Machine Learning (6.867)

David Sontag

Massachusetts Institute of Technology

Lecture 15, Nov. 2, 2017

Course announcements

Project milestone 4 (*produce initial results*) due Tuesday

Where are we in the course?

1. Until now, we focused on *supervised learning*
   - Regression, classification
   - Loss functions, regularization
   - Linear, SVMs, nearest neighbor, neural networks, decision trees

---

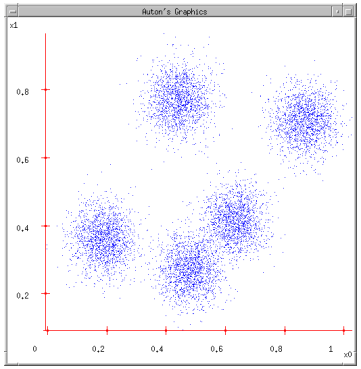
2. Today's lecture: *probabilistic modeling*
3. 11/7 – 11/14: using probabilistic models for *unsupervised learning*
4. 11/16: *approximate inference* in probabilistic models (Gibbs sampling)

---

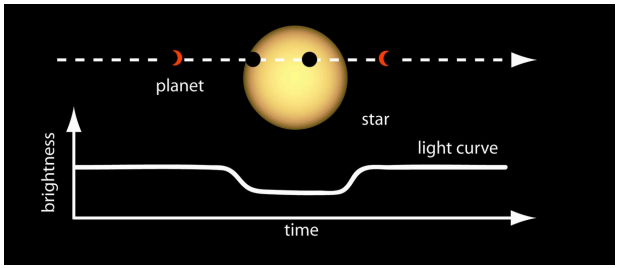
5. 11/21 and 12/5: *dimensionality reduction* with matrix factorization
6. 11/28: *recommendation systems* (not quite (un)supervised learning)
7. 12/7 and 12/12: *reinforcement learning*

How do we tackle problems like these?

- How many clusters are there in this data?



- Does there exist an exoplanet here?



Bernhard Schölkopf et al. PNAS 2016;113:7391-7398

- How do we learn to predict y in the presence of missing data?

x	y
0 1 ? ? 1 1 0 ?	1
1 ? 1 0 ? ? 0 0	0
1 0 0 0 0 0 1 0	?
1 ? ? ? ? ? ? 0	0

Probabilistic modeling

1. **Represent** the world as a collection of random variables
   - Observed variables $X_1, \ldots, X_n$
   - Latent variables $Z_1, \ldots, Z_m$ (*optional*)

   with joint distribution $p(X_1, \ldots, X_n, Z_1, \ldots, Z_m)$.

2. **Learn** the distribution from data

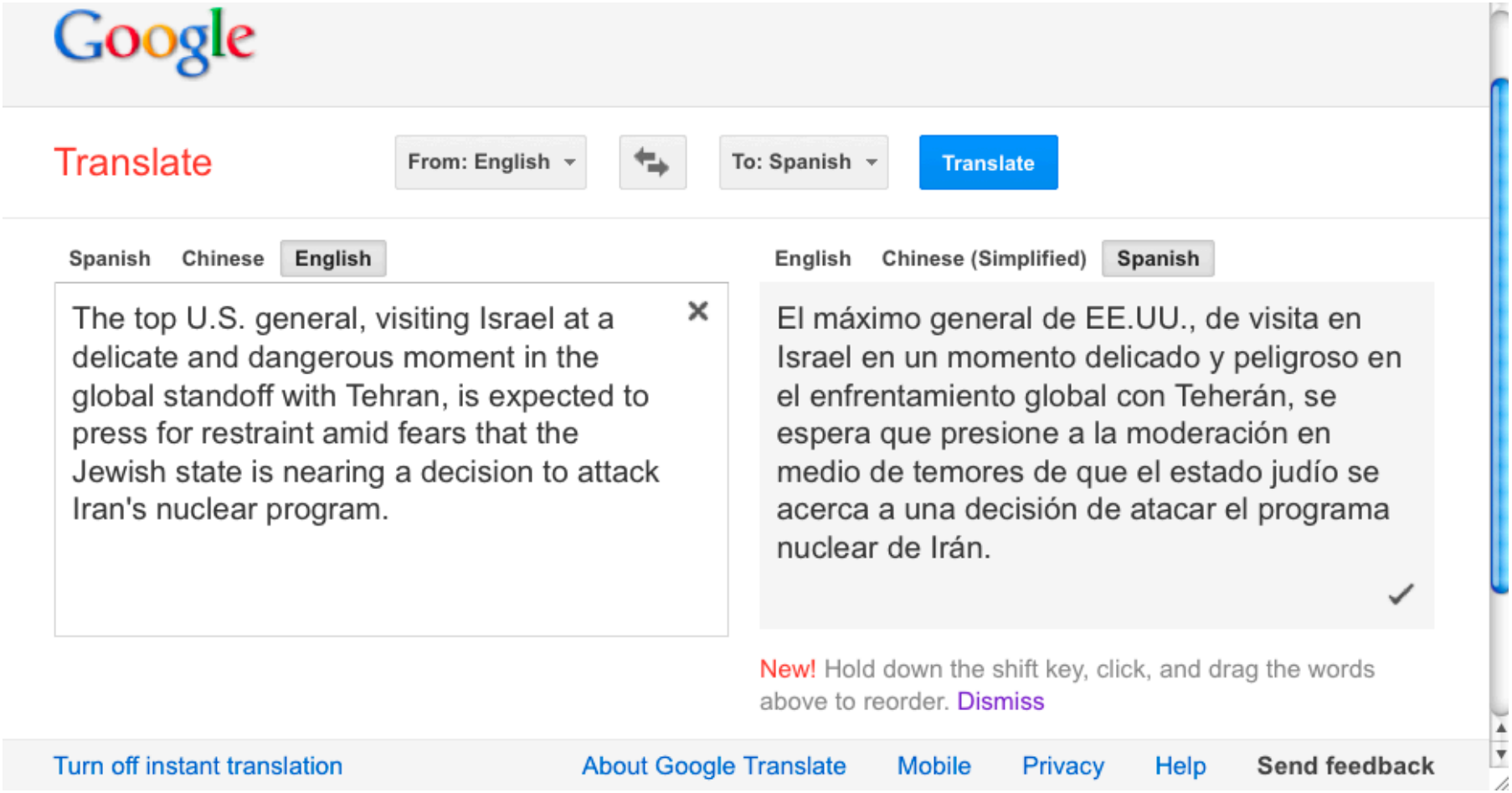
3. Either:
   - Obtain insight from the distribution itself (e.g. the learned parameters)
   - Use learned model for **prediction** by performing posterior **inference**,

$$f(x) = p(Z, X_{\text{query}} \mid X_{\text{evidence}} = x)$$

   where $f(x)$ is a vector-valued function which returns the posterior distribution given observed evidence

