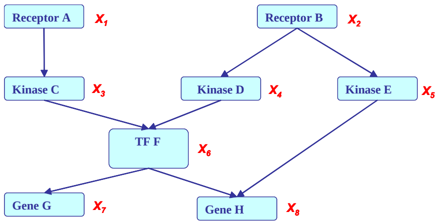
Advanced Introduction to Machine Learning

10715, Fall 2014

Intro to Graphical Models

Eric Xing

Lecture 13, October 15, 2014



Reading:

Multivariate Distribution in High-D Space

- A possible world for cellular signal transduction:

Receptor A X_1

Receptor B X_2

Kinase C X_3

Kinase D X_4

Kinase E X_5

TF F X_6

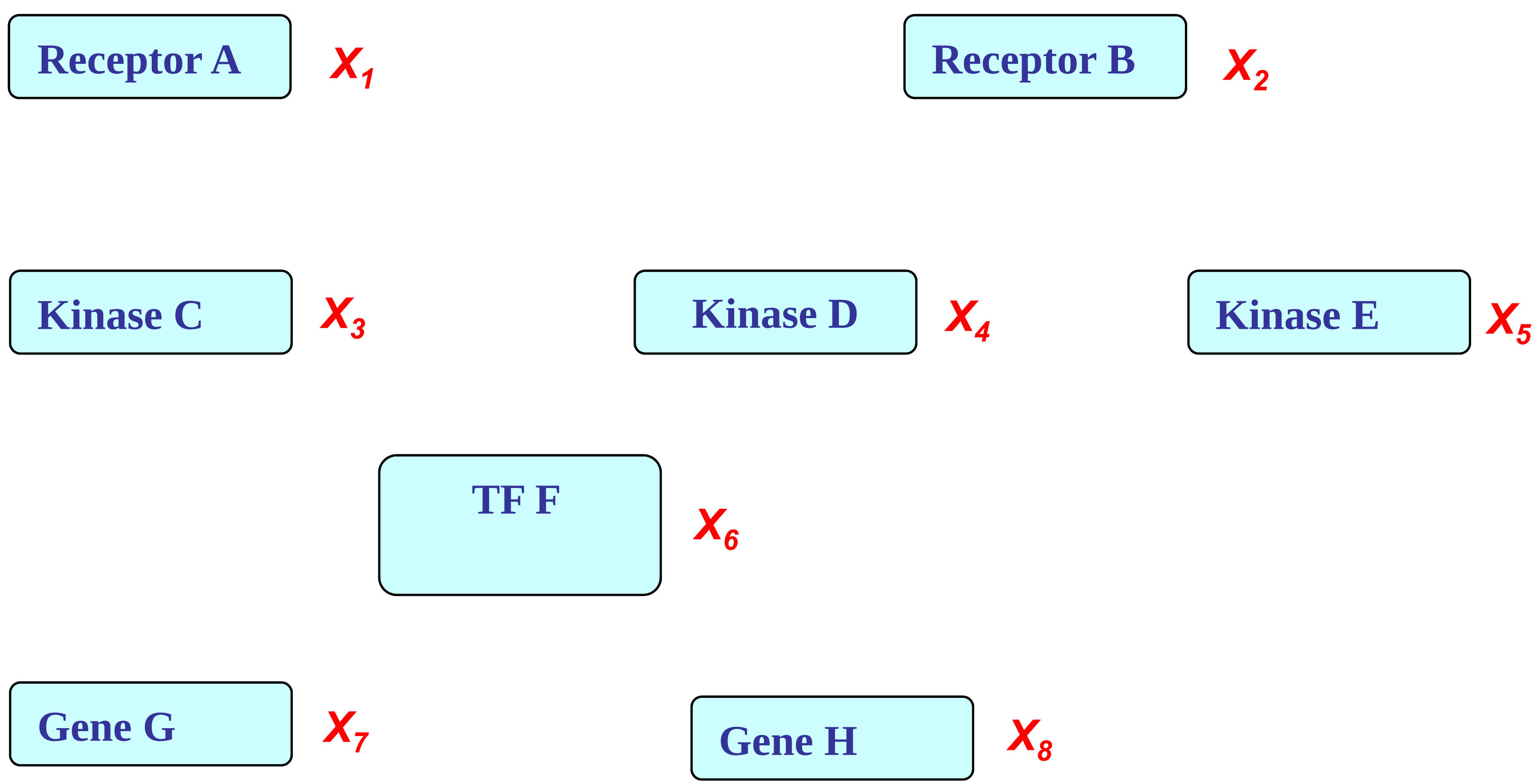
Gene G X_7

Gene H X_8

What is a Graphical Model?

