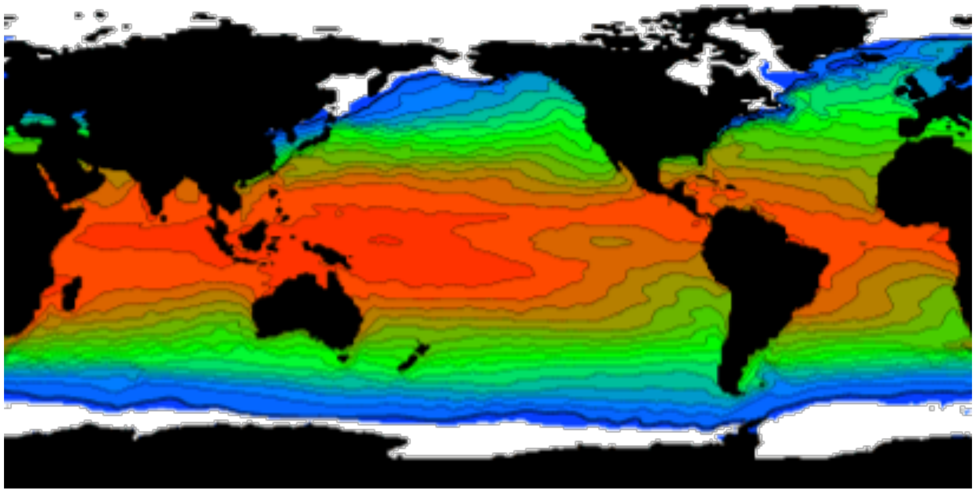
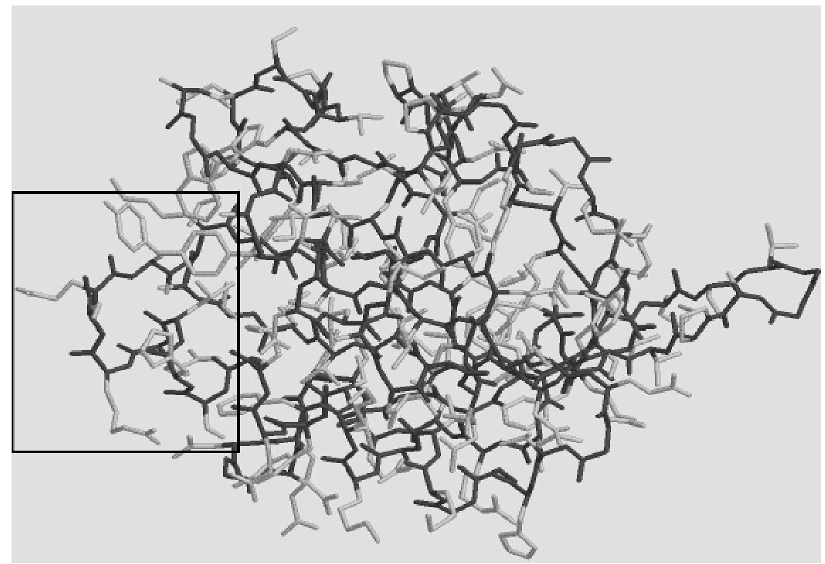
Probabilistic Graphical Models

Special Topics in Machine Learning
Brown University CSCI 2950-P, Spring 2013
Tuesdays & Thursdays, 1:00-2:20pm, CIT506

Instructor: *Erik Sudderth*
Teaching Assistant: *Jason Pacheco*

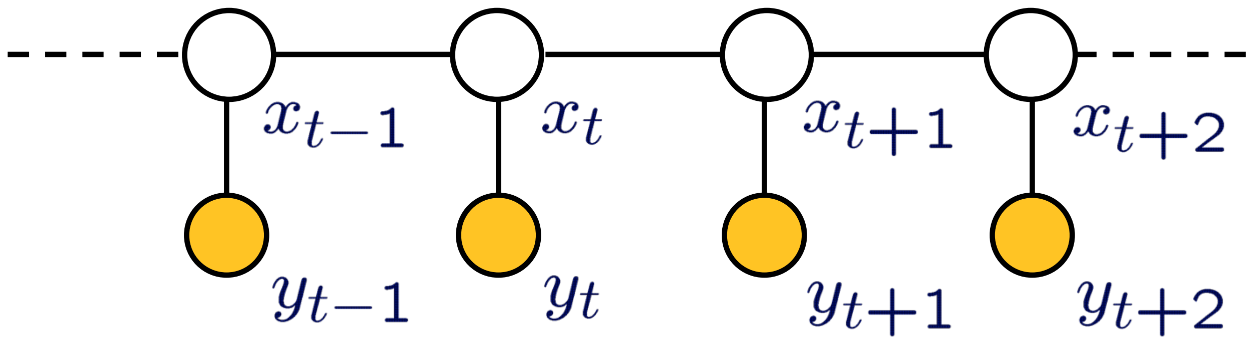
Learning from Structured Data





Hidden Markov Models (HMMs)

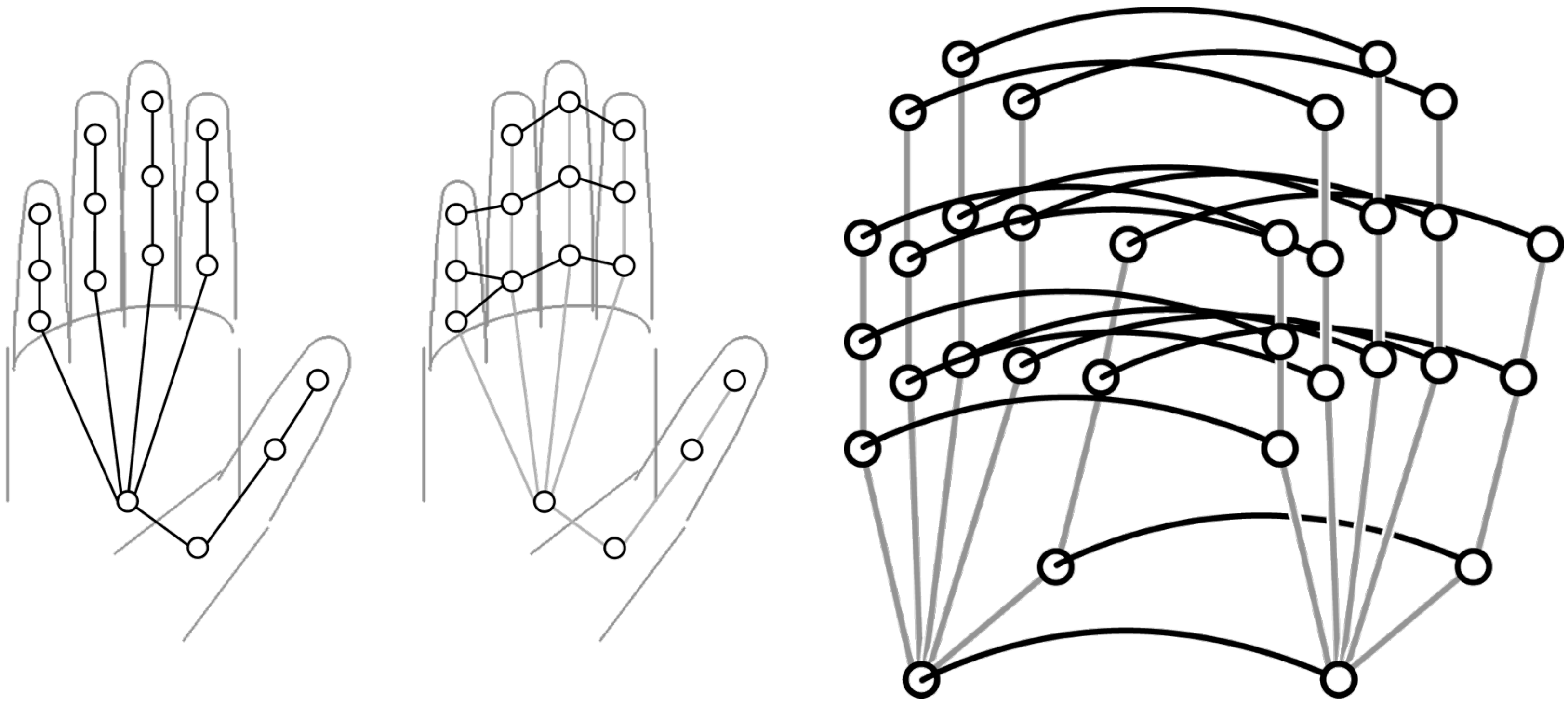
Visual Tracking



$$p(x, y) = p(x_0) \prod_{t=1}^{T} p(x_t \mid x_{t-1}) p(y_t \mid x_t)$$

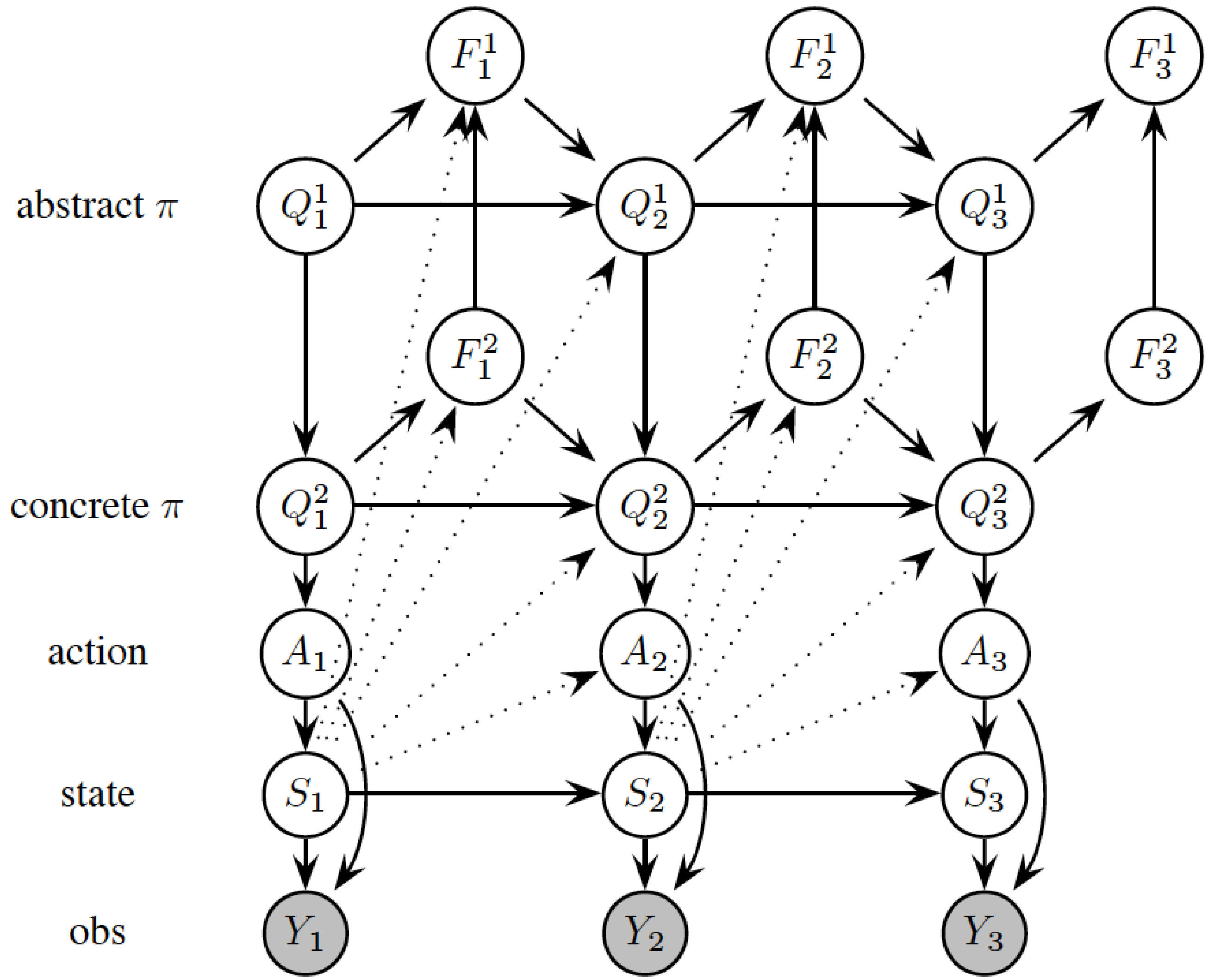
"Conditioned on the present, the past and future are statistically independent"

