

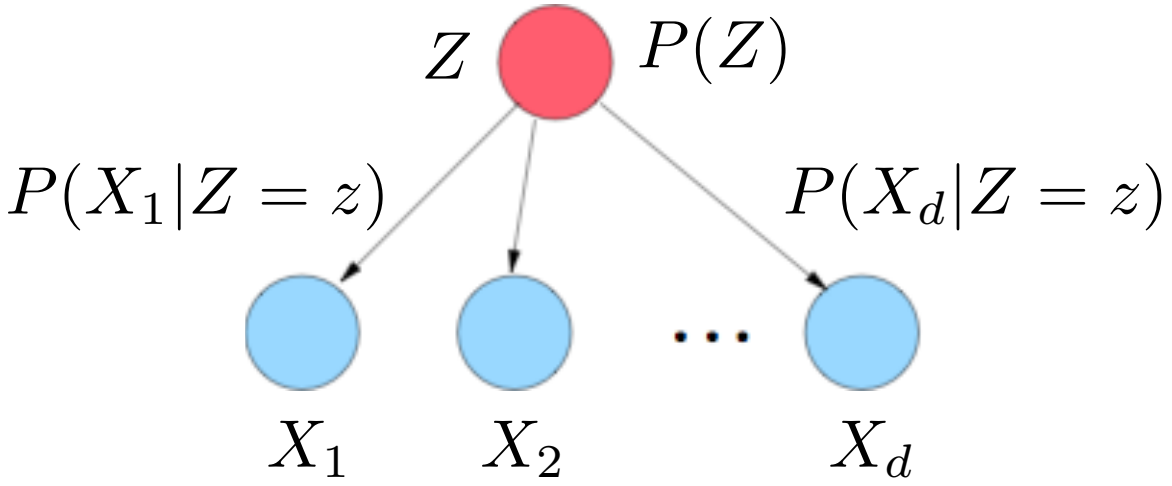
- The joint probability table $P(X_1, X_2, Z)$ has $2^3 = 8$ entries.
- Assuming the graph's independence structure, the joint factors into:

$$P(X_1, X_2, Z) = P(Z)P(X_1|Z)P(X_2|Z)$$

- To specify this Bayes Net requires only 5 parameters:

$$p(Z = 1)\, p(X_1 = 1|Z = 1), p(X_1 = 1|Z = 0)$$
$$p(X_2 = 1|Z = 1), p(X_2 = 1|Z = 0)$$

Naïve Bayes model



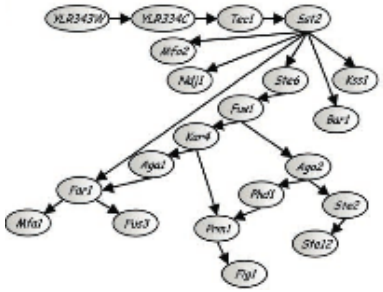
A k-component mixture over d features X_i that assumes conditional independence among the X_i's, given Z:

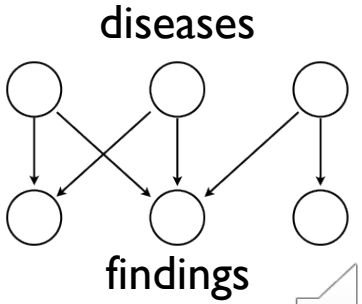
Fix the value of Z = z,

then X_i is independent of X_j for all i,j $\in \{1, \ldots, d\}$

Bayesian networks

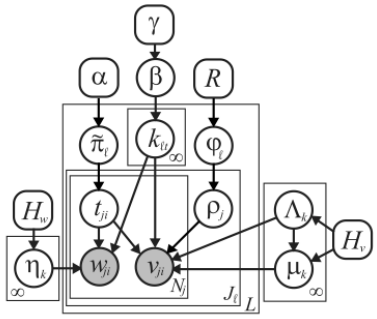
- Bayesian networks provide a simple language for succinctly expressing, using, and learning probabilistic information
- In a Bayesian network, nodes correspond to random variables and directed edges indicate dependencies
  - the graph provides a qualitative description of how the variables relate to each other
  - the probability distribution underlying the graph quantifies numerically how the variables depend on each other
- Both the graph structure and the associated distribution are important in applications
  - diagnostic tasks, medicine
  - data analysis in biology
  - etc.

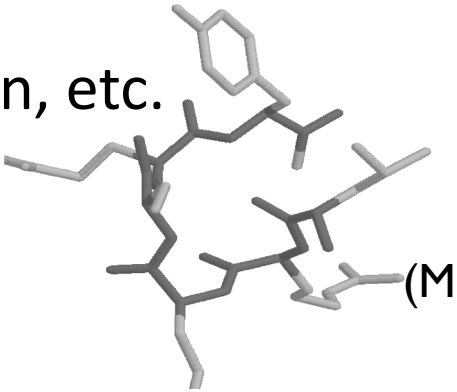


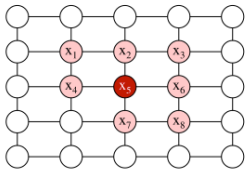


Applications of Bayes Nets

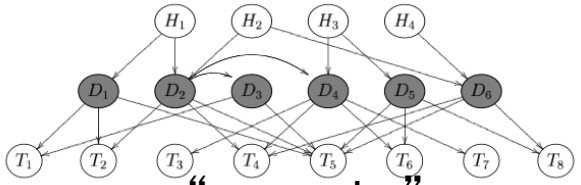
- Computer vision
  - scene analysis, stereo matching, etc.
- Molecular biology
  - networks, molecular design, etc.
- Information retrieval / NLP
  - topic models, parsing
- Diagnosis
  - fault, medical
- Signal processing
  - deconvolution, sensor networks
- Communication
  - decoding algorithms