Applications: Modeling sequential data (*today*)

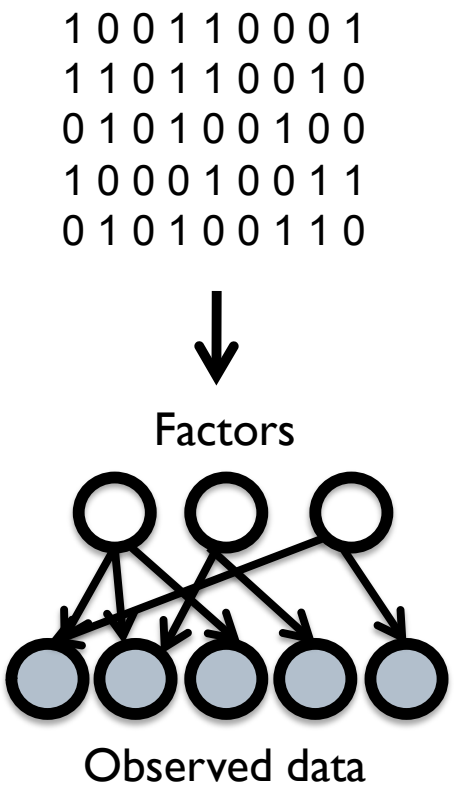
Machine translation, speech recognition, time series, ...



Recurrent neural networks, hidden Markov models, Gaussian processes

Applications: Modeling survey data (*next week*, 11/2)

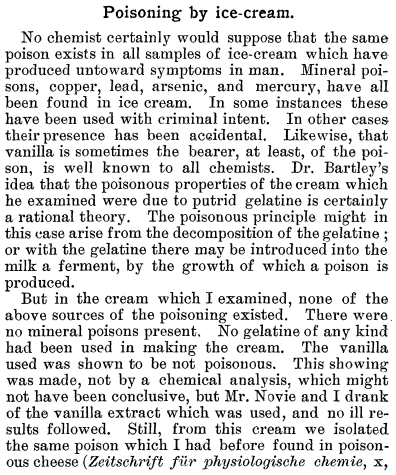
Factor analysis



- Social surveys with questions such as:
  - Should the tax rate be progressive?
  - Do you consider Edward Snowden a hero or a criminal?
  - What is your annual salary?
  - Should affirmative action be used in college admissions?
- Automatically *discover* the relevant hidden variables or factors, e.g.
  - Socioeconomic status
  - Health
  - Political attitudes
  - Family values
- *Infer* the underlying beliefs of each respondent

Applications: Modeling text data (*next week*, 11/9)

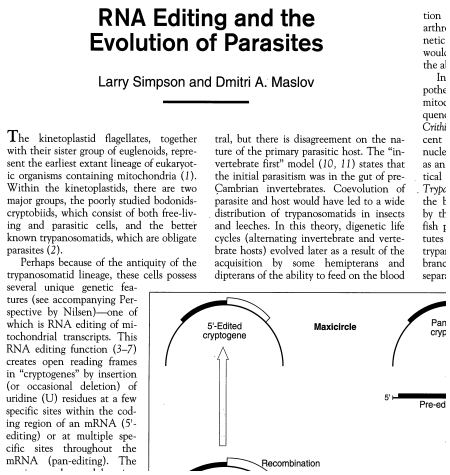
Topic models



Poisoning by ice-cream.

No chemist certainly would suppose that the same poison exists in all samples of ice-cream which have produced untoward symptoms in man. Mineral poisons, copper, lead, arsenic, and mercury, have all been found in ice cream. In some instances these have been used with criminal intent. In other cases their presence has been accidental. Likewise, that vanilla is sometimes the bearer, at least, of the poison, is well known to all chemists. Dr. Bartley's idea that the poisonous properties of the cream which he examined were due to putrid gelatine is certainly a rational theory. The poisonous principle might in this case arise from the decomposition of the gelatine; or with the gelatine there may be introduced into the milk a ferment, by the growth of which a poison is produced.

But in the cream which I examined, none of the above sources of the poisoning existed. There were no mineral poisons present. No gelatine of any kind had been used in making the cream. The vanilla used was shown to be not poisonous. This showing was made, not by a chemical analysis, which might not have been conclusive, but Mr. Novie and I drank of the vanilla extract which was used, and no ill results followed. Still, from this cream we isolated the same poison which I had before found in poisonous cheese (*Zeitschrift für physiologische chemie*, x,



RNA Editing and the Evolution of Parasites

Larry Simpson and Dmitri A. Maslov

The kinetoplastid flagellates, together with their sister group of euglenoids, represent the earliest extant lineage of eukaryotic organisms containing mitochondria (*1*). Within the kinetoplastids, there are two major groups, the poorly studied bodonids-cryptobiids, which consist of both free-living and parasitic cells, and the better known trypanosomatids, which are obligate parasites (*2*).

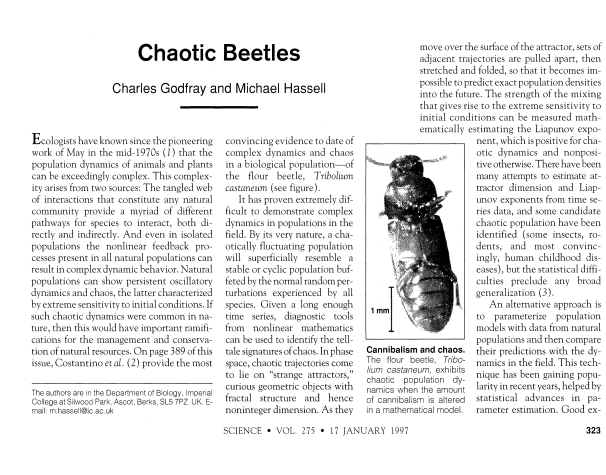
Perhaps because of the antiquity of the trypanosomatid lineage, these cells possess several unique genetic features (see accompanying Perspective by Nilsen)—one of which is RNA editing of mitochondrial transcripts. This RNA editing function (*3–7*) creates open reading frames in "cryptogenes" by insertion (or occasional deletion) of uridine (U) residues at a few specific sites within the coding region of an mRNA (5′-editing) or at multiple specific sites throughout the mRNA (pan-editing). The

tral, but there is disagreement on the nature of the primary parasitic host. The "invertebrate first" model (*10, 11*) states that the initial parasitism was in the gut of pre-Cambrian invertebrates. Coevolution of parasite and host would have led to a wide distribution of trypanosomatids in insects and leeches. In this theory, digenetic life cycles (alternating invertebrate and vertebrate hosts) evolved later as a result of the acquisition by some hemipterans and dipterans of the ability to feed on the blood