Kinematic Hand Tracking



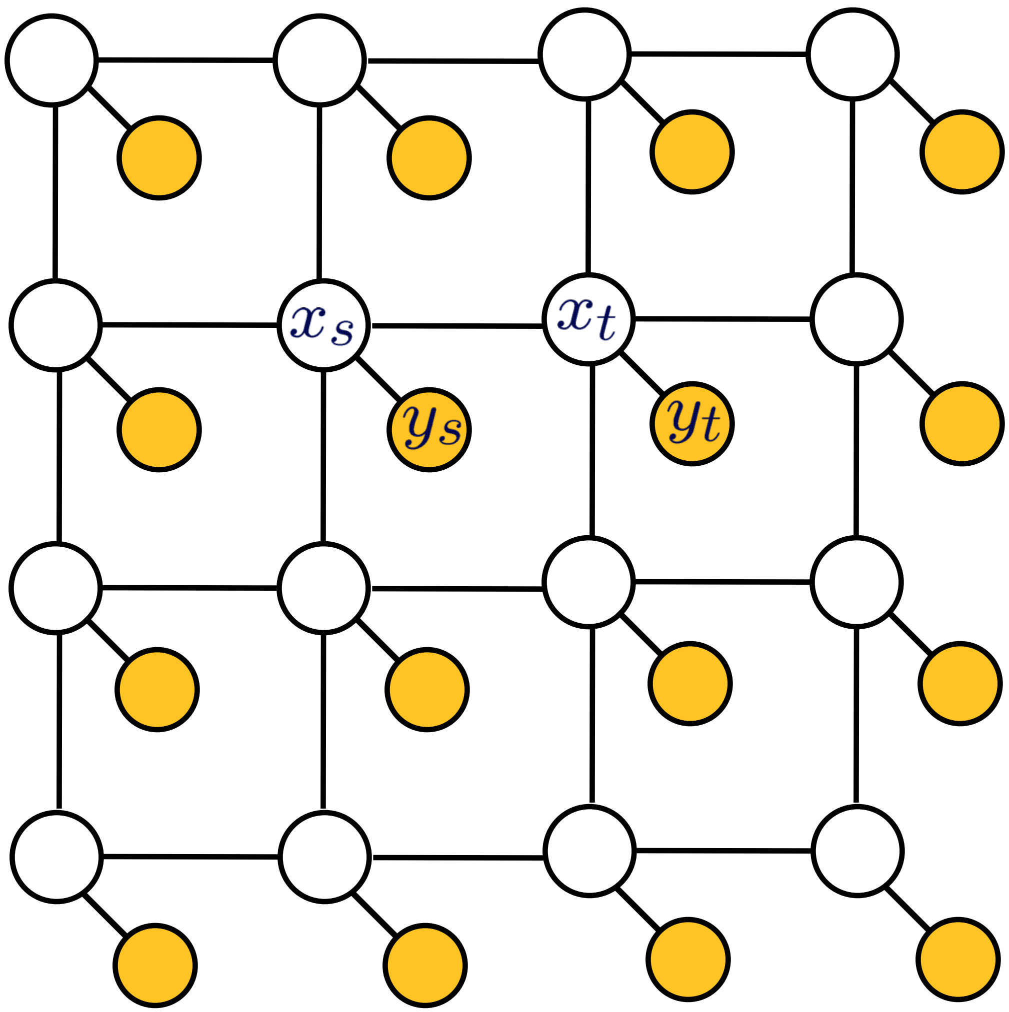
Kinematic Prior *Structural Prior* *Dynamic Prior*

Dynamic Bayesian Networks



Murphy,

Nearest-Neighbor Grids



Low Level Vision

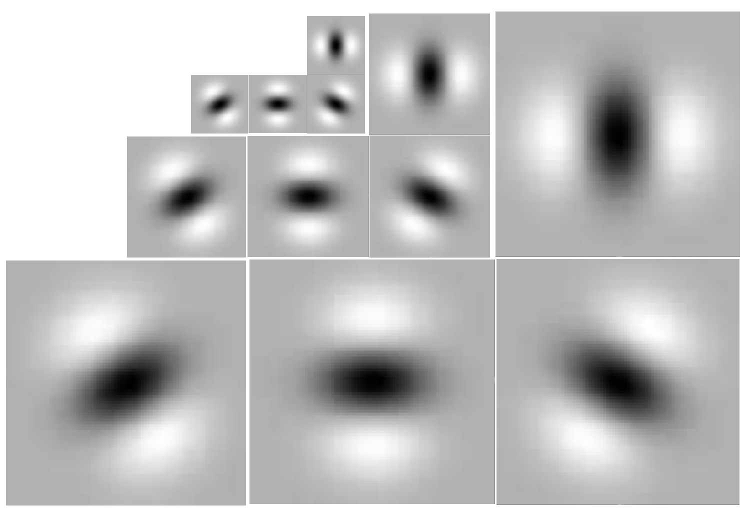
- Image denoising
- Stereo
- Optical flow
- Shape from shading
- Superresolution
- Segmentation

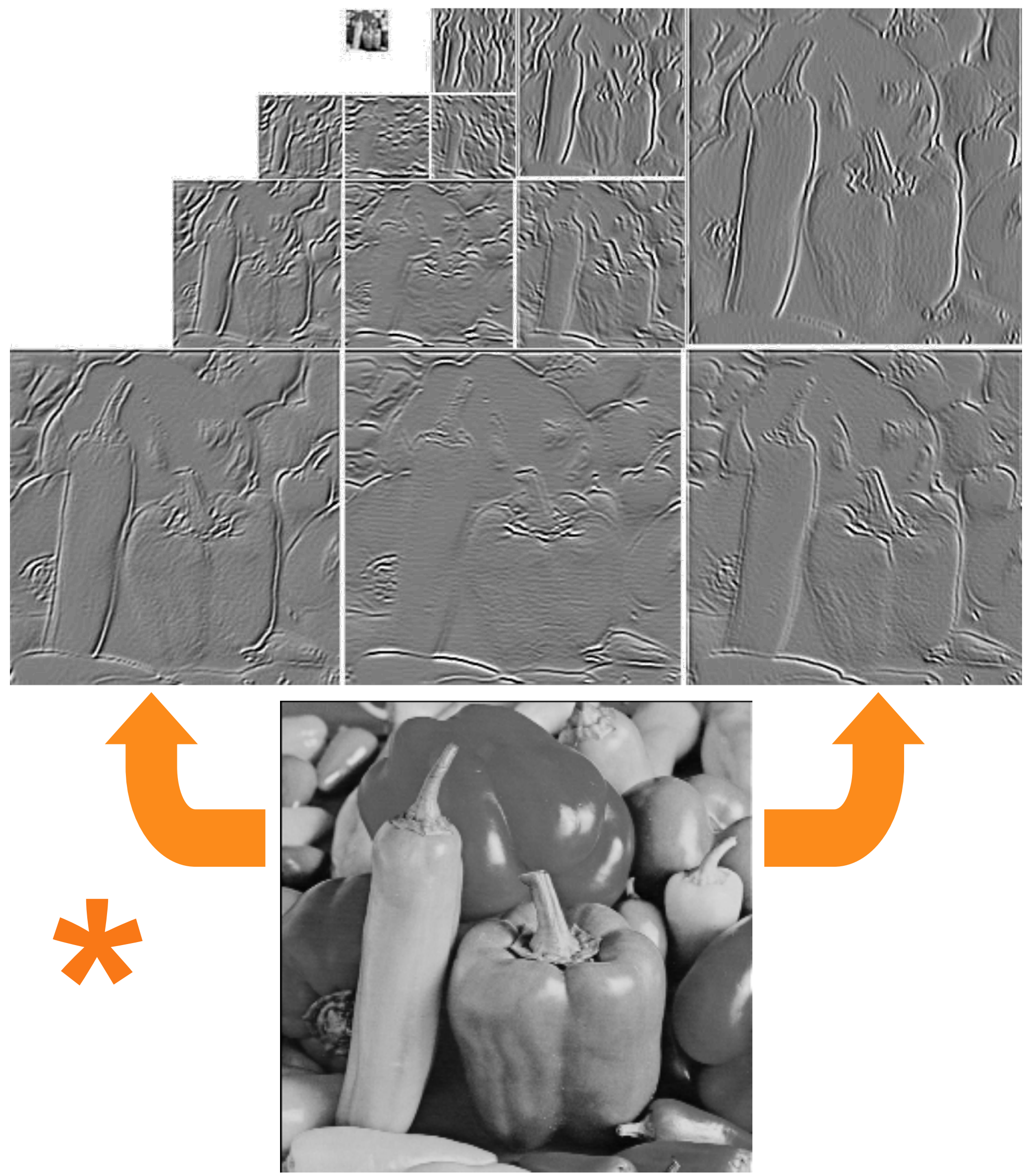
x_s → unobserved or hidden variable

y_s → local observation of x_s

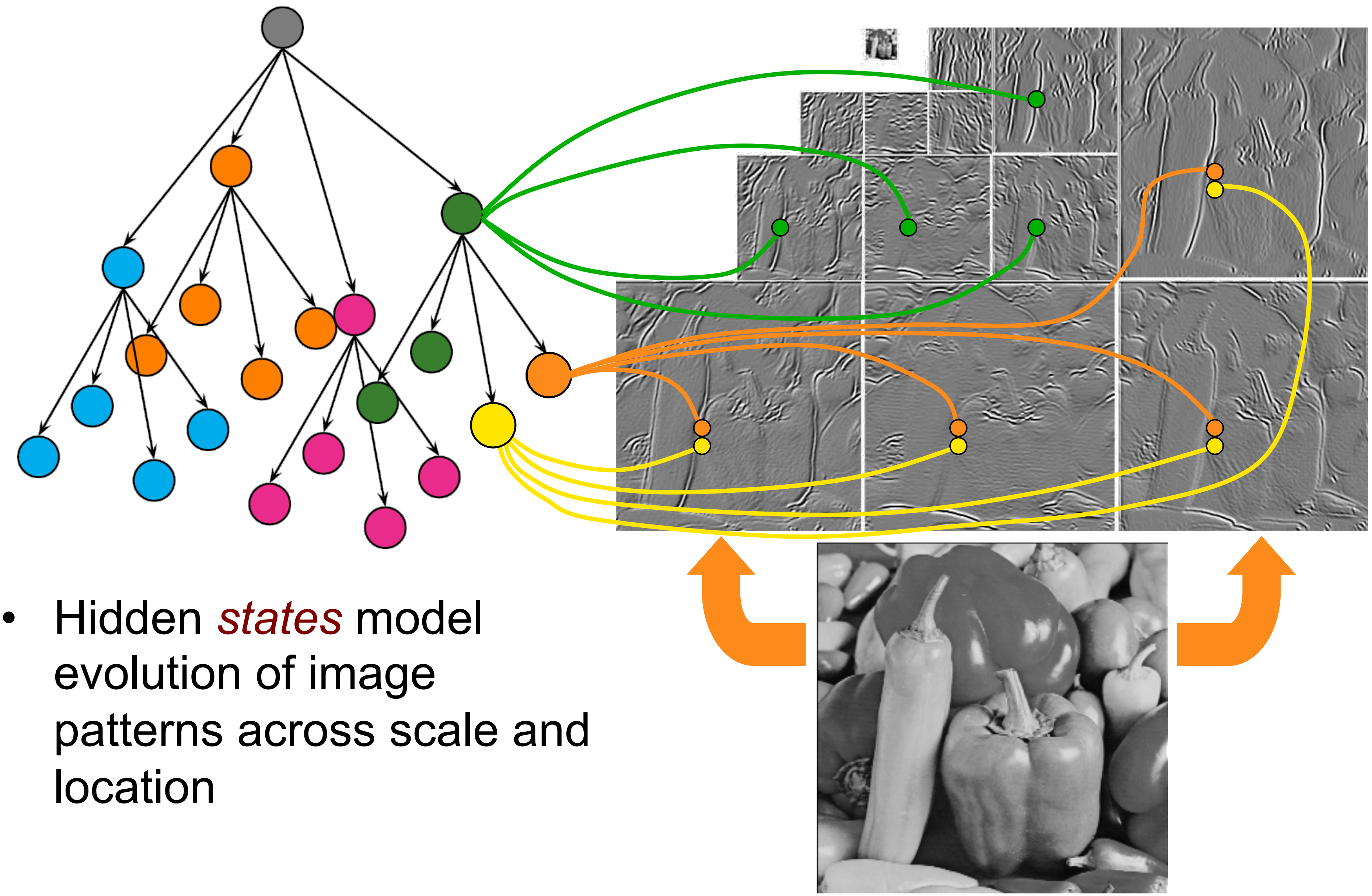
Wavelet Decompositions

- Bandpass decomposition of images into multiple *scales* & *orientations*
- Dense features which *simplify* statistics of natural images