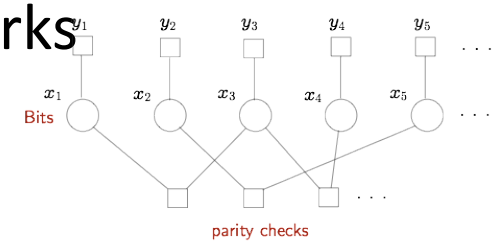




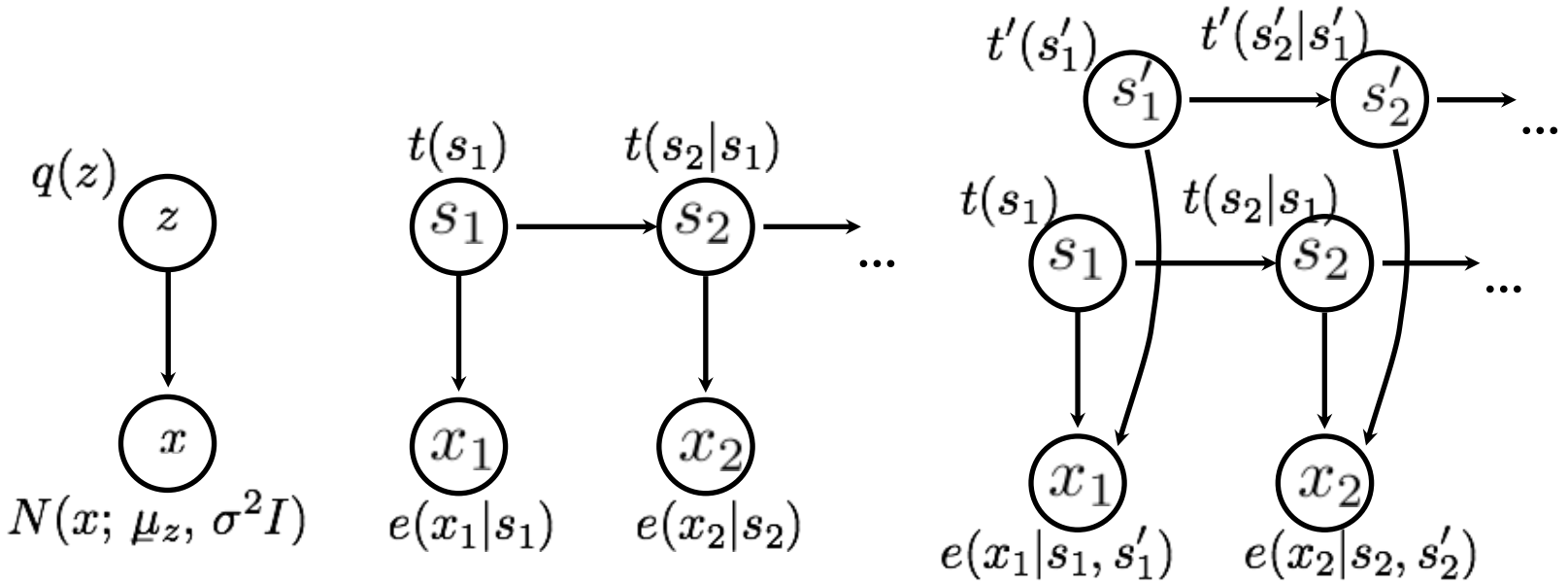
(Meltzer et al.)



"promedas"



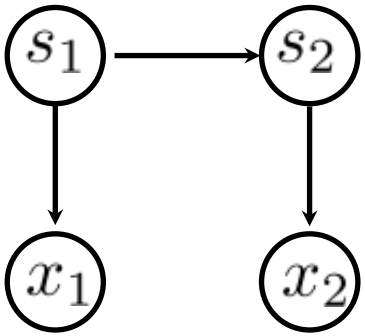
Graphs and probabilities



- Bayesian networks as probability models are represented by directed acyclic graphs (DAGs) where the nodes specify variables and the directed edges identify dependencies
- Formally, the graph encodes conditional independence properties about the variables (cf. Markov properties)
- Any probability distribution we associate with the graph has to be consistent with such independence properties

Bayesian networks

- Some basic terminology:
  - s_1 is a "parent" of x_1
  - s_1 and s_2 are "ancestors" of x_2; x_1 is not.
  - x_1 is a "child" of s_1
  - x_1, s_2 and x_2 are all "descendants" of s_1



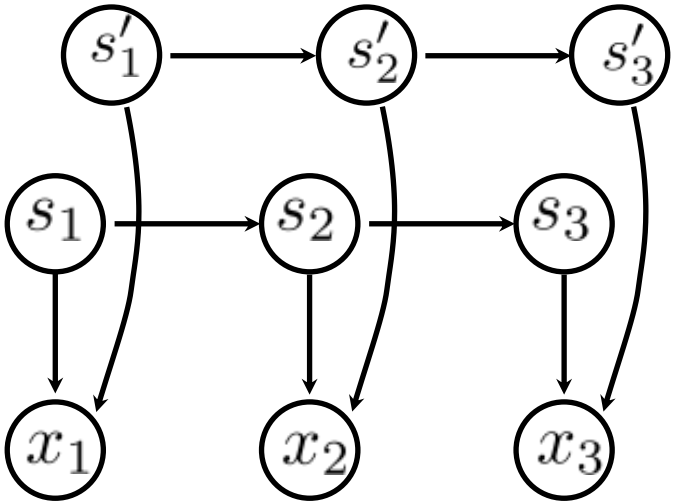
- Joint probability distribution factors as:

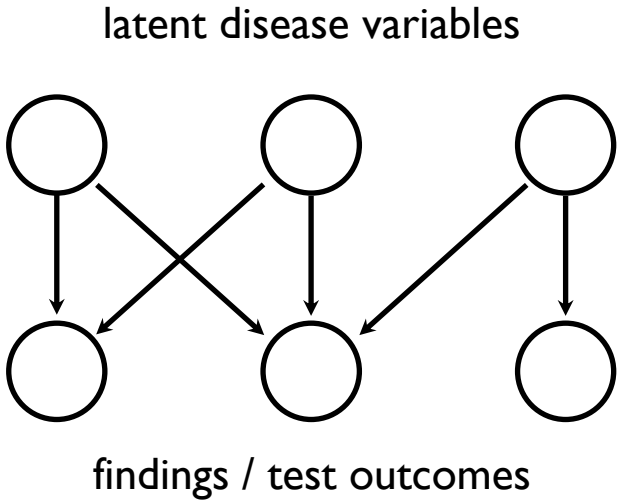
$$P(V_1, V_2, \ldots, V_k) = \prod_{i=1}^{k} P(v_i | \texttt{Parents}(V_i))$$

Bayesian networks

Examples:

- a factorial HMM, where observations depend on two Markov chains running in parallel (two speakers, one microphone)
- a two layer model for medical diagnosis





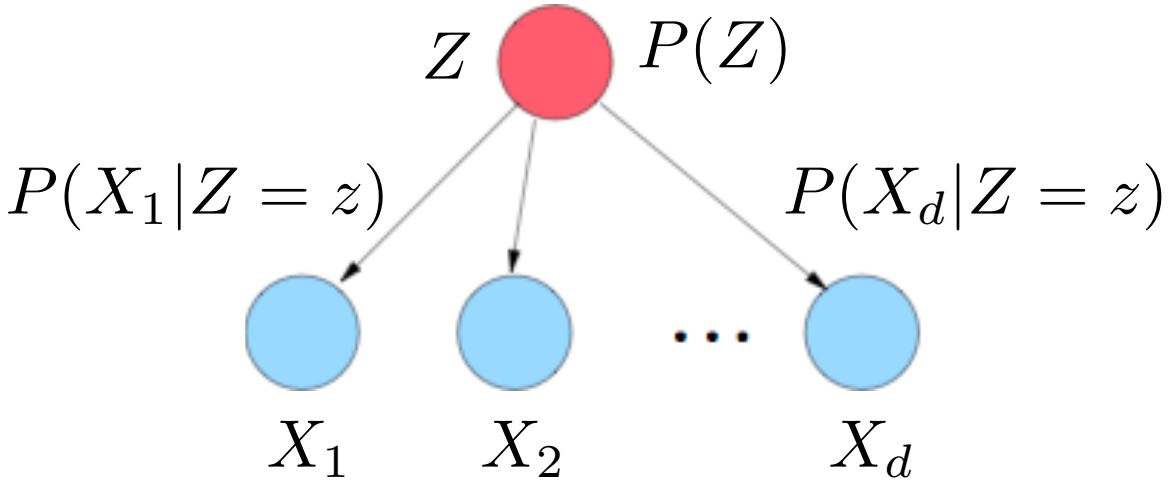
Learning Bayesian networks

$$D = \begin{array}{cccccccccc} x_1 & x_2 & & & \cdots & & & & x_n \\ 2 & 2 & 1 & 1 & 1 & 2 & 1 & 1 & 2 & 1 & 2 \\ 1 & 2 & 2 & 2 & 2 & 2 & 3 & 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 3 & 1 & 1 & 2 & 3 & 3 & 2 & 3 & 2 \\ 1 & 2 & 2 & 2 & 2 & 2 & 3 & 1 & 1 & 2 & 2 \\ 1 & 1 & 3 & 1 & 1 & 1 & 3 & 1 & 1 & 1 & 3 \\ & & & & \cdots & & & & & & \end{array}$$

complete data

- **Parameter estimation:** find the maximum likelihood (or Bayesian) estimates of parameters for a graph G
- **Model selection:** appropriately score each G based on its degree of fit to the data
- **Structure search:** find the highest scoring structure $\hat{G}$

Naïve Bayes model



Estimation from labeled training data: Filter the data by labels z:

For each label z, fit a probability model, $P(X_i|Z)$, for each X_i.

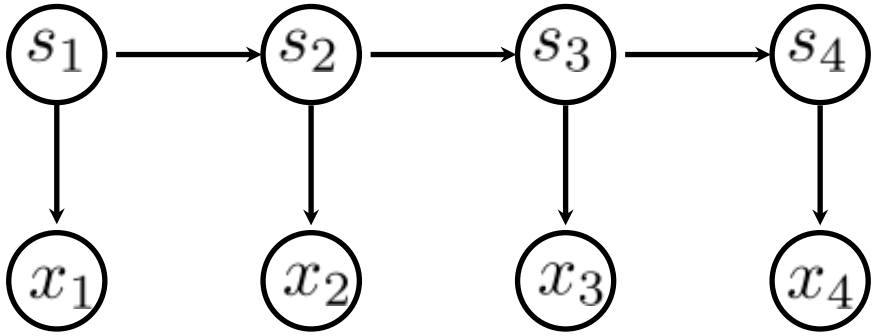
E.g. if $P(X_i|Z)$ is modeled with a Gaussian, then for each i, just need to compute $\hat{\mu}_i, \hat{\sigma}_i^2$ from the training data.