5′-Edited cryptogene

Maxicircle

Recombination



Chaotic Beetles

Charles Godfray and Michael Hassell

Ecologists have known since the pioneering work of May in the mid-1970s (*1*) that the population dynamics of animals and plants can be exceedingly complex. This complexity arises from two sources: The tangled web of interactions that constitute any natural community provide a myriad of different pathways for species to interact, both directly and indirectly. And even in isolated populations the nonlinear feedback processes present in all natural populations can result in complex dynamic behavior. Natural populations can show persistent oscillatory dynamics and chaos, the latter characterized by extreme sensitivity to initial conditions. If such chaotic dynamics were common in nature, then this would have important ramifications for the management and conservation of natural resources. On page 389 of this issue, Costantino *et al.* (*2*) provide the most convincing evidence to date of complex dynamics and chaos in a biological population—of the flour beetle, *Tribolium castaneum* (see figure).

It has proven extremely difficult to demonstrate complex dynamics in populations in the field. By its very nature, a chaotically fluctuating population will superficially resemble a stable or cyclic population buffeted by the normal random perturbations experienced by all species. Given a long enough time series, diagnostic tools from nonlinear mathematics can be used to identify the telltale signatures of chaos. In phase space, chaotic trajectories come to lie on "strange attractors," curious geometric objects with fractal structure and hence noninteger dimension. As they move over the surface of the attractor, sets of adjacent trajectories are pulled apart, then stretched and folded, so that it becomes impossible to predict exact population densities into the future. The strength of the mixing that gives rise to the extreme sensitivity to initial conditions can be measured mathematically estimating the Liapunov exponent, which is positive for chaotic dynamics and nonpositive otherwise. There have been many attempts to estimate attractor dimension and Liapunov exponents from time series data, and some candidate chaotic population have been identified (some insects, rodents, and most convincingly, human childhood diseases), but the statistical difficulties preclude any broad generalization (*3*).

An alternative approach is to parameterize population models with data from natural populations and then compare their predictions with the dynamics in the field. This technique has been gaining popularity in recent years, helped by statistical advances in parameter estimation. Good ex-

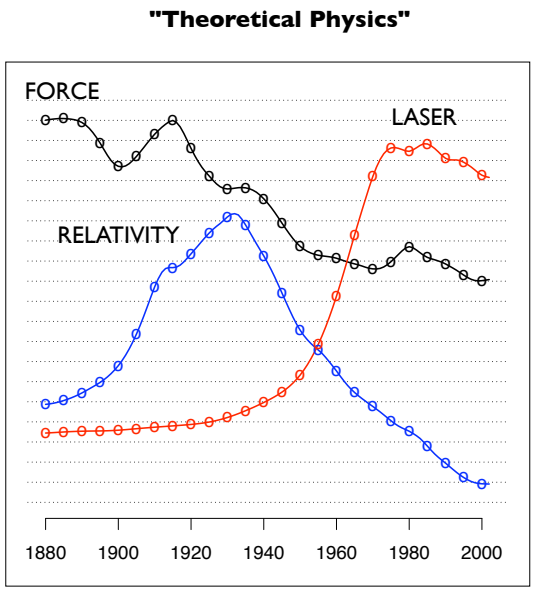
1 mm

Cannibalism and chaos. The flour beetle, *Tribolium castaneum*, exhibits chaotic population dynamics when the amount of cannibalism is altered in a mathematical model.

The authors are in the Department of Biology, Imperial College at Silwood Park, Ascot, Berks, SL5 7PZ UK. E-mail: m.hassel@ic.ac.uk

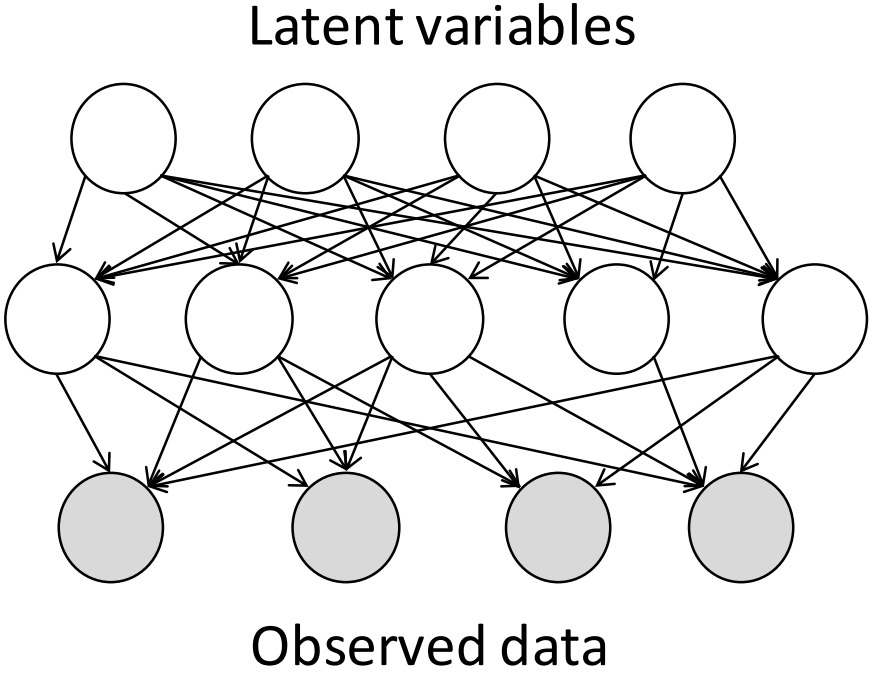
SCIENCE • VOL. 275 • 17 JANUARY 1997 323