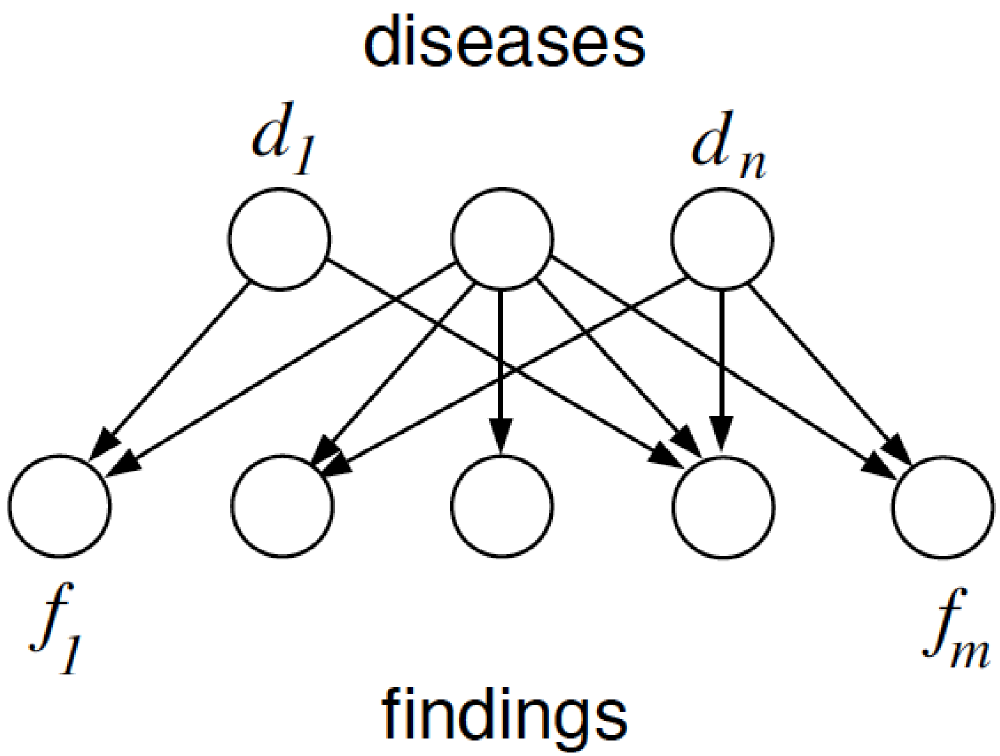


Hidden Markov Trees



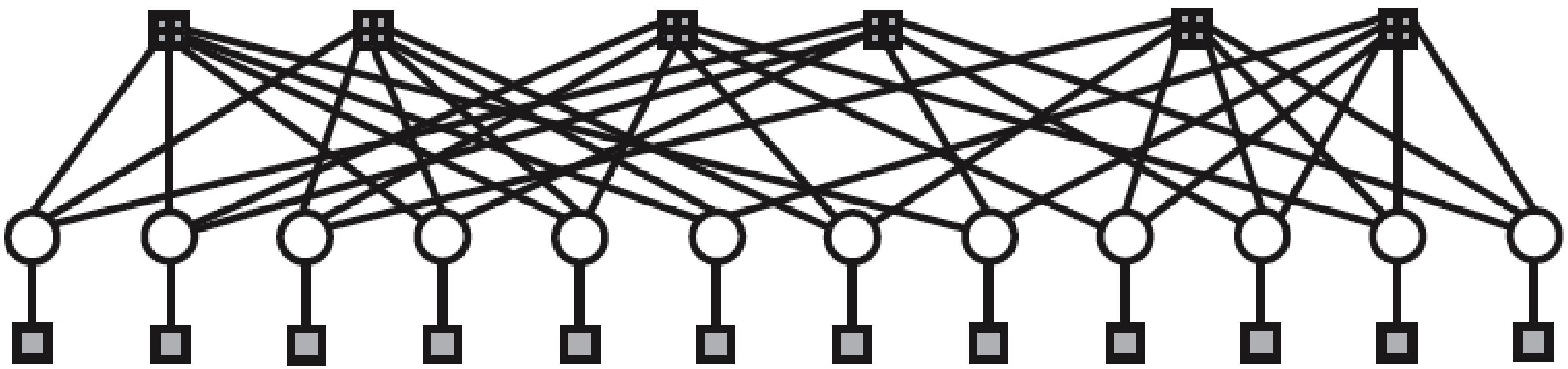
- Hidden *states* model evolution of image patterns across scale and location