Prediction on a new example $x = (x_1, \ldots, x_d)$:

$$\hat{z} = \arg\max_z P(Z|x_1, \ldots, x_d) = \arg\max_z P(Z) \prod_{i=1}^{d} P(x_i|Z)$$

The Hidden Markov Model (HMM)

- Each node corresponds to a variable, either state s_t or observation x_t, and the arcs represent dependencies such as "s_2 depends on s_1"



HMMs

- Hidden Markov models are widely used models
- speech recognition
  - e.g., each word is a Markov sequence of phonemes which are then coupled to acoustic measurements
- computational biology
  - e.g., gene structure labels (coding region, etc.) form a Markov chain of states and each state generates a base-pair (sequence)
- natural language processing
  - e.g., part of speech tags (noun, verb, etc) form a Markov chain and the tags give rise to observed words

Example: topic / document models

Modeling documents, e.g. news stories, webpages, tweets.

- We view each document d as a sequence of words $\{w_1, \ldots, w_N\}$ out of some restricted vocabulary $\mathcal{W}$. E.g.,

$$d = \{\text{White,House,officials,consulted,with,...}\}$$

- To generate the words in the document we could simply sample each word independently from some multinomial distribution $P(w;\theta) = \theta_w$, $\sum_{w \in \mathcal{W}} \theta_w = 1$ so that

$$P(d;\theta) = \prod_{w \in d} \theta_w$$

... but all the documents would look the same according to this model.

Document models

- Documents pertain to different topics so we should introduce different word distributions, one for each topic $z = 1, \ldots, k$

$$P(w|z;\theta) = \theta_{w|z}, \quad \sum_{w \in \mathcal{W}} \theta_{w|z} = 1$$

- The document d could then be viewed as a mixture

$$P(d;\theta) = \sum_{z=1}^{k} P(z;\theta) P(d|z;\theta) = \sum_{z=1}^{k} \theta_z \prod_{w \in d} \theta_{w|z}$$

Document models

- Documents pertain to different topics so we should introduce different word distributions, one for each topic $z = 1, \ldots, k$

$$P(w|z;\theta) = \theta_{w|z}, \quad \sum_{w \in \mathcal{W}} \theta_{w|z} = 1$$

- The document d could then be viewed as a mixture

$$P(d;\theta) = \sum_{z=1}^{k} P(z;\theta) P(d|z;\theta) = \sum_{z=1}^{k} \theta_z \prod_{w \in d} \theta_{w|z}$$

... but each document is still associated with a single topic

$z \sim \theta_z$ sampled once for the document

$w_i \sim \theta_{w|z}, \ i = 1, \ldots, N$ uses the same topic

Simple topic models

- Each document may involve many topics. We could allow each word in a document to have a different (independently chosen) topic

$$\begin{array}{cccc} z_1, & z_2, & \ldots, & z_N \\ w_1, & w_2, & \ldots, & w_N \end{array}$$

- As a result, each word is generated from a mixture

$$P(w;\theta) = \sum_{z=1}^{k} \theta_z \, \theta_{w|z}$$

Simple topic models

- Each document may involve many topics. We could allow each word in a document to have a different (independently chosen) topic

$$\begin{array}{cccc} z_1, & z_2, & \ldots, & z_N \\ w_1, & w_2, & \ldots, & w_N \end{array}$$

- As a result, each word is generated from a mixture

$$P(w;\theta) = \sum_{z=1}^{k} \theta_z \, \theta_{w|z}$$

and the probability of the words in a document is given by

$$P(d;\theta) = \prod_{w \in d} P(w;\theta) = \prod_{w \in d} \sum_{z=1}^{k} \theta_z \, \theta_{w|z}$$

θ_z = topic usage (per document)
$\theta_{w|z}$ = topic definitions (same across documents)

Simple topic models

Document d_1 $P(d_1|1;\theta) = \prod_{w \in d_1} \sum_{z=1}^{k} \theta_{z|1}\, \theta_{w|z}$

Document d_2 $P(d_2|2;\theta) = \prod_{w \in d_2} \sum_{z=1}^{k} \theta_{z|2}\, \theta_{w|z}$

Document d_3 $P(d_3|3;\theta) = \prod_{w \in d_3} \sum_{z=1}^{k} \theta_{z|3}\, \theta_{w|z}$

different

same

...

- Note that the topics $\theta_{w|z}$ are estimated across documents while the topic usage $\theta_{z|t}$ is linked to a single document.

  (In LDA, we would integrate over $\theta_{z|t}$ per document. As a result, there would be no document specific parameters)

Sequence models

- We modeled a document as a sequence of topics and the corresponding words

$$\begin{array}{cccc} z_1, & z_2, & \ldots, & z_N \\ w_1, & w_2, & \ldots, & w_N \end{array}$$

- We made strong assumptions about how they are generated

$$P(z_1, \ldots, z_N, w_1, \ldots, w_N) =$$

Sequence models

- We modeled a document as a sequence of topics and the corresponding words

$$\begin{array}{cccc} z_1, & z_2, & \ldots, & z_N \\ w_1, & w_2, & \ldots, & w_N \end{array}$$

- We made strong assumptions about how they are generated

$$P(z_1, \ldots, z_N, w_1, \ldots, w_N) = P(w_1, \ldots, w_N | z_1, \ldots, z_N) P(z_1, \ldots, z_N)$$

Sequence models

- We modeled a document as a sequence of topics and the corresponding words

$$\begin{array}{cccc} z_1, & z_2, & \ldots, & z_N \\ w_1, & w_2, & \ldots, & w_N \end{array}$$