human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

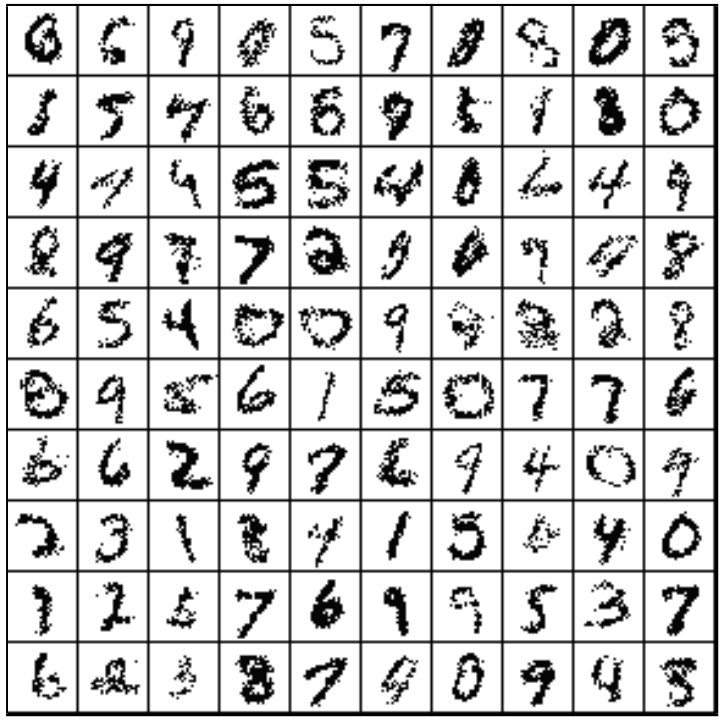


Applications: Modeling images and high-dimensional data (*in two weeks*, 11/14)

Deep generative models and variational autoencoders, adversarial nets



Samples from the model:



Key challenges

1. **Represent** the world as a collection of random variables with joint distribution $p(X_1, \ldots, X_n, Z_1, \ldots, Z_m)$
   - How does one *compactly describe* this joint distribution?
   - Directed graphical models (Bayesian networks)
   - Undirected graphical models (Markov random fields, factor graphs) *not covered in this class – see 6.438*

2. **Learn** the distribution from data
   - Maximum likelihood estimation. Other estimation methods?
   - How much data do we need?
   - How much computation does it take?

3. Perform **inference** to compute posterior distribution $p(Z \mid X_1 = x_1, \ldots, X_n = x_n)$
   - Exactly doing this is typically computationally intractable
   - What are computationally efficient *approximate* inference algorithms?

Example: Medical diagnosis

- Variable for each **symptom** (e.g. "fever", "cough", "fast breathing", "shaking", "nausea", "vomiting")
- Variable for each **disease** (e.g. "pneumonia", "flu", "common cold", "bronchitis", "tuberculosis")
- Diagnosis is performed by **inference** in the model:

$$p(\text{pneumonia} = 1 \mid \text{cough} = 1, \text{fever} = 1, \text{vomiting} = 0)$$

- One famous model, Quick Medical Reference (QMR-DT), has 600 diseases and 4000 findings

Representing the distribution

- Naively, could represent multivariate distributions with table of probabilities for each outcome (assignment)
- How many outcomes are there in QMR-DT? 2^{4600}
- **Estimation** of joint distribution would require a huge amount of data
- **Inference** of conditional probabilities, e.g.

$$p(\text{pneumonia} = 1 \mid \text{cough} = 1, \text{fever} = 1, \text{vomiting} = 0)$$

  would require summing over exponentially many variables' values
- Moreover, defeats the purpose of probabilistic modeling, which is to make predictions with *previously unseen observations*

Structure through independence

- If $X_1, \ldots, X_n$ are independent, then

$$p(x_1, \ldots, x_n) = p(x_1)p(x_2)\cdots p(x_n)$$

- 2^n entries can be described by just n numbers (if $|\text{Val}(X_i)| = 2$)!
- However, this is not a very *useful* model – observing a variable X_i cannot influence our predictions of X_j
- If $X_1, \ldots, X_n$ are *conditionally independent* given Y, denoted as $X_i \perp \mathbf{X}_{-i} \mid Y$, then

$$\begin{aligned} p(y, x_1, \ldots, x_n) &= p(y)p(x_1 \mid y) \prod_{i=2}^{n} p(x_i \mid x_1, \ldots, x_{i-1}, y) \\ &= p(y)p(x_1 \mid y) \prod_{i=2}^{n} p(x_i \mid y). \end{aligned}$$

- This is a simple, yet *powerful*, model

Example: naive Bayes for classification

- Classify e-mails as spam ($Y = 1$) or not spam ($Y = 0$)
  - Let $1 : n$ index the words in our vocabulary (e.g., English)
  - $X_i = 1$ if word i appears in an e-mail, and 0 otherwise
  - E-mails are drawn according to some distribution $p(Y, X_1, \ldots, X_n)$
- Suppose that the words are conditionally independent given Y. Then,

$$p(y, x_1, \ldots x_n) = p(y) \prod_{i=1}^{n} p(x_i \mid y)$$

Estimate the model with maximum likelihood. **Predict** with:

$$p(Y = 1 \mid x_1, \ldots x_n) = \frac{p(Y = 1) \prod_{i=1}^{n} p(x_i \mid Y = 1)}{\sum_{y=\{0,1\}} p(Y = y) \prod_{i=1}^{n} p(x_i \mid Y = y)}$$

- Are the independence assumptions made here reasonable?
- Philosophy: Nearly all probabilistic models are "wrong", but many are nonetheless useful