Medical Diagnosis

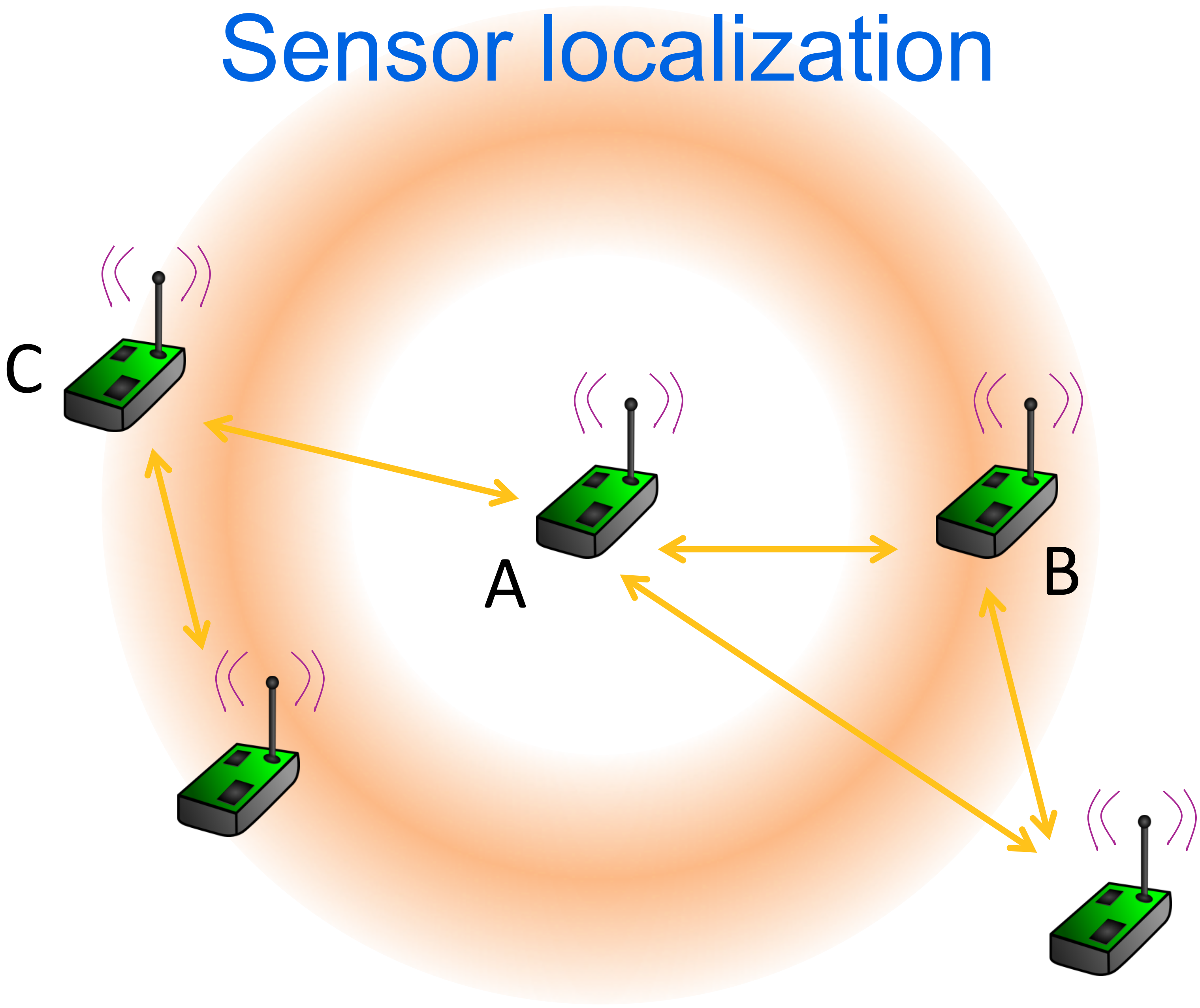


Parameterization: Noisy-OR, logistic regression, generalized linear models…

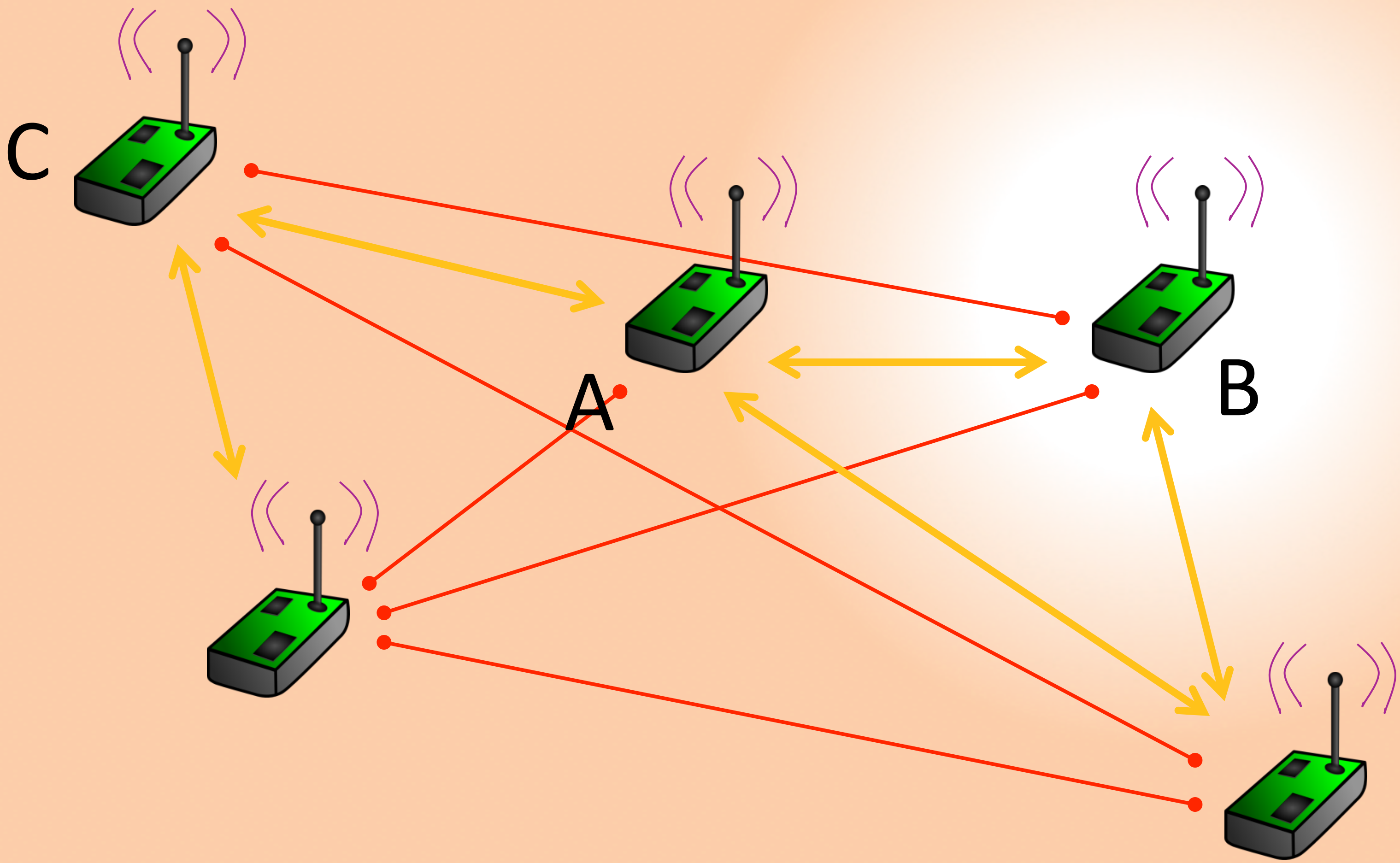
Low Density Parity Check (LDPC) Code



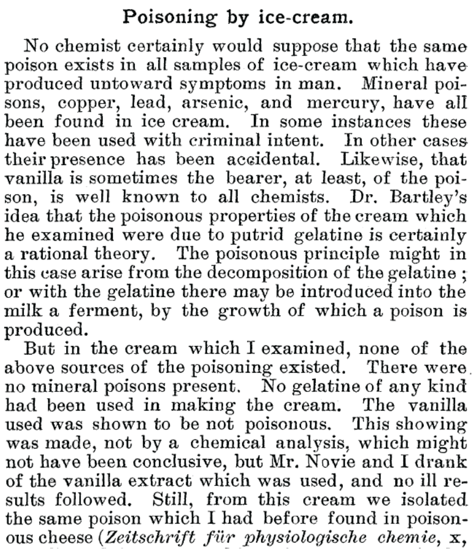
Sensor localization



Sensor localization



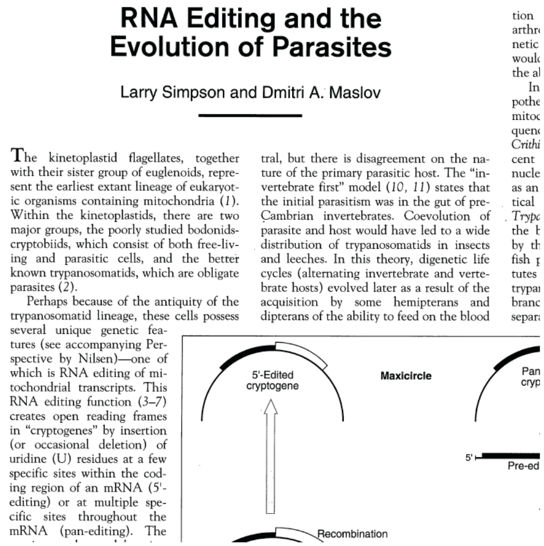
Example Data for a Topic Model



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

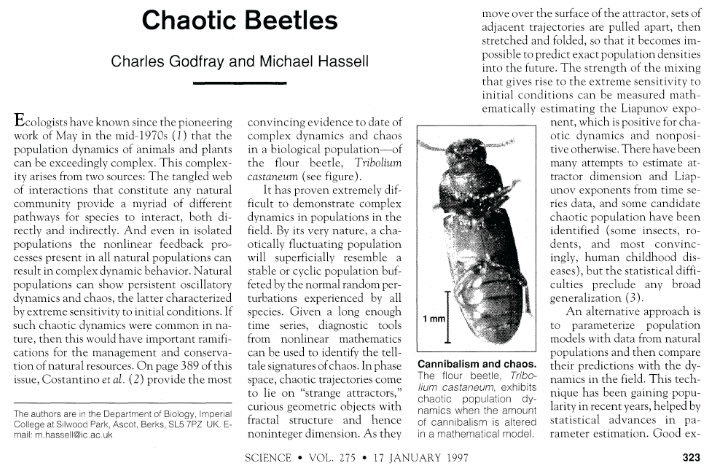
Larry Simpson and Dmitri A. Maslov

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

tion arthr netic woul the al In pothe mitoc quenc Crithi cent nucle as an tical Trypa the b by th fish p tutes trypa branc separ



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

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

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-

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

- Our data are the pages *Science* from 1880-2002 (from JSTOR)
- No reliable punctuation, meta-data, or references.
- Note: this is just a subset of JSTOR's archive.

D. Blei, 2008

Example Output: 4 Topics

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

Columns sorted by probability of word given topic.

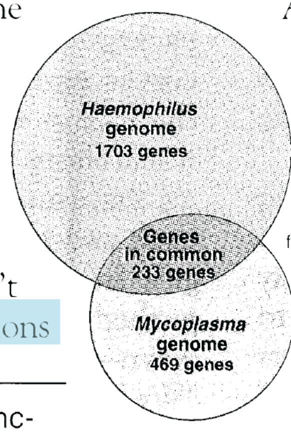
D. Blei, 2008

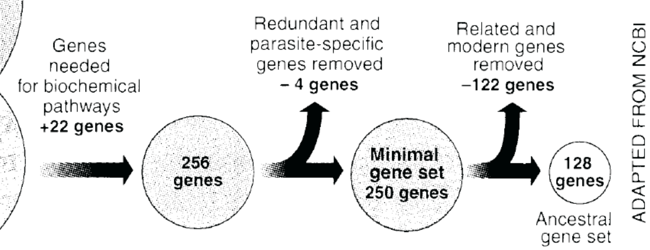
LDA: Intuition

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an





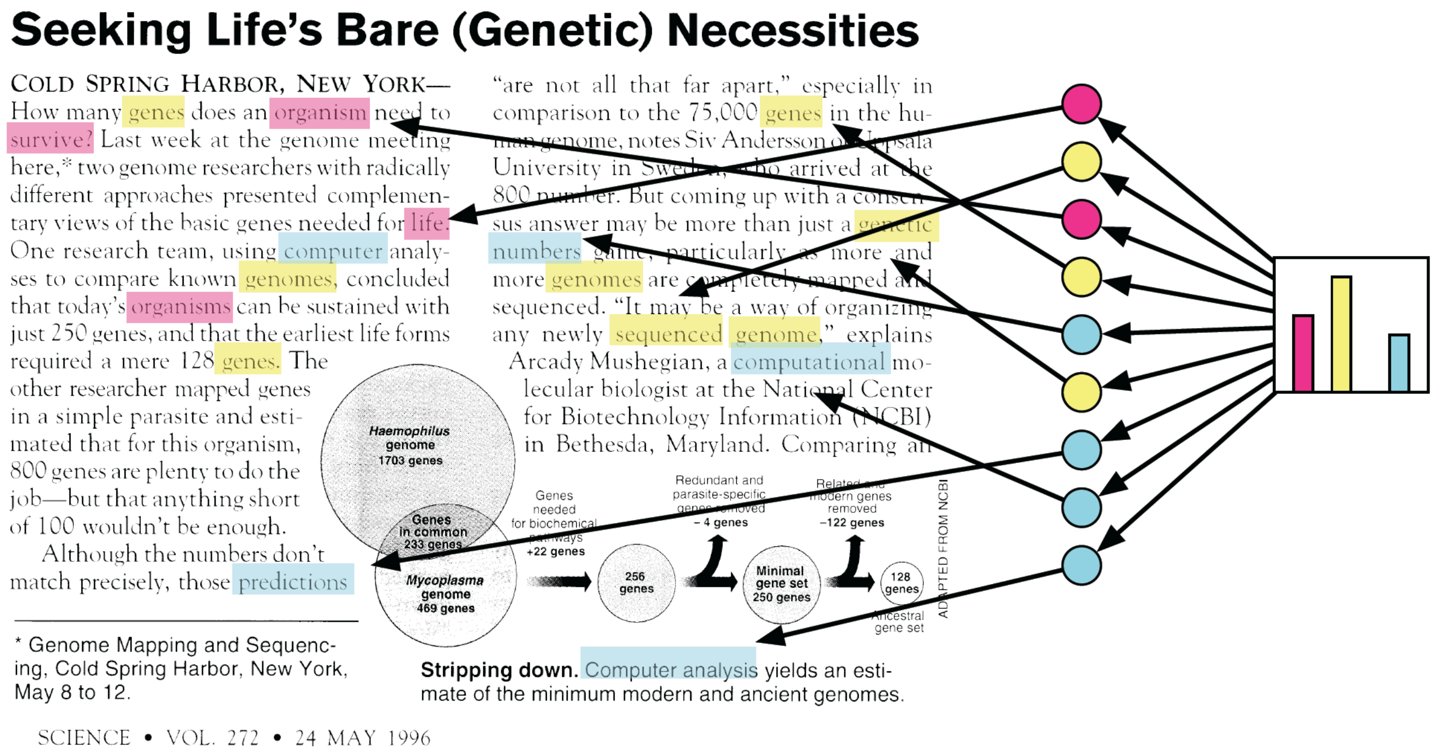
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Every document discusses a mixture of multiple topics.

D. Blei, 2008

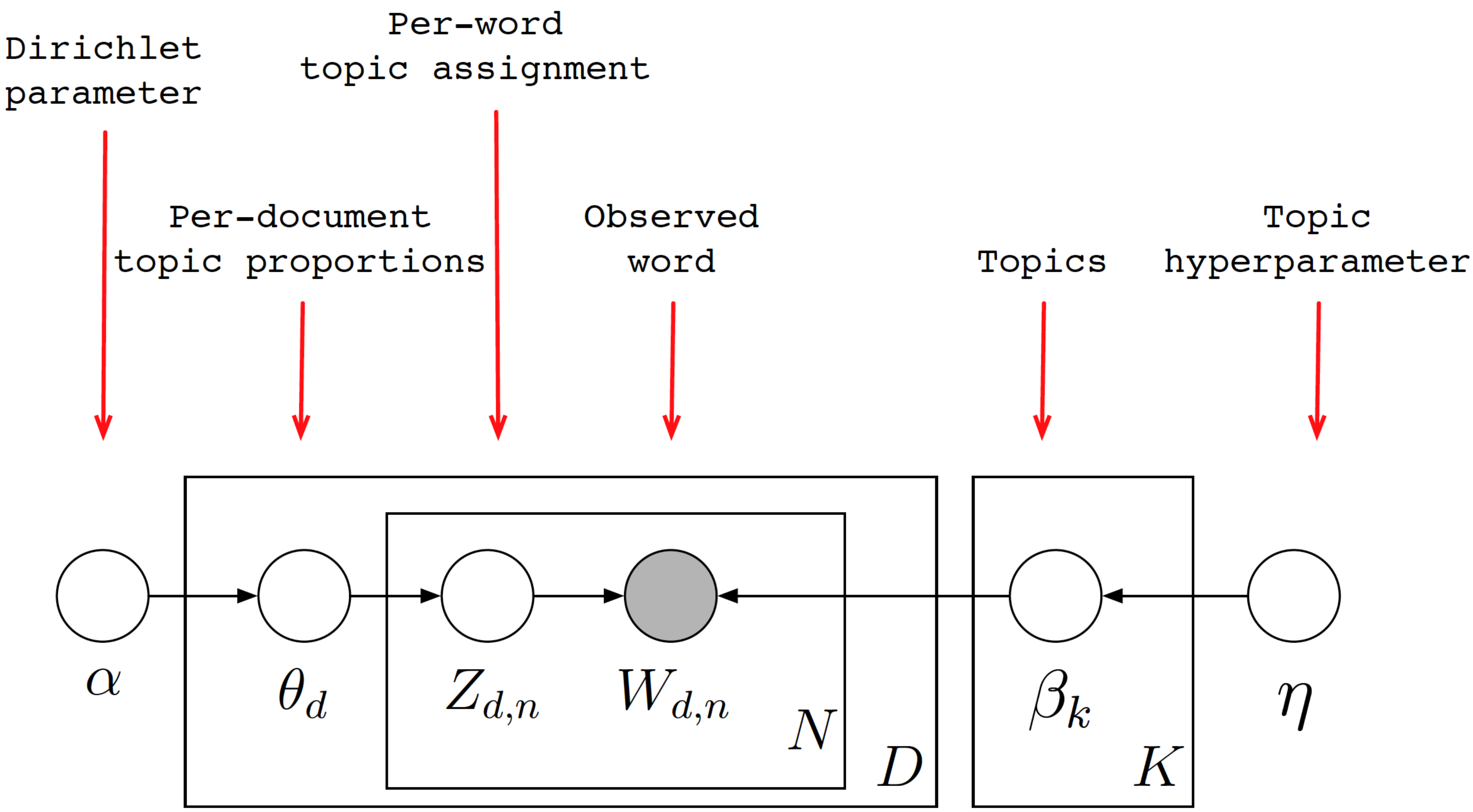
LDA: Generative Model



- Cast these intuitions into a generative probabilistic process
- Each document is a random mixture of corpus-wide topics
- Each word is drawn from one of those topics