- We made strong assumptions about how they are generated

$$P(z_1, \ldots, z_N, w_1, \ldots, w_N) = P(w_1, \ldots, w_N | z_1, \ldots, z_N) P(z_1, \ldots, z_N)$$
$$\stackrel{(1)}{=} \left[\prod_{i=1}^{N} P(w_i | z_i) \right] P(z_1, \ldots, z_N)$$

Sequence models

- We modeled a document as a sequence of topics and the corresponding words

$$\begin{array}{cccc} z_1, & z_2, & \ldots, & z_N \\ w_1, & w_2, & \ldots, & w_N \end{array}$$

- We made strong assumptions about how they are generated

$$P(z_1, \ldots, z_N, w_1, \ldots, w_N) = P(w_1, \ldots, w_N | z_1, \ldots, z_N) P(z_1, \ldots, z_N)$$

$$\stackrel{(1)}{=} \left[\prod_{i=1}^{N} P(w_i | z_i) \right] P(z_1, \ldots, z_N)$$

$$\stackrel{(2)}{=} \left[\prod_{i=1}^{N} P(w_i | z_i) \right] \left[\prod_{i=1}^{N} P(z_i) \right]$$

1) words are independent given topics, depend only on the current topic, 2) topics are independent along the sequence

Markov model

- We could extend the simple independent topic model by "coupling" the choice of topics along the sequence

$$z_1, \quad z_2, \quad \ldots, \quad z_N$$
$$w_1, \quad w_2, \quad \ldots, \quad w_N$$

- For example, we could relate successive topic choices by assuming that they are governed by a Markov chain

$$P(z_1, \ldots, z_N) = P(z_1)P(z_2|z_1)P(z_3|z_1, z_2) \cdots P(z_N|z_{N-1}, \ldots, z_1)$$
$$\stackrel{(2)}{=} P(z_1)P(z_2|z_1)P(z_3|z_2) \cdots P(z_N|z_{N-1})$$

where in 2) we assume that each topic selection can depend on the preceding selection but not further back.

On Markov chains

- The Markov chain is homogeneous if the "transition probabilities" $P(z_i|z_{i-1})$ do not depend on the position i along the sequence.

  In this case, we can parameterize the chain

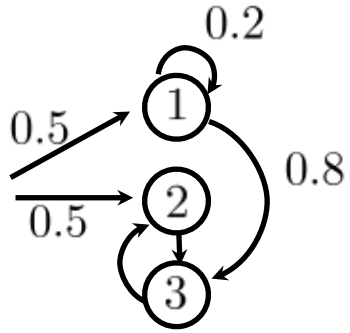
  $$P(z_1, \ldots, z_N; \theta) = q(z_1) \prod_{i=2}^{N} q(z_i|z_{i-1})$$

  by specifying the initial "state" (here topic) distribution $q(z_1)$ and the state transition probability matrix $q(z_i|z_{i-1})$ that is reused along the sequence

Markov Chains: representation

- A state transition diagram: nodes correspond to "states" (here topics), arcs represent possible transitions between states

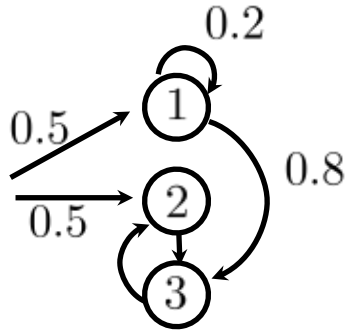
$$z_i = 1, \ldots, 3$$



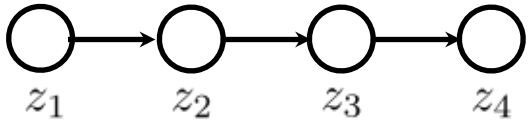
Markov Chains: representation

- A state transition diagram: nodes correspond to "states" (here topics), arcs represent possible transitions between states

$$z_i = 1, \ldots, 3$$



- A graphical model: nodes in the graph correspond to variables (state/topic selections) and arcs represent how the variables depend on each other



On Markov Chains

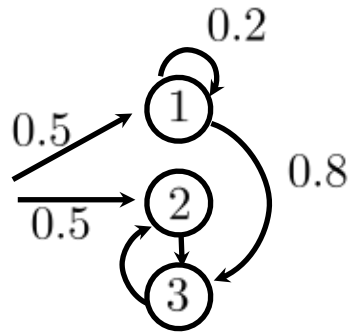
- Given any initial state distribution $q(z_1)$ and the state transition probability matrix $q(z_i|z_{i-1})$, we can generate sample sequences from the corresponding homogeneous Markov chain according to

$$z_1 \sim q(z_1),$$

$$z_i \sim q(z_i|z_{i-1}), \quad \text{where } z_{i-1} \text{ is fixed}, \quad i = 2, \ldots, N$$

(Note: we do not model the length of the sequence)

$$z_i = 1, \ldots, 3$$



Hidden Markov Model

- Only the words are observed in our (topic,word) sequence

$$\begin{array}{cccc} z_1, & z_2, & \ldots, & z_N \\ w_1, & w_2, & \ldots, & w_N \end{array}$$

- The underlying Markov Chain that we defined over topics is therefore latent or hidden

Hidden Markov Model

- Only the words are observed in our (topic,word) sequence

$$\begin{array}{cccc} z_1, & z_2, & \ldots, & z_N \\ w_1, & w_2, & \ldots, & w_N \end{array}$$

- The underlying Markov Chain that we defined over topics is therefore latent or hidden
- If the observations (words) only depend on the current state (current topic), the resulting model is known as Hidden Markov Model (HMM)

$$P(w_1, \ldots, w_N, z_1, \ldots, z_N) =$$

$$= q(z_1)P(w|z_1)\prod_{i=2}^{N} q(z_i|z_{i-1})P(w_i|z_i)$$

HMM construction

- Two ways to think about HMMs

$$z_1, \quad z_2, \quad \ldots, \quad z_N$$
$$w_1, \quad w_2, \quad \ldots, \quad w_N$$

1) We start with independent mixture models for the observations along the sequence, then couple the choices (types,states)
2) We construct a Markov Model for the underlying (abstract) state sequence and then relate each state to possible observables (hide the Markov chain)

HMM definitions

- **State:** state $s = 1, \ldots, k$ is an abstraction such as a topic, phoneme, genomic label, or a part of speech tag.