Bayesian networks

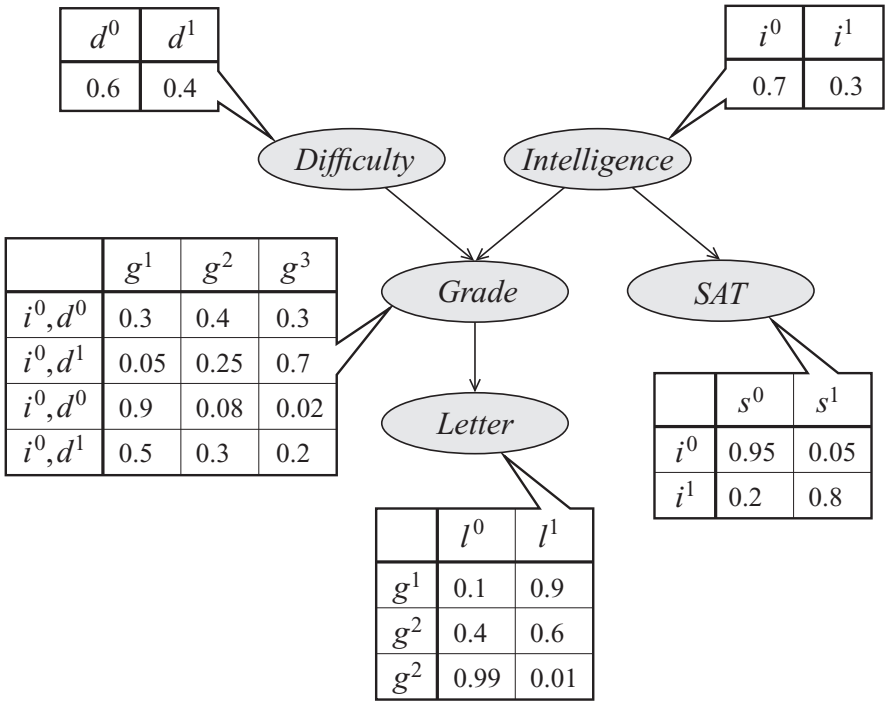
- A **Bayesian network** is specified by a directed *acyclic* graph $G = (V, E)$ with:
  1. One node $i \in V$ for each random variable X_i
  2. One conditional probability distribution (CPD) per node, $p(x_i \mid \mathbf{x}_{\mathrm{Pa}(i)})$, specifying the variable's probability conditioned on its parents' values
- Corresponds 1-1 with a particular factorization of the joint distribution:

$$p(x_1, \ldots x_n) = \prod_{i \in V} p(x_i \mid \mathbf{x}_{\mathrm{Pa}(i)})$$

- Powerful framework for designing *algorithms* to perform probability computations
- Enables use of *prior knowledge* to specify (part of) model structure

Example

- Consider the following Bayesian network:



- What is its joint distribution?

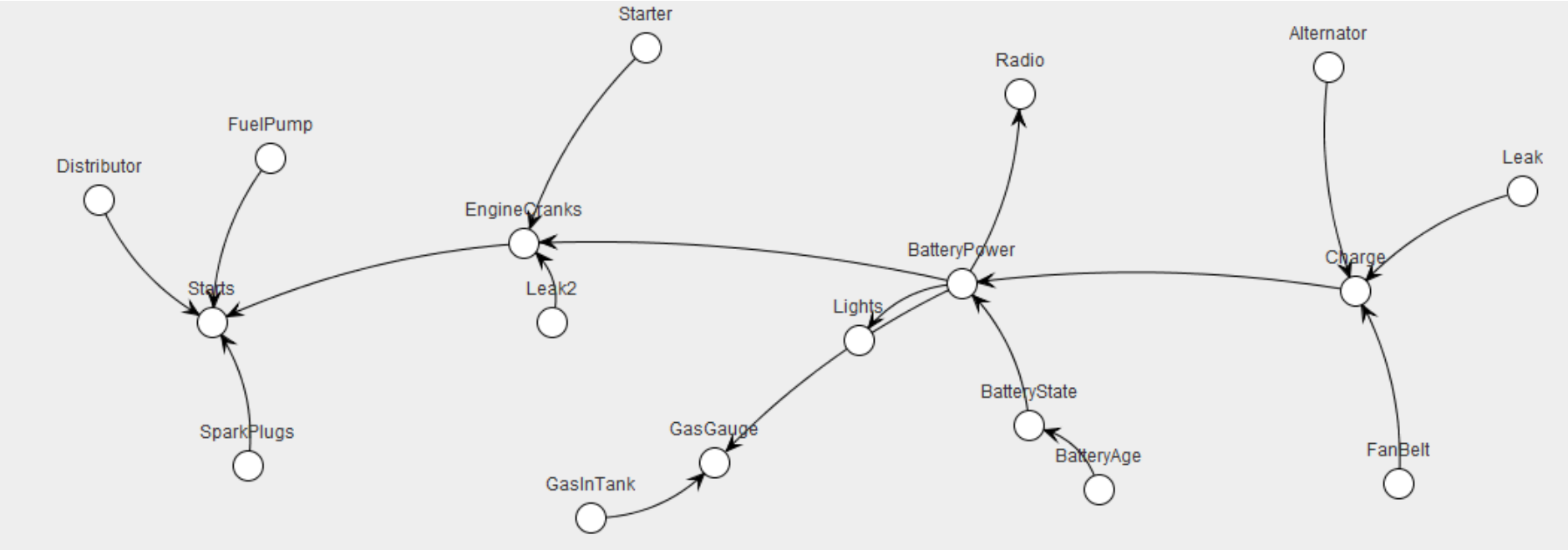
$$p(x_1, \ldots x_n) = \prod_{i \in V} p(x_i \mid \mathbf{x}_{\mathrm{Pa}(i)})$$

$$p(d, i, g, s, l) = p(d)p(i)p(g \mid i, d)p(s \mid i)p(l \mid g)$$

More examples

$$p(x_1, \ldots x_n) = \prod_{i \in V} p(x_i \mid \mathbf{x}_{\mathrm{Pa}(i)})$$

Will my car start this morning?



Heckerman *et al.*, Decision-Theoretic Troubleshooting, 1995

More examples

$$p(x_1, \ldots x_n) = \prod_{i \in V} p(x_i \mid \mathbf{x}_{\mathrm{Pa}(i)})$$

What is the differential diagnosis?