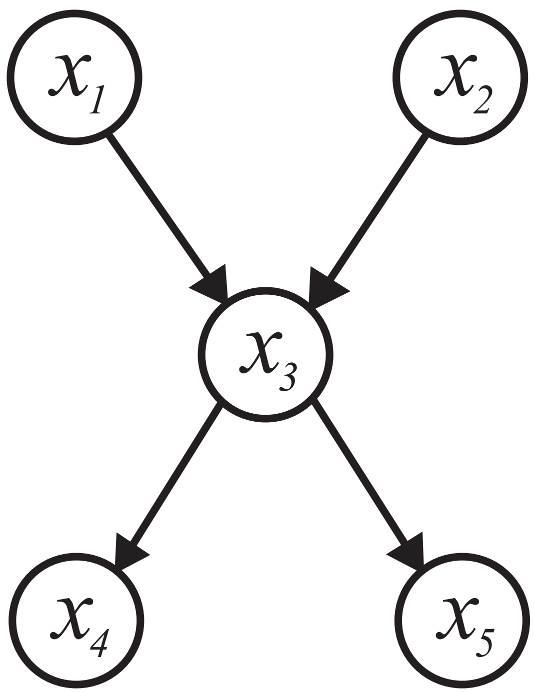
D. Blei, 2008

LDA: Graphical Model

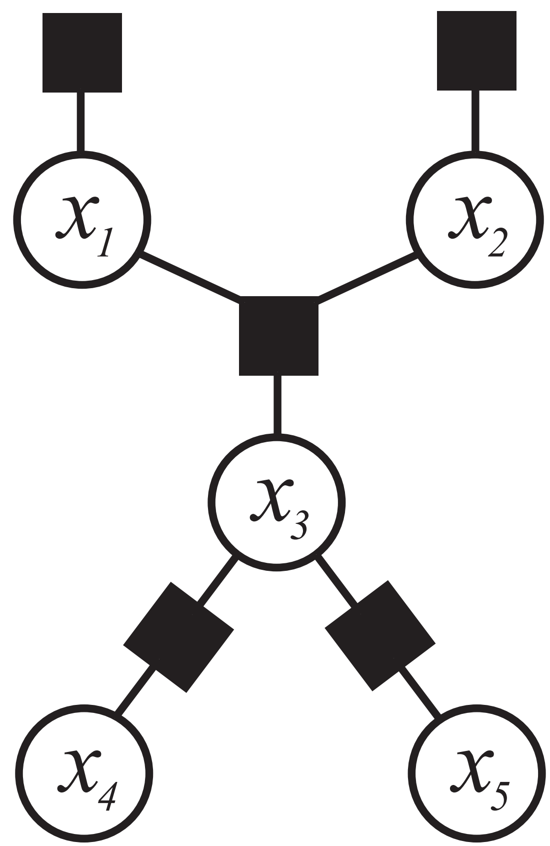


D. Blei, 2008

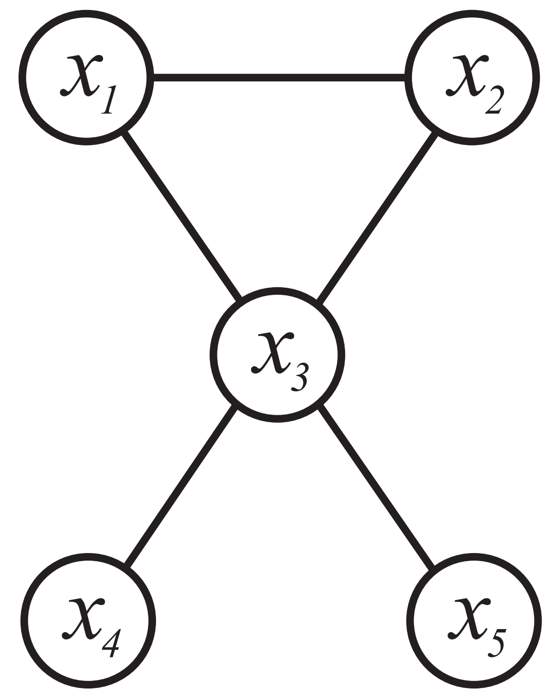
Graphical Models



Directed Bayesian Network



Factor Graph



Undirected Graphical Model

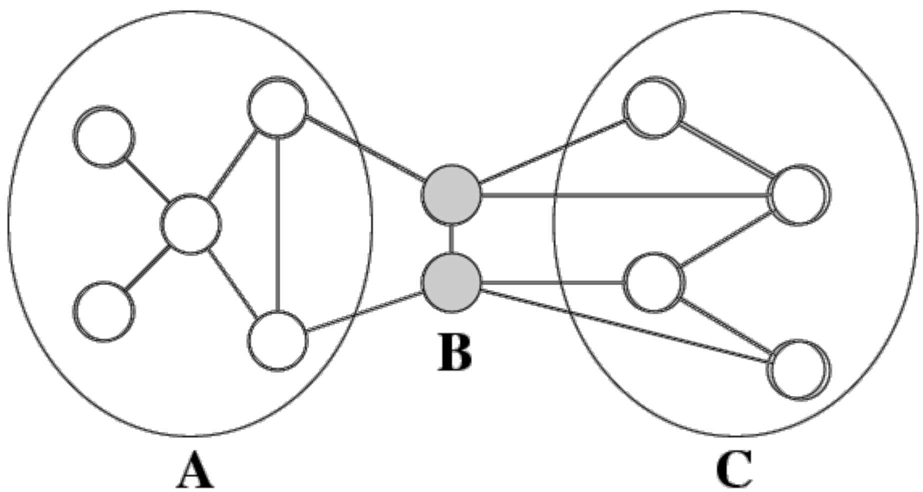
Undirected Graphical Models

An undirected graph $\mathcal{G}$ is defined by

$\mathcal{V}$ ⟶ set of N nodes $\{1, 2, \ldots, N\}$

$\mathcal{E}$ ⟶ set of edges (s, t) connecting nodes $s, t \in \mathcal{V}$

Nodes $s \in \mathcal{V}$ are associated with random variables x_s



Graph Separation

↕

Conditional Independence

$$p(x_A, x_C | x_B) = p(x_A | x_B) p(x_C | x_B)$$

Inference in Graphical Models

$$p(x \mid y) = \frac{1}{Z} \prod_{s \in \mathcal{V}} \psi_s(x_s) \prod_{(s,t) \in \mathcal{E}} \psi_{st}(x_s, x_t)$$

$y \longrightarrow$ observations (implicitly encoded via compatibilities)

Maximum a Posteriori (MAP) Estimates

$$\hat{x} = \arg\max_x \; p(x \mid y)$$

Posterior Marginal Densities

$$p_t(x_t \mid y) = \sum_{x_{\mathcal{V} \setminus t}} p(x \mid y)$$

- Provide both estimators and confidence measures
- Sufficient statistics for iterative *parameter estimation*

Why the Partition Function?

$$Z = \sum_{x} \prod_{s \in \mathcal{V}} \psi_s(x_s) \prod_{(s,t) \in \mathcal{E}} \psi_{st}(x_s, x_t)$$

Statistical Physics

- Sensitivity of physical systems to external stimuli

Hierarchical Bayesian Models

- Marginal likelihood of observed data
- Fundamental in hypothesis testing & model selection

Cumulant Generating Function

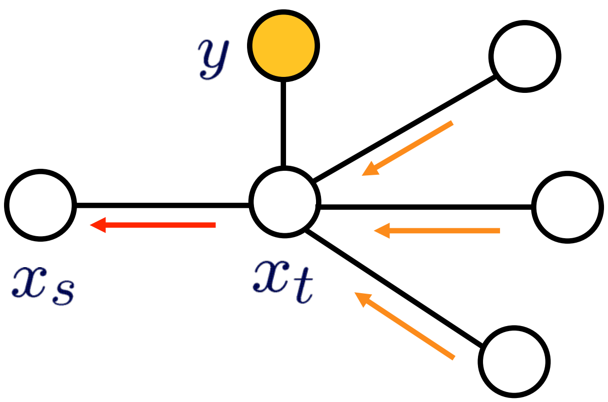
- For exponential families, derivatives with respect to parameters provide marginal statistics

PROBLEM: Computing Z in general graphs is NP-complete

Exact Inference

MESSAGES: Sum-product or belief propagation algorithm

$$m_{ts}(x_s) = \alpha \sum_{x_t} \psi_{st}(x_s, x_t)\psi_t(x_t, y) \prod_{u \in \Gamma(t) \backslash s} m_{ut}(x_t)$$



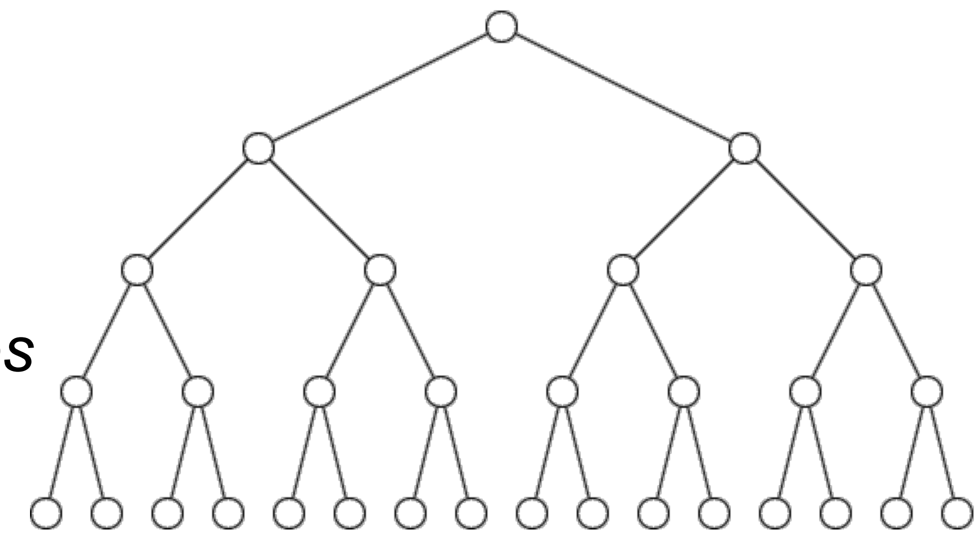
Computational cost:

N ⟶ *number of nodes*

M ⟶ *discrete states for each node*

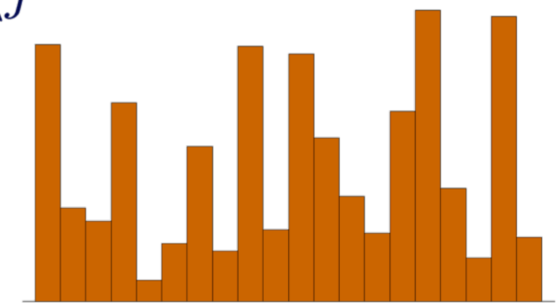
Belief Prop: $\mathcal{O}(NM^2)$

Brute Force: $\mathcal{O}(M^N)$



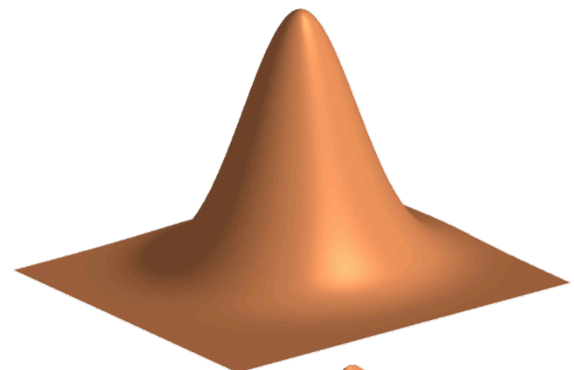
Continuous Variables

$$m_{ij}(x_j) \propto \int_{x_i} \psi_{j,i}(x_j, x_i)\psi_i(x_i, y) \prod_{k \in \Gamma(i) \setminus j} m_{ki}(x_i)\, dx_i$$