HMM definitions

- **State:** state $s = 1, \ldots, k$ is an abstraction such as a topic, phoneme, genomic label, or a part of speech tag.
- **Markov chain:** we model the sequence of states using a first order Markov model with parameters

$$q(s_1) = \text{initial state distribution}$$

$$q(s_i|s_{i-1}) = \text{state transition matrix}$$

HMM definitions

- **State:** state $s = 1, \ldots, k$ is an abstraction such as a topic, phoneme, genomic label, or a part of speech tag.
- **Markov chain:** we model the sequence of states using a first order Markov model with parameters

$$q(s_1) = \text{initial state distribution}$$

$$q(s_i|s_{i-1}) = \text{state transition matrix}$$

- **Output/emission probabilities:** each state s is associated with a distribution over observables $x = 1, \ldots, m$ or $\underline{x} \in \mathcal{R}^d$ and parameters

Examples:

$q_e(x|s) = \text{matrix of probabilities} \quad \sum_{x=1}^{m} q_e(x|s) = 1$

$q_e(\underline{x}|s) = N(\underline{x};\, \mu_s, \Sigma_s)$ state dependent Gaussian in $\mathcal{R}^d$

HMM examples

- Consider an HMM with binary state and binary observations such that (rows of transition and emission probabilities sum to one)

$$q(s_1) = 1/2, \;\; q(s_2|s_1) \; : \; \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

$$q_e(x|s) \; : \; \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$

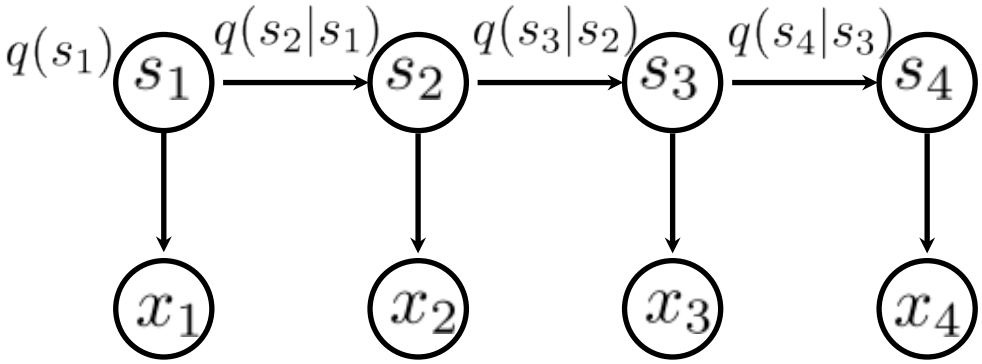
- Consider an HMM with binary state and Gaussian observations

$$q(1) = 1, \;\; q(s_2|s_1) \; : \; \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$q_e(x|s) = N(x; \, s, \sigma^2)$$

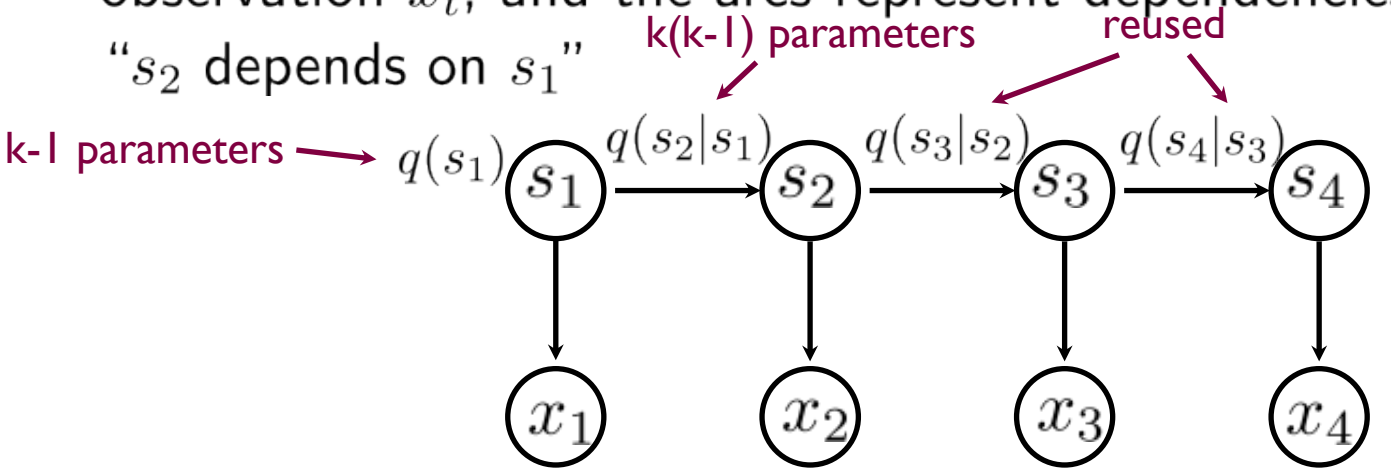
HMM

- Each node corresponds to a variable, either state s_t or observation x_t, and the arcs represent dependencies such as "s_2 depends on s_1"