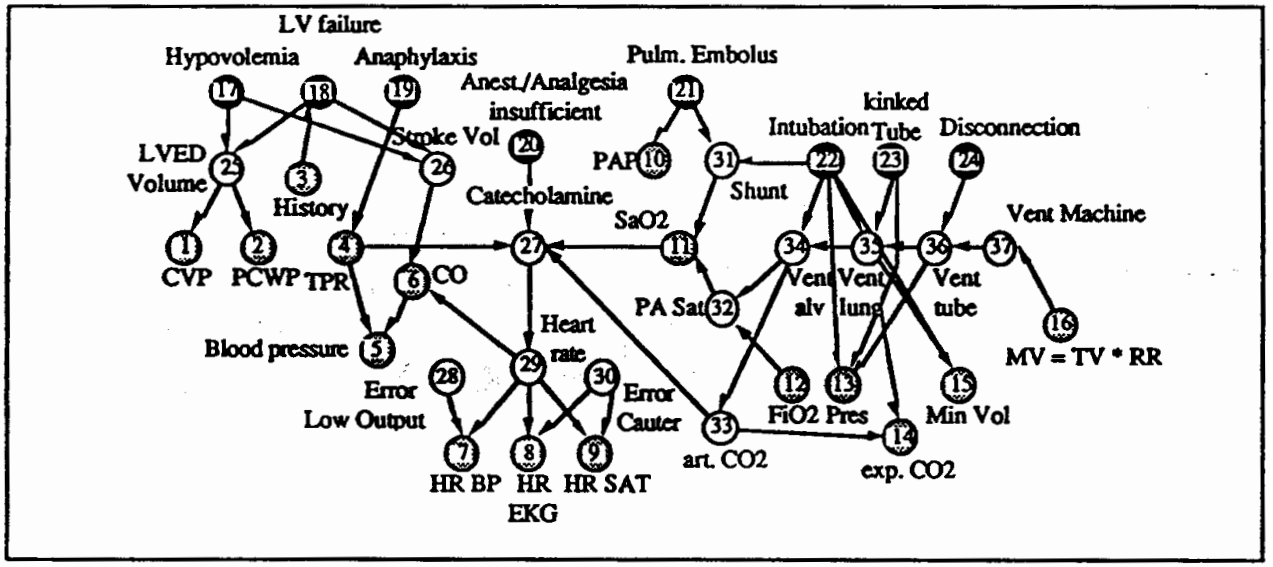
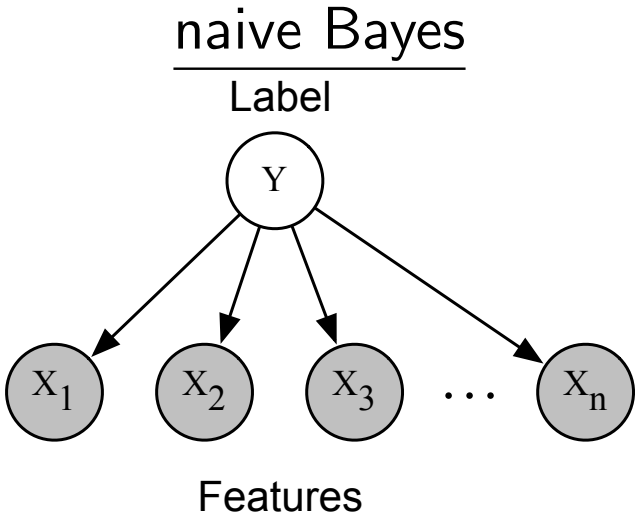


Fig. 1 The ALARM network representing causal relationships is shown with diagnostic (●), intermediate (○) and measurement (⊗) nodes. CO: cardiac output, CVP: central venous pressure, LVED volume: left ventricular end-diastolic volume, LV failure: left ventricular failure, MV: minute ventilation, PA Sat: pulmonary artery oxygen saturation, PAP: pulmonary artery pressure, PCWP: pulmonary capillary wedge pressure, Pres: breathing pressure, RR: respiratory rate, TPR: total peripheral resistance, TV: tidal volume

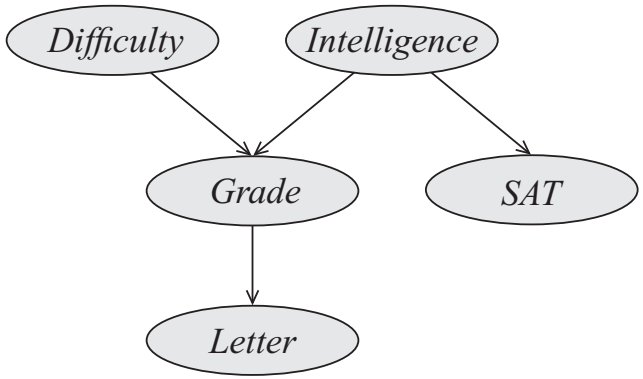
Beinlich *et al.*, The ALARM Monitoring System, 1989

Bayesian networks are *generative models*



- Evidence is denoted by shading in a node
- Can interpret Bayesian network as a **generative process**. For example, to *generate* an e-mail, we
  1. Decide whether it is spam or not spam, by samping $y \sim p(Y)$
  2. For each word $i = 1$ to n, sample $x_i \sim p(X_i \mid Y = y)$

Bayesian network structure implies conditional independencies!



- The joint distribution corresponding to the above BN factors as

$$p(d, i, g, s, l) = p(d)p(i)p(g \mid i, d)p(s \mid i)p(l \mid g)$$

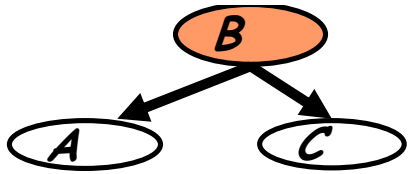
- However, by the chain rule, *any* distribution can be written as

$$p(d, i, g, s, l) = p(d)p(i \mid d)p(g \mid i, d)p(s \mid i, d, g)p(l \mid g, d, i, g, s)$$

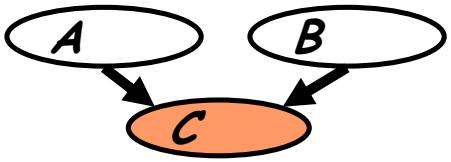
- Thus, we are assuming the following additional independencies:
  $D \perp I$, $S \perp \{D, G\} \mid I$, $L \perp \{I, D, S\} \mid G$. What else?

Bayesian network structure implies conditional independencies!