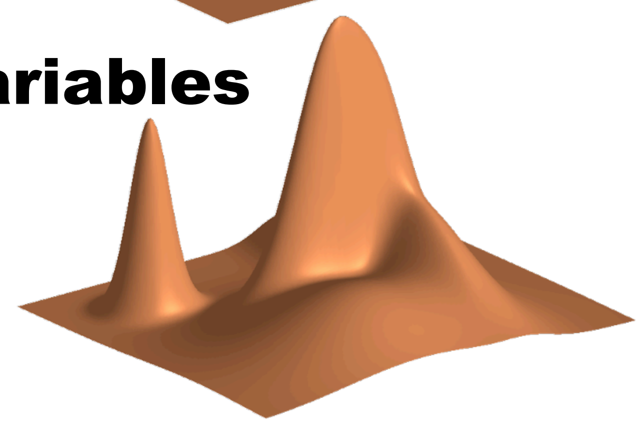
Discrete State Variables

- Messages are *finite vectors*
- Updated via matrix-vector products



Gaussian State Variables

- Messages are *mean & covariance*
- Updated via information Kalman filter



Continuous Non-Gaussian State Variables

- Closed parametric forms unavailable
- Discretization can be *intractable* even with 2 or 3 dimensional states

Variational Inference: An Example

$$p(x \mid y) = \frac{1}{Z} \prod_{(s,t) \in \mathcal{E}} \psi_{st}(x_s, x_t) \prod_{s \in \mathcal{V}} \psi_s(x_s, y)$$

- Choose a family of approximating distributions which is tractable. The simplest example:

$$q(x) = \prod_{s \in \mathcal{V}} q_s(x_s)$$

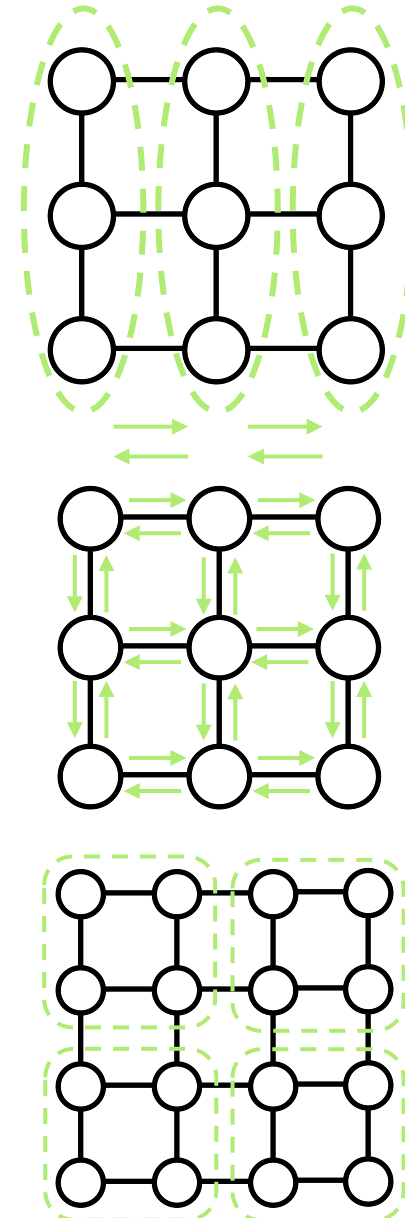
- Define a distance to measure the quality of different approximations. One possibility:

$$D(q \mid\mid p) = \sum_x q(x) \log \frac{q(x)}{p(x \mid y)}$$

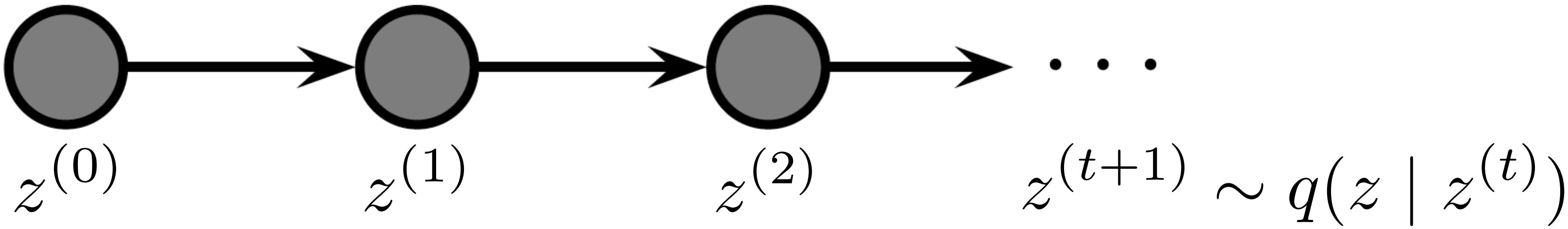
- Find the approximation minimizing this distance

Advanced Variational Methods

- Exponential families
- Mean field methods: naïve and structured
- Variational EM for parameter estimation
- Loopy belief propagation (BP)
- Bethe and Kikuchi entropies
- Generalized BP, fractional BP
- Convex relaxations and bounds
- MAP estimation and linear programming
- ………



Markov Chain Monte Carlo



- At each time point, state $z^{(t)}$ is a configuration of *all the variables in the model:* parameters, hidden variables, etc.
- We design the transition distribution $q(z \mid z^{(t)})$ so that the chain is *irreducible* and *ergodic*, with a unique stationary distribution $p^*(z)$

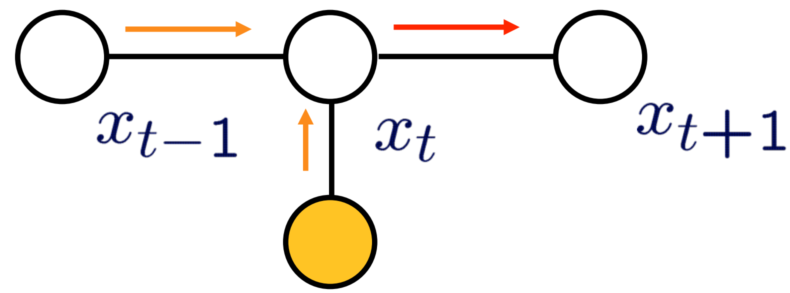
$$p^*(z) = \int_{\mathcal{Z}} q(z \mid z')p^*(z')\, dz'$$

- For learning, the target equilibrium distribution is usually the posterior distribution given data *x*: $p^*(z) = p(z \mid x)$
- Popular recipes: *Metropolis-Hastings and Gibbs samplers*

Sequential Monte Carlo

Particle Filters, Condensation, Survival of the Fittest,…

- Nonparametric approximation to optimal BP estimates
- Represent messages and posteriors using a set of samples, found by simulation



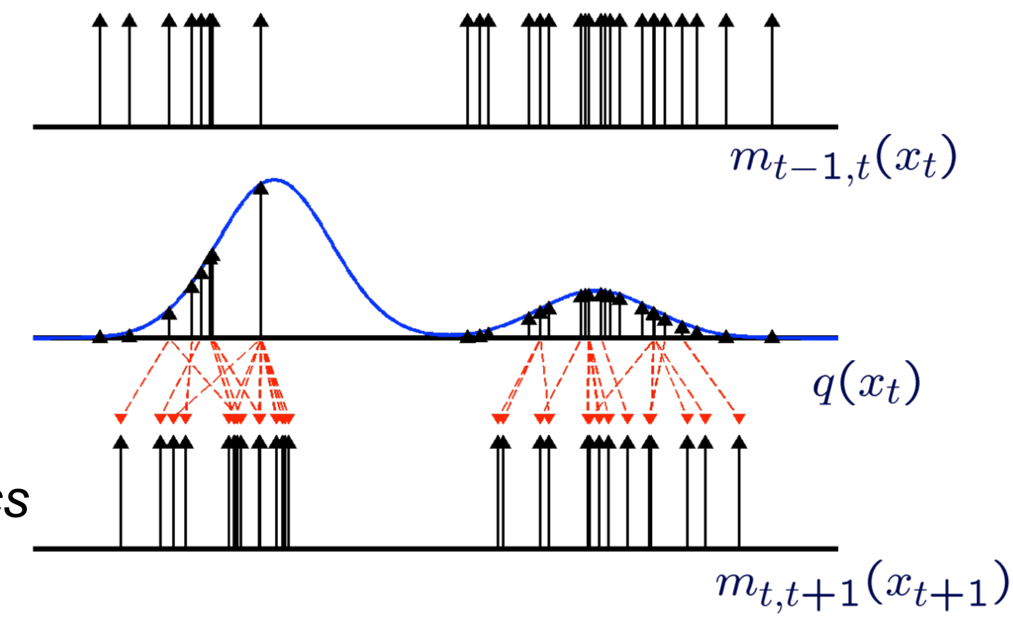
Sample-based density estimate

↓

Weight by observation likelihood

↓

Resample & propagate by dynamics



Course Evaluation

Homeworks: 60%

- Four equally weighted assignments
- Each assignment available for two weeks before due date
- Combine mathematical derivations, algorithm design, programming, and analysis of real datasets
  - Multiscale models of images, objects, visual scenes
  - Particle filters for localization and tracking
  - Topic models of text document collections
  - …

Final Project: 40%

- Proposal: 1-3 pages, due on March 22 (5%)
- Presentation: ~10 minutes, on May 7 (10%)
- Conference-style technical report, due on May 13 (25%)

Final Projects

Best case: Application of course material to your own area of research

Key Requirements: Novel use of graphical models

- Identify a family of graphical models suitable for a particular application, try baseline learning algorithms
- Propose, develop, and experimentally test a new type of graphical learning or inference algorithm
- Experimentally compare different models or algorithms on an interesting, novel dataset
- **There will not be a list of projects to choose from.** You must propose your own (with the instructor's advice). We will include pointers to many research papers with relevant applications.

Changes from Previous Years

- Readings from books & in-depth tutorials, not recent research papers. *More accessible.*
- No reading comments or student presentation of research papers. *Course staff will lecture.*
- Homework assignments require *mathematical derivations and algorithm implementation.*
- Subject matter: *Probabilistic Graphical Models*
  - Fall 2011 topic was *Applied Bayesian Nonparametrics*, may repeat for credit
  - Spring 2010 topics similar. You are welcome to (officially) audit, but see me about taking for credit.