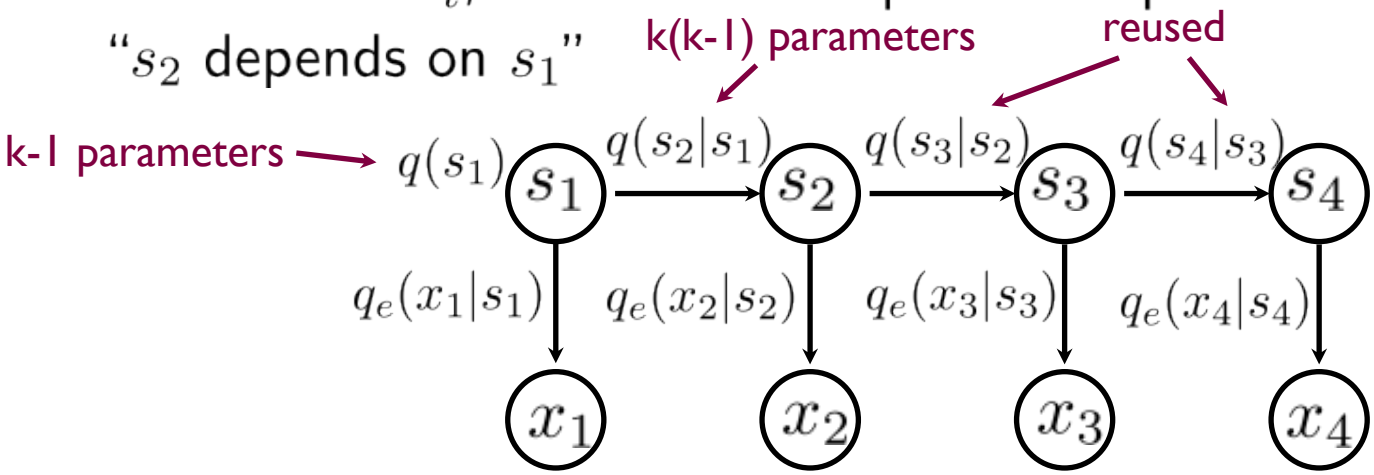
HMM

- Each node corresponds to a variable, either state s_t or observation x_t, and the arcs represent dependencies such as "s_2 depends on s_1"



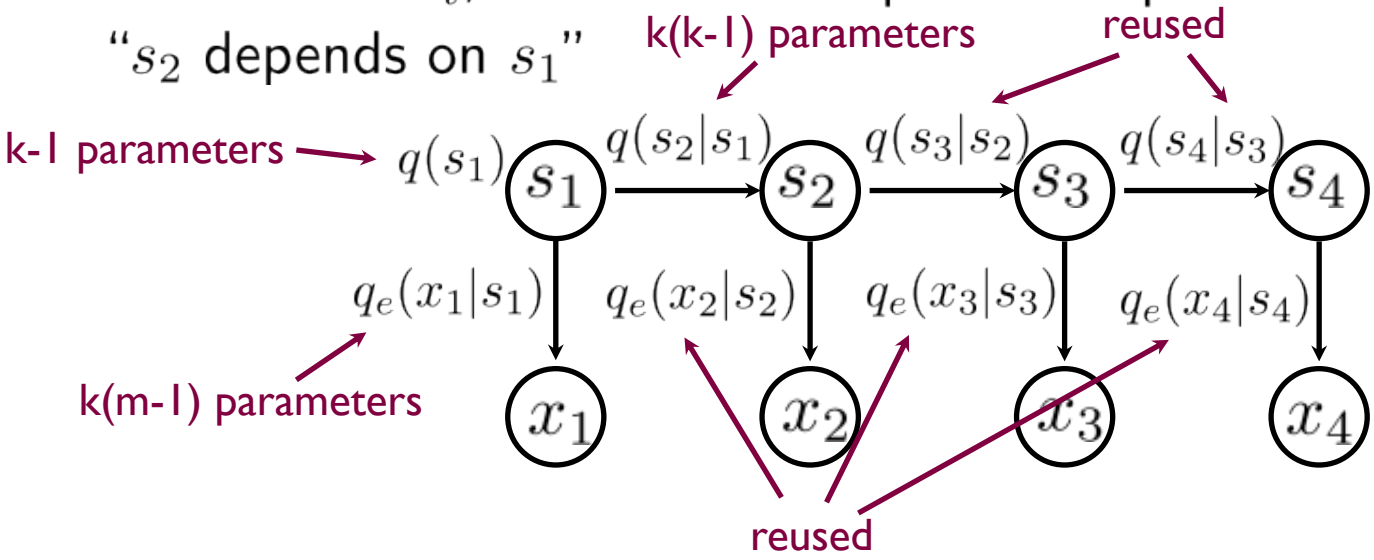
HMM

- Each node corresponds to a variable, either state s_t or observation x_t, and the arcs represent dependencies such as "s_2 depends on s_1"