- Generalizing the above arguments, we obtain that a variable is independent from its non-descendants given its parents

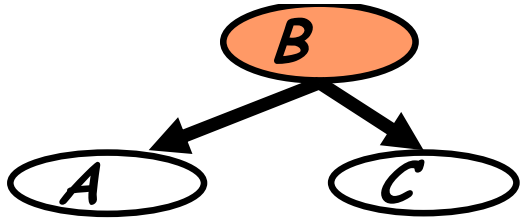


- **Common parent** – fixing B *decouples* A and C
- **Cascade** – knowing B *decouples* A and C



- **V-structure** – Knowing C *couples* A and B
  - This important phenomona is called **explaining away** and is what makes Bayesian networks so powerful

A simple justification (for common parent)



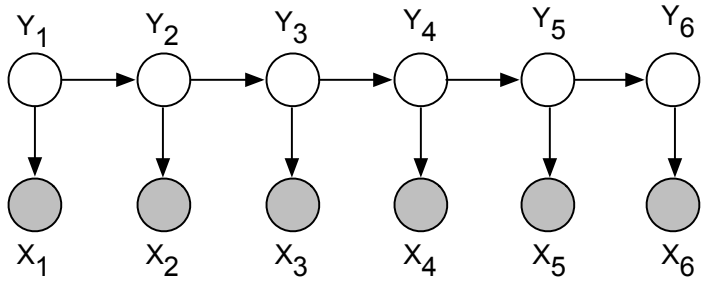
We'll show that $p(A, C \mid B) = p(A \mid B)p(C \mid B)$ for *any* distribution $p(A, B, C)$ that factors according to this graph structure, i.e.

$$p(A, B, C) = p(B)p(A \mid B)p(C \mid B)$$

Proof.

$$p(A, C \mid B) = \frac{p(A, B, C)}{p(B)} = p(A \mid B)p(C \mid B)$$

Hidden Markov models

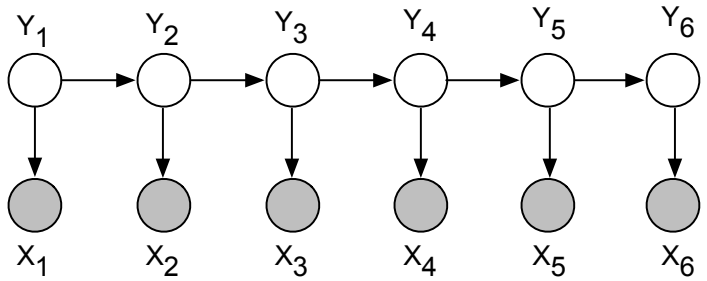


- Widely used in control theory and signal processing. Later applications included speech recognition and computational biology
- Joint distribution factors as:

$$p(\mathbf{y}, \mathbf{x}) = p(y_1)p(x_1 \mid y_1) \prod_{t=2}^{T} p(y_t \mid y_{t-1})p(x_t \mid y_t)$$

  - $p(y_1)$ is the distribution for the starting state
  - $p(y_t \mid y_{t-1})$ is the *transition* probability between any two states
  - $p(x_t \mid y_t)$ is the *emission* probability
- What are the conditional independencies here? For example, $Y_1 \perp \{Y_3, \ldots, Y_6\} \mid Y_2$

Hidden Markov models



- Joint distribution factors as:

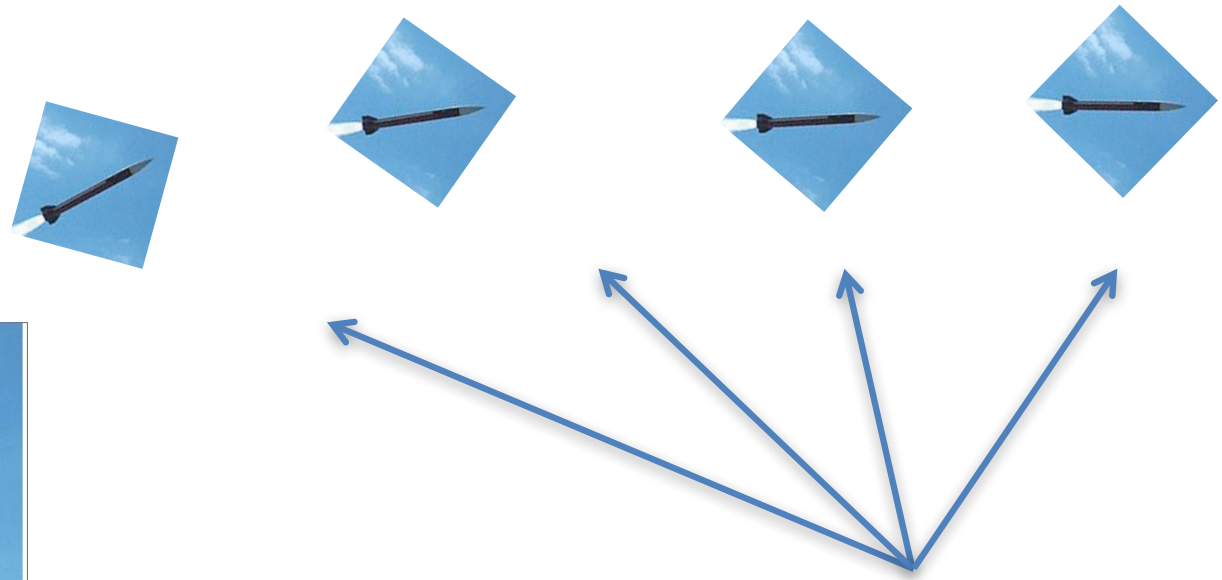
$$p(\mathbf{y}, \mathbf{x}) = p(y_1)p(x_1 \mid y_1) \prod_{t=2}^{T} p(y_t \mid y_{t-1})p(x_t \mid y_t)$$

- A **homogeneous** HMM uses the same parameters (β and α below) for each transition and emission distribution (**parameter sharing**):

$$p(\mathbf{y}, \mathbf{x}) = p(y_1)\alpha_{x_1,y_1} \prod_{t=2}^{T} \beta_{y_t,y_{t-1}} \alpha_{x_t,y_t}$$

How many parameters need to be learned?

Example application of HMMs: Tracking



Radar

Observe noisy measurements of missile location: X_1, X_2, …

Where is the missile **now**? Where will it be in 10 seconds?