Textbook & Readings

An Introduction to Probabilistic Graphical Models

Michael I. Jordan
University of California, Berkeley

- Draft textbook by Michael I. Jordan, available as a printed course reader, more details soon…
- Variational tutorial by Wainwright and Jordan (2008)
- Background chapter of Prof. Sudderth's thesis
- Tutorial articles on Markov chain Monte Carlo, particle filters
- A few other papers for advanced topics…

Course Prerequisites

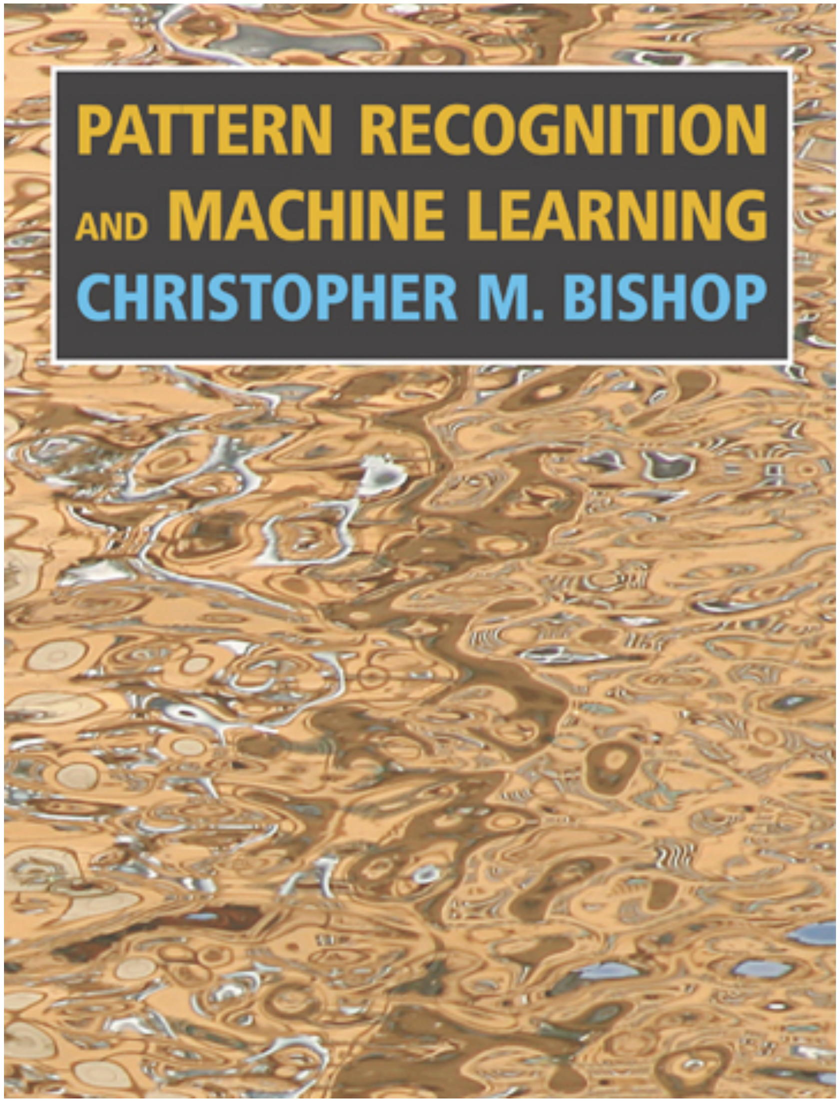
- A course in modern statistical machine learning
  - ➢ Brown CSCI 1950F: Intro to Machine Learning
  - ➢ Brown APMA 1690: Computational Probability and Statistics (also APMA 2690)
  - ➢ Possibly other classes or experience...
- Programming experience (Matlab, Java, …)
- Readings will require "mathematical maturity"
- Insufficient background by themselves:
  - ➢ Brown CSCI 1410: Introduction to AI
  - ➢ Traditional undergrad statistics (APMA 1650/1660)

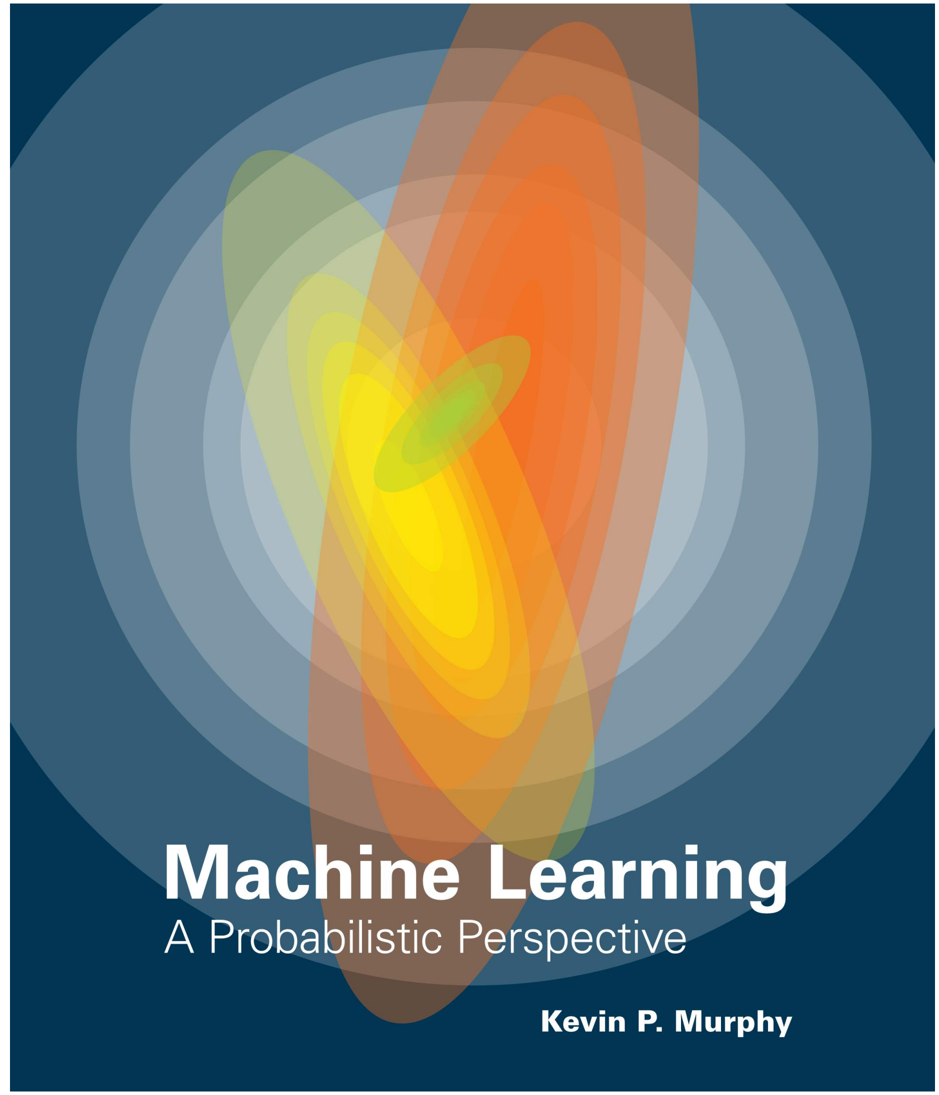
Prereq: Intro Machine Learning

	Supervised Learning	***Unsupervised Learning***
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

- Bayesian and frequentist estimation
- Model selection, cross-validation, overfitting
- Expectation-Maximization (EM) algorithm

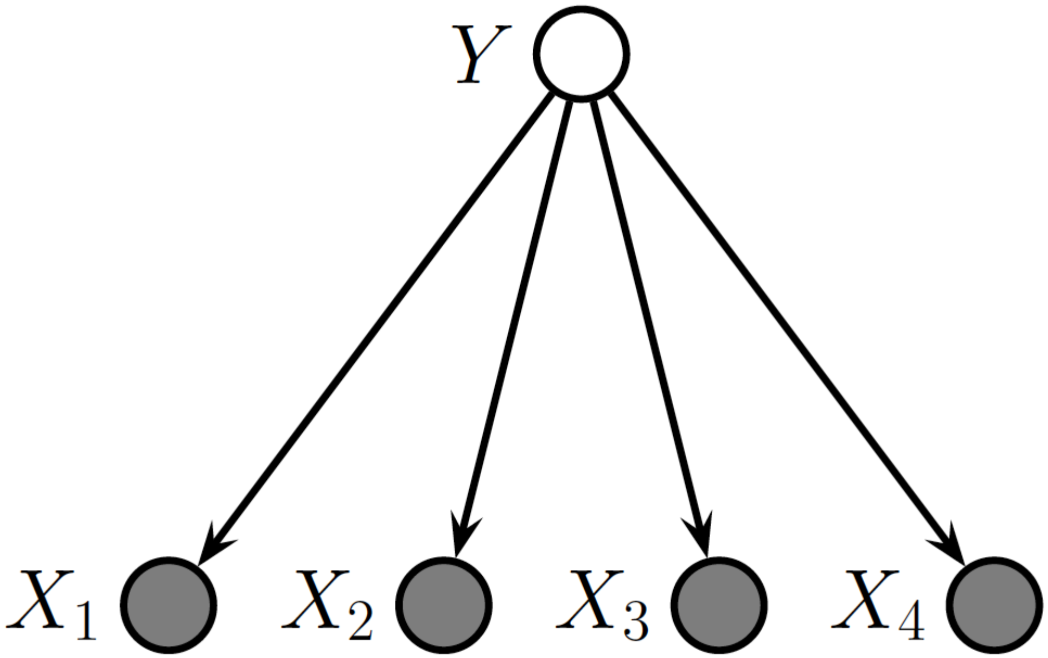
Background Material

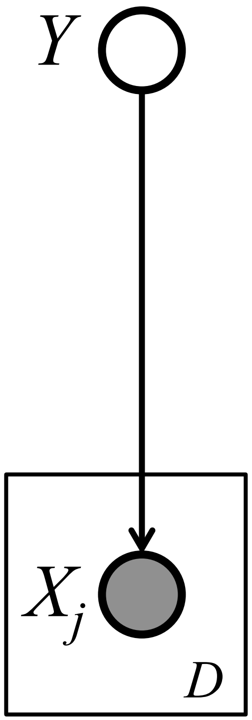




You will probably want a copy of one of these books…

Shading & Plate Notation





Plates denote replication of random variables

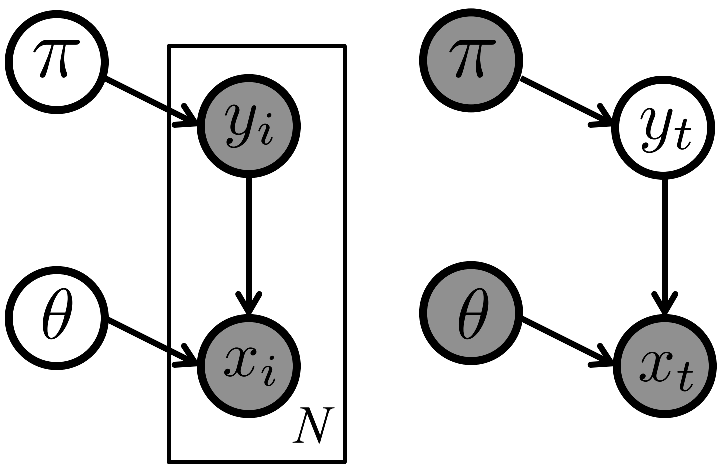
Naïve Bayes Inference: $$p(y, \mathbf{x}) = p(y) \prod_{j=1}^{D} p(x_j | y)$$

Convention: Shaded nodes are observed, open nodes are latent/hidden

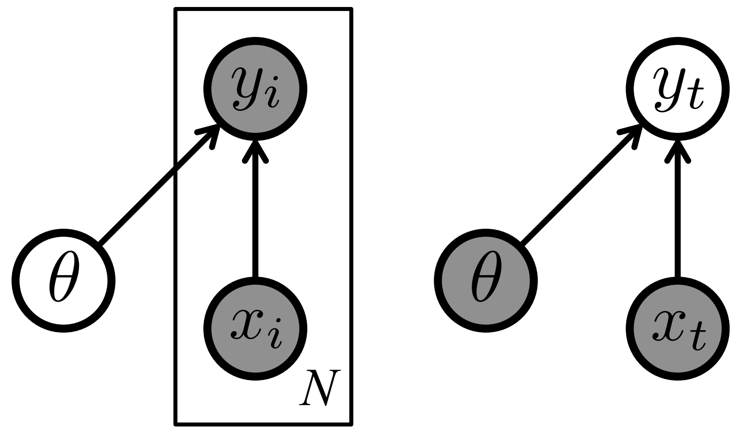
Supervised Learning

Generative ML or MAP Learning: *Naïve Bayes*

$$\max_{\pi,\theta} \; \log p(\pi) + \log p(\theta) + \sum_{i=1}^{N} [\log p(y_i \mid \pi) + \log p(x_i \mid y_i, \theta)]$$



Train *Test*



Train *Test*

Discriminative ML or MAP Learning: *Logistic regression*

$$\max_{\theta} \; \log p(\theta) + \sum_{i=1}^{N} \log p(y_i \mid x_i, \theta)$$

Learning via Optimization

ML Estimate: $$\hat{w} = \arg\min_w \; -\sum_i \log p(y_i \mid x_i, w)$$

MAP Estimate: $$\hat{w} = \arg\min_w \; -\log p(w) - \sum_i \log p(y_i \mid x_i, w)$$