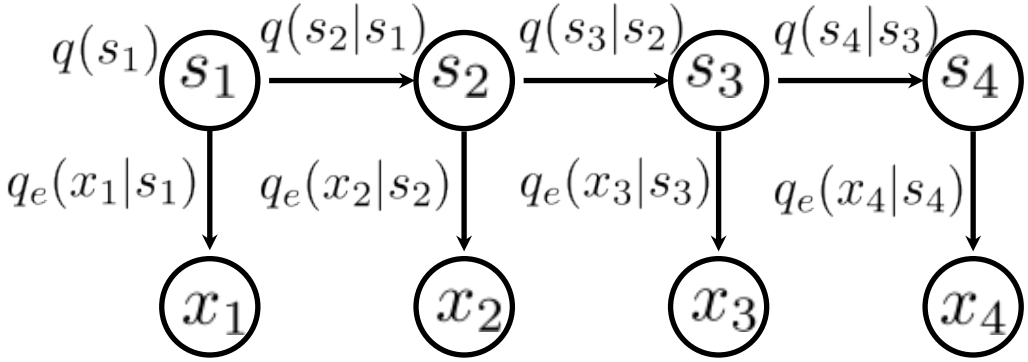
HMM

- Each node corresponds to a variable, either state s_t or observation x_t, and the arcs represent dependencies such as "s_2 depends on s_1"



HMM

- Each node corresponds to a variable, either state s_t or observation x_t, and the arcs represent dependencies such as "s_2 depends on s_1"



- The joint distribution is simply the product of the terms in the graph [Since this is a probabilistic graphical model].

$$P(x_1, \ldots, x_4, s_1, \ldots, s_4; \theta) = q(s_1) q_e(x_1|s_1) \prod_{t=2}^{4} q(s_t|s_{t-1}) q_e(x_t|s_t)$$

HMM problems

- Three problems we need to solve in the context of HMMs

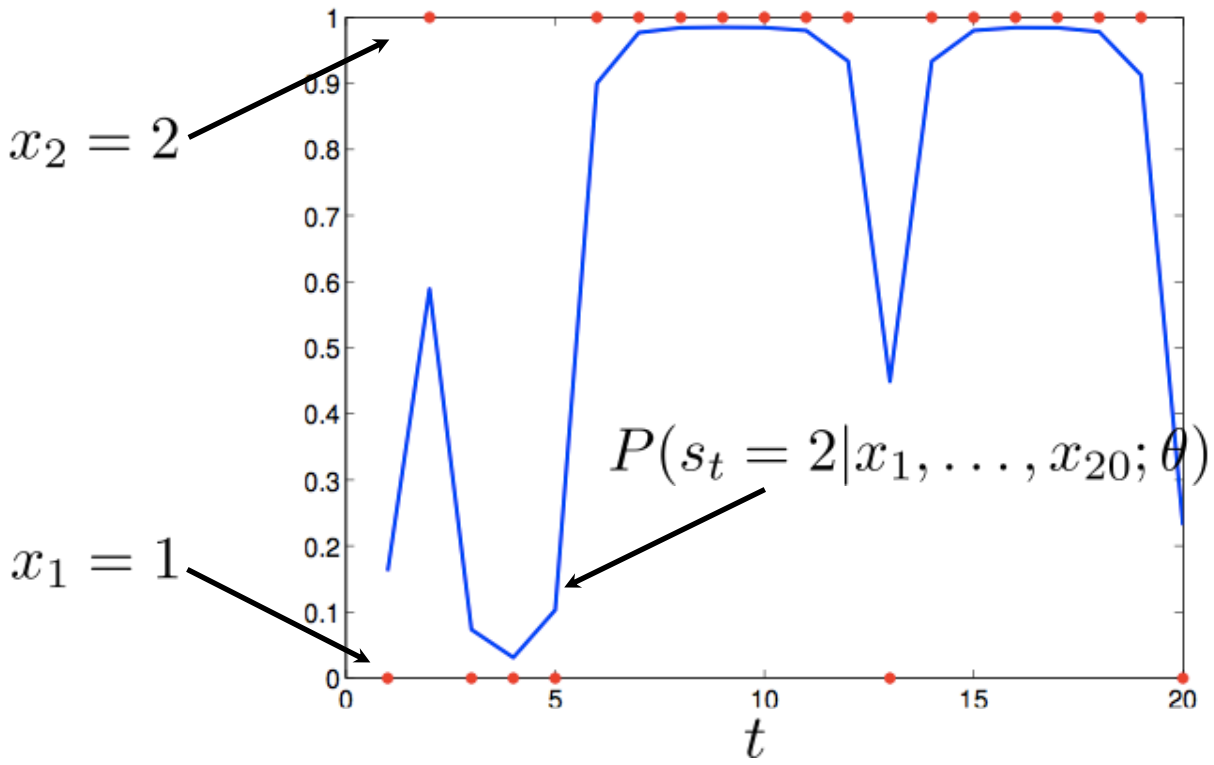
1) How to evaluate the probability of any observation sequence $P(x_1, \ldots, x_N; \theta)$
2) How to estimate the HMM parameters $\{q(s_1)\}$, $\{q(s_i|s_{i-1})\}$ and $\{q_e(x_i|s_i)\}$ on the basis of n observed sequences of varying length
3) How to find the most likely hidden state sequence $\hat{s}_1, \ldots, \hat{s}_N$ corresponding to observations $x_1, \ldots, x_N$

HMM example

- Underlying HMM

$$q(s_1) : \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \quad q(s_t|s_{t-1}) : \begin{bmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{bmatrix}$$

$$q_e(x_t|s_t) : \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$



HMM problems: common techniques

1) Compute probability of an observation sequence: Baum-Welch, forward-backward, alpha-beta algorithms.
   - Iterative, two-stage message passing algorithms
2) Estimate HMM parameters: apply EM to iteratively maximize the likelihood of the training data given the parameters.
   - Relies on some quantities computed in 1)

3) Compute most likely hidden state sequence for an observation sequence: Viterbi Algorithm: uses dynamic programming.

Optimal substructure:

- Due to the Markov property: the most probable path (through the hidden states) for the rest of any sequence, only depends on the state in which it starts.

→ For each state, only need to keep track of most probable path that ends in that state (not all possible paths to that state).