Inference

- To find out where the missile is *now*, we do **marginal inference**:

$$p(y_T \mid x_1, \ldots, x_T)$$

- To find the most likely *trajectory*, we do **MAP (maximum a posteriori) inference**:

$$\arg\max_y p(y_1, \ldots, y_T \mid x_1, \ldots, x_T)$$

- Key challenge: efficiently computing these quantities
- Suppose Y_t takes k states. Naively, would seem to require enumerating or summing over k^{T-1} sequences:

$$\begin{aligned} p(y_T \mid x_1, \ldots, x_T) &\propto p(y_T, x_1, \ldots, x_T) \\ &= \sum_{y_1, \ldots, y_{T-1}} p(y_1, \ldots, y_T, x_1, \ldots, x_T) \end{aligned}$$

Marginal inference in HMMs

- Use **dynamic programming**

$$
\begin{aligned}
p(y_T, x_1, \quad \ldots \quad, x_T) &= \sum_{y_{T-1}} p(y_{T-1}, y_T, x_1, \ldots, x_T) \\
&= \sum_{y_{T-1}} p(y_{T-1}, x_1, \ldots, x_{T-1}) p(y_T, x_T \mid y_{T-1}, x_1, \ldots, x_{T-1}) \\
&= \sum_{y_{T-1}} p(y_{T-1}, x_1, \ldots, x_{T-1}) p(y_T, x_T \mid y_{T-1}) \\
&= \sum_{y_{T-1}} p(y_{T-1}, x_1, \ldots, x_{T-1}) p(y_T \mid y_{T-1}) p(x_T \mid y_{T-1}, y_T) \\
&= \sum_{y_{T-1}} p(y_{T-1}, x_1, \ldots, x_{T-1}) p(y_T \mid y_{T-1}) p(x_T \mid y_T)
\end{aligned}
$$

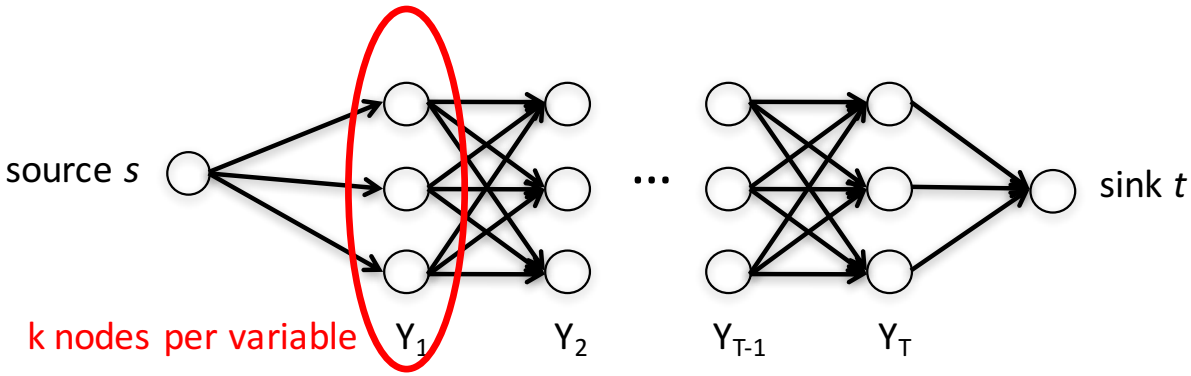
- For $T = 1$, initialize $p(y_1, x_1) = p(y_1) p(x_1 \mid y_1)$
- Total running time is $O(Tk^2)$ – linear time! Easy to do **filtering**

MAP inference in HMMs

- MAP inference in HMMs can *also* be solved in linear time!

$$\begin{aligned}\arg\max_y \quad & Pr(y_1, \ldots, y_T \mid x_1, \ldots, x_T) = \arg\max_y p(y_1, \ldots, y_T, x_1, \ldots, x_T) \\ = \quad & \arg\max_y \log p(y_1, \ldots, y_T, x_1, \ldots, x_T) \\ = \quad & \arg\max_y \log \left[p(y_1)p(x_1 \mid y_1)\right] + \sum_{t=2}^{T} \log \left[p(y_t \mid y_{t-1})p(x_t \mid y_t)\right]\end{aligned}$$

- Formulate as a shortest paths problem (called the *Viterbi algorithm*)



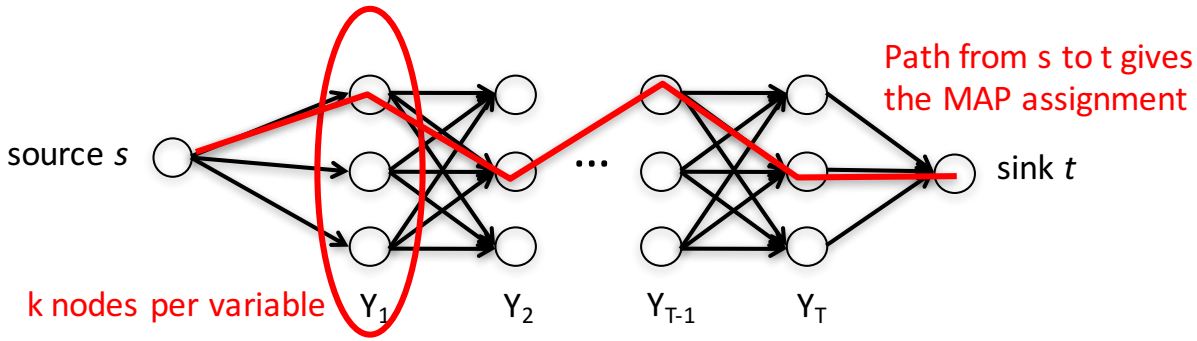
- What should the weight for edges (s, y_1) and (y_T, t) be?
- Weight should the weight for edges (y_{t-1}, y_t) be?

MAP inference in HMMs

- MAP inference in HMMs can *also* be solved in linear time!

$$\begin{aligned}\arg\max_y \quad & Pr(y_1, \ldots, y_T \mid x_1, \ldots, x_T) = \arg\max_y p(y_1, \ldots, y_T, x_1, \ldots, x_T) \\ = \quad & \arg\max_y \log p(y_1, \ldots, y_T, x_1, \ldots, x_T) \\ = \quad & \arg\max_y \log \left[p(y_1)p(x_1 \mid y_1)\right] + \sum_{t=2}^{T} \log \left[p(y_t \mid y_{t-1})p(x_t \mid y_t)\right]\end{aligned}$$

- Formulate as a shortest paths problem (called the *Viterbi algorithm*)



- Weight for edge (s, y_1) is $-\log \left[p(y_1)p(x_1 \mid y_1)\right]$. Weight for edge $(y_T, t) = 0$
- Weight for edge (y_{t-1}, y_t) is $-\log \left[p(y_t \mid y_{t-1})p(x_t \mid y_t)\right]$

Summary

- **Bayesian networks** given by (G, P) where P is specified as a set of local **conditional probability distributions** associated with G's nodes
- One interpretation of a BN is as a **generative model**, where variables are sampled in topological order
- Factorization implies conditional independencies
- Computing the probability of any assignment is obtained by multiplying CPDs
  - **Bayes' rule** is used to compute conditional probabilities
  - Marginalization or **inference** is often computationally difficult