Gradient vectors:

$$f : \mathbb{R}^M \rightarrow \mathbb{R}$$

$$\nabla_w f : \mathbb{R}^M \rightarrow \mathbb{R}^M$$

$$(\nabla_w f(w))_k = \frac{\partial f(w)}{\partial w_k}$$

Hessian matrices:

$$\nabla_w^2 f : \mathbb{R}^M \rightarrow \mathbb{R}^{M \times M}$$

$$(\nabla_w f(w))_{k,\ell} = \frac{\partial^2 f(w)}{\partial w_k \partial w_\ell}$$

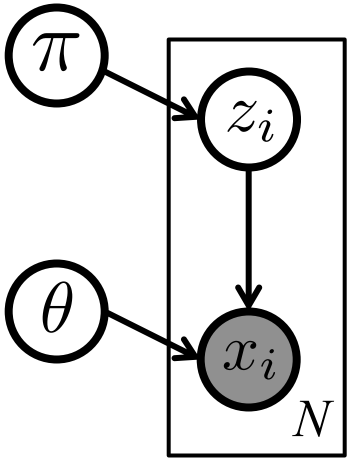
Optimization of Smooth Functions:

- *Closed form:* Find zero gradient points, check curvature
- *Iterative:* Initialize somewhere, use gradients to take steps towards better (by convention, smaller) values

Unsupervised Learning

Clustering:

$$\max_{\pi,\theta} \; \log p(\pi) + \log p(\theta) + \sum_{i=1}^{N} \log \left[\sum_{z_i} p(z_i \mid \pi) p(x_i \mid z_i, \theta) \right]$$

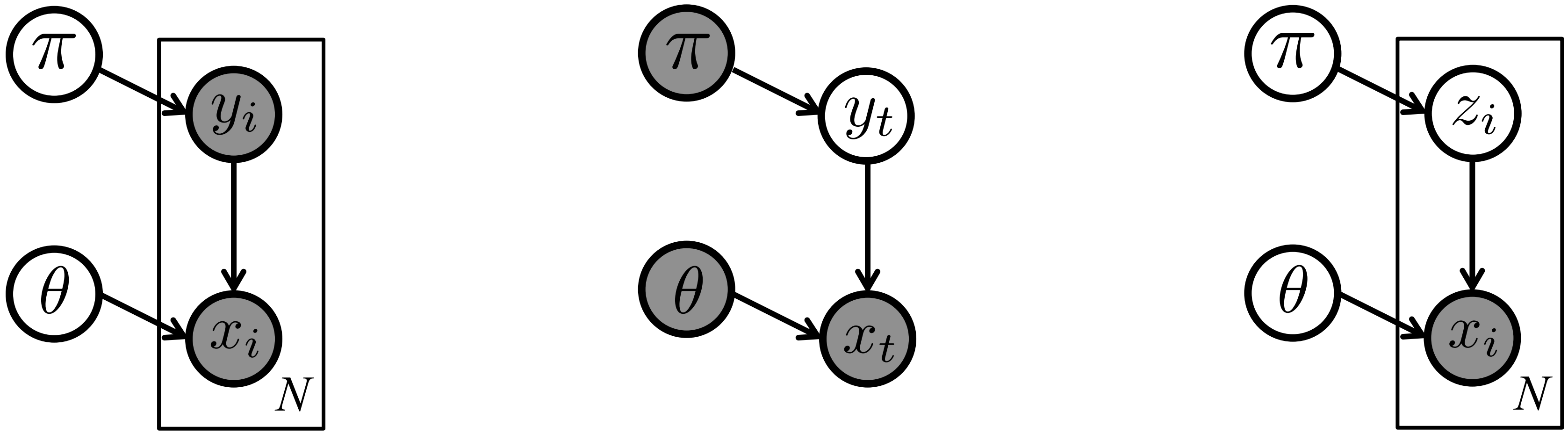


Dimensionality Reduction:

$$\max_{\pi,\theta} \; \log p(\pi) + \log p(\theta) + \sum_{i=1}^{N} \log \left[\int_{z_i} p(z_i \mid \pi) p(x_i \mid z_i, \theta) \; dz_i \right]$$

- No notion of training and test data: labels are *never* observed
- As before, *maximize* posterior probability of model parameters
- For hidden variables associated with each observation, we *marginalize* over possible values rather than estimating
  - Fully accounts for uncertainty in these variables
  - There is one hidden variable per observation, so cannot perfectly estimate even with infinite data
- Must use generative model (discriminative degenerates)

Expectation Maximization (EM)



Supervised Training | *Supervised Testing* | *Unsupervised Learning*

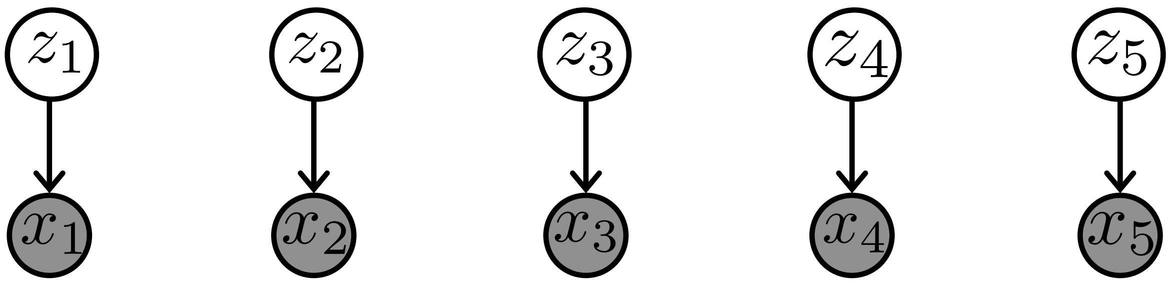
π, θ → *parameters (define low-dimensional manifold)*

$z_1, \ldots, z_N$ → *hidden data (locate observations on manifold)*

- **Initialization:** Randomly select starting parameters
- **E-Step:** Given parameters, find posterior of hidden data
  - Equivalent to test inference of full posterior distribution
- **M-Step:** Given posterior distributions, find likely parameters
  - Similar to supervised ML/MAP training
- **Iteration:** Alternate E-step & M-step until convergence

Gaussian Mixture Models vs. HMMs

Mixture Model

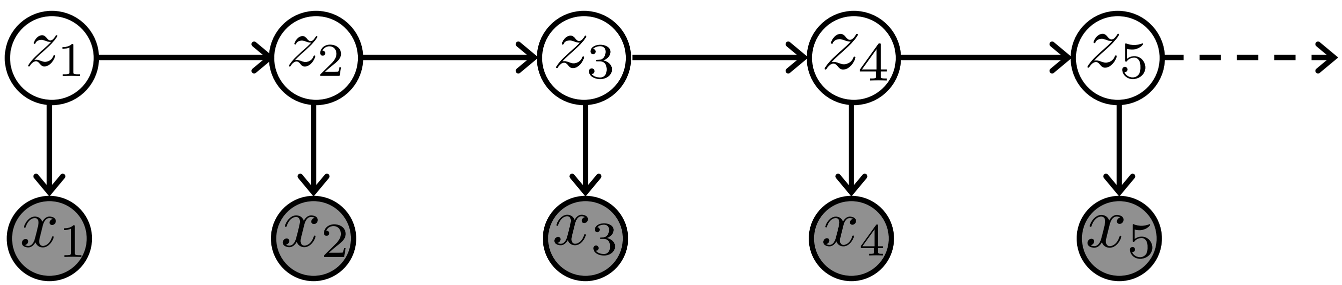


$z_i \in \{1, \ldots, K\}$

$$p(z_i \mid \pi, \mu, \Sigma) = \mathrm{Cat}(z_i \mid \pi)$$

$$p(x_i \mid z_i, \pi, \mu, \Sigma) = \mathrm{Norm}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

Hidden Markov Model



$$p(z_t \mid \pi, \mu, \Sigma, z_{t-1}, z_{t-2}, \ldots) = \mathrm{Cat}(z_t \mid \pi_{z_{t-1}})$$

$$p(x_t \mid z_t, \pi, \mu, \Sigma) = \mathrm{Norm}(x_t \mid \mu_{z_t}, \Sigma_{z_t})$$

Recover mixture model when all rows of state transition matrix are equal.

Probabilistic PCA & Factor Analysis

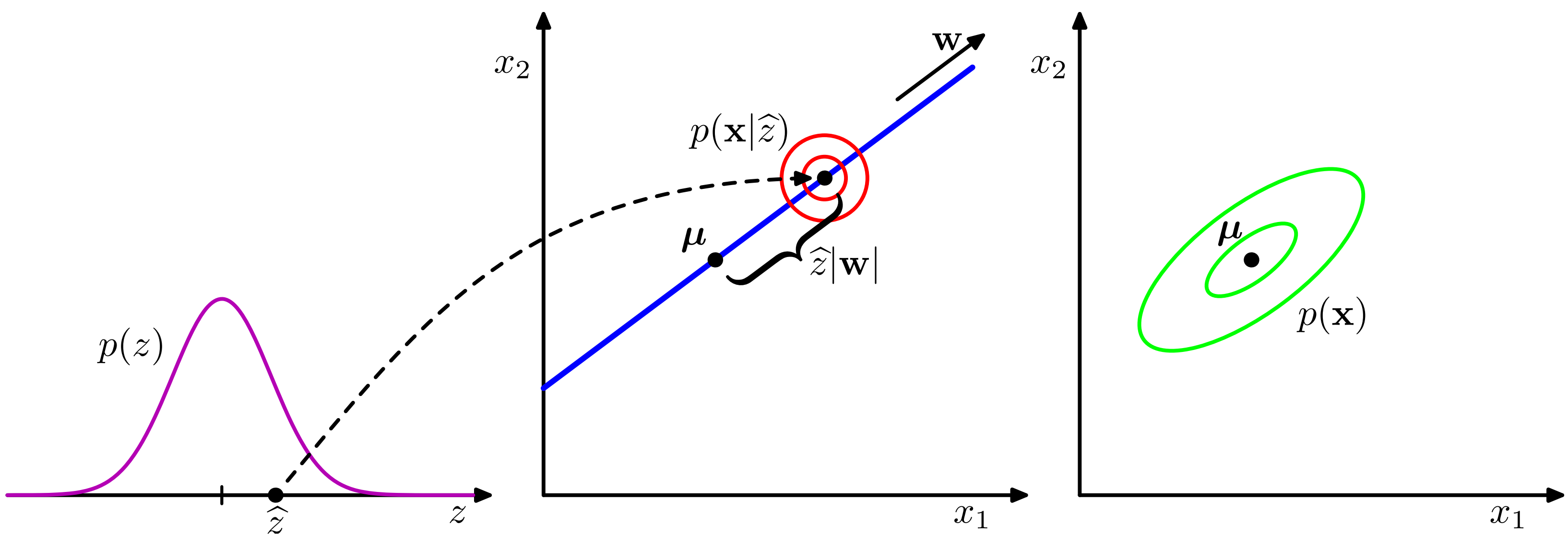
- **Both Models:** Data is a linear function of low-dimensional latent coordinates, plus Gaussian noise

$$p(x_i \mid z_i, \theta) = \mathcal{N}(x_i \mid W z_i + \mu, \Psi) \qquad p(z_i \mid \theta) = \mathcal{N}(z_i \mid 0, I)$$

$$p(x_i \mid \theta) = \mathcal{N}(x_i \mid \mu, WW^T + \Psi)$$

low rank covariance parameterization

- **Factor analysis:** Ψ is a general diagonal matrix
- **Probabilistic PCA:** $\Psi = \sigma^2 I$ is a multiple of identity matrix



C. Bishop, Pattern Recognition & Machine Learning

A Quick Poll

Administration

Registration: **E-mail sudderth@cs.brown.edu with**

- Your name and CS logon
- Your department, major, and year
- Your background in statistical machine learning
  - If you've taken Brown courses, just say which ones
  - Otherwise, a few sentences about your experience

Course webpage: Up now, watch for more information

http://cs.brown.edu/courses/csci2950-p/index.html

Readings for Tuesday:

- *Graphical Models*, M. Jordan, Stat. Science 2004.
- Chapter 2 from textbook (available soon)