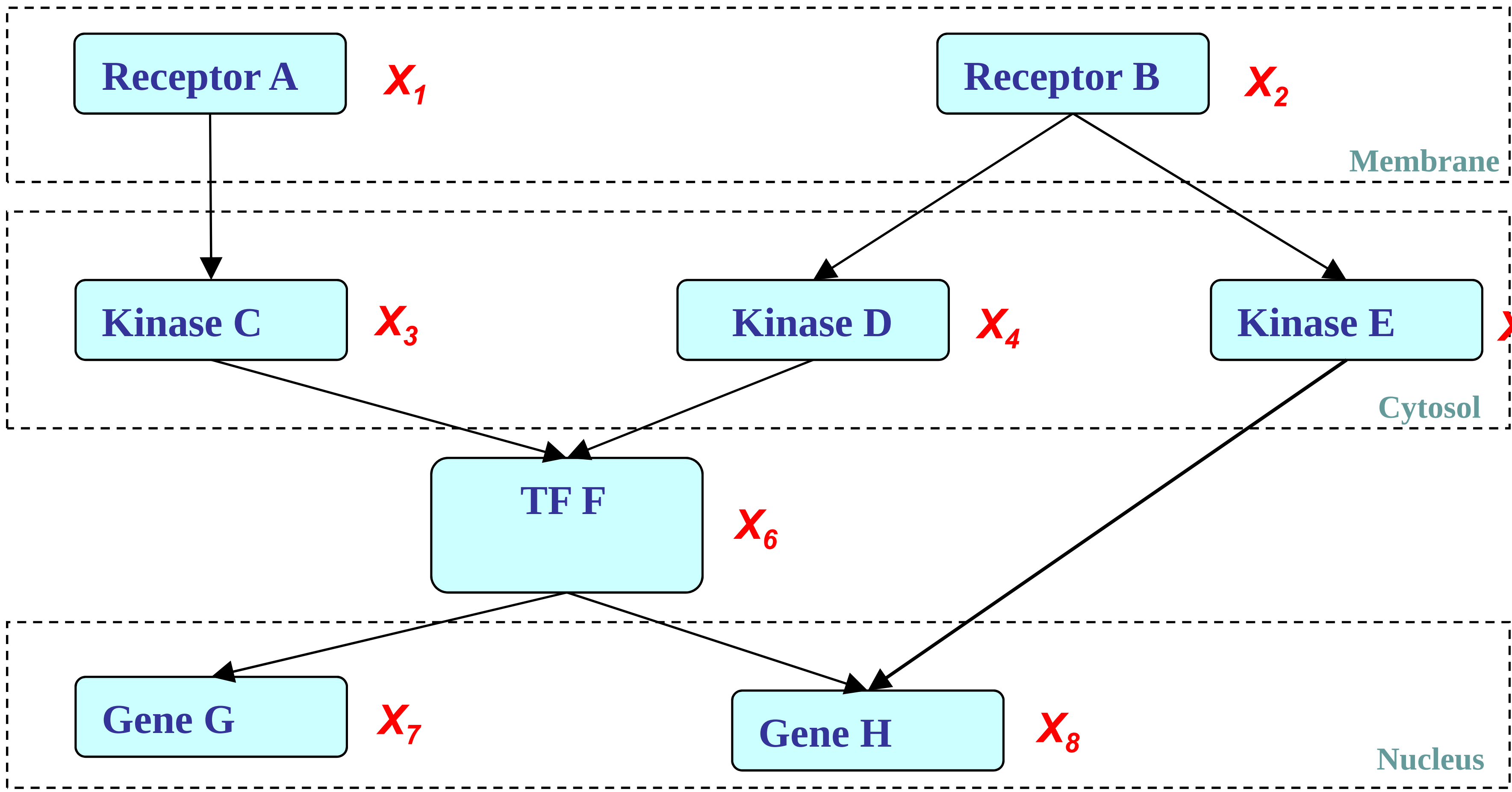
--- example from a signal transduction pathway

- A possible world for cellular signal transduction:



GM: Structure Simplifies Representation

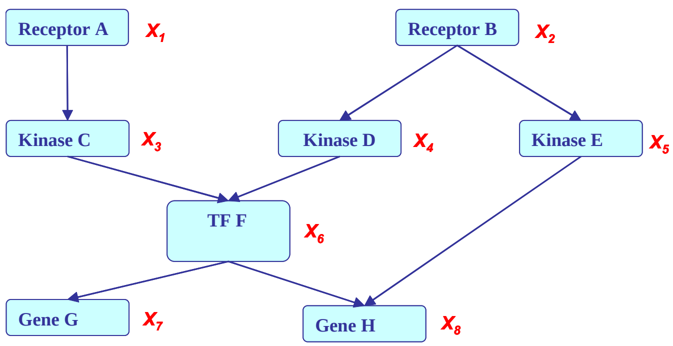
- Dependencies among variables



Probabilistic Graphical Models, con'd

$P(Y_t | Y_{t-1}, \cancel{X_{t}\ldots X_{t-1}})$

- If X_i's are **conditionally independent** (as described by a **PGM**), the joint can be factored to a product of simpler terms, e.g.,



$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$

$$= P(X_1)\, P(X_2)\, P(X_3 | X_1)\, P(X_4 | X_2)\, P(X_5 | X_2)\, P(X_6 | X_3, X_4)\, P(X_7 | X_6)\, P(X_8 | X_5, X_6)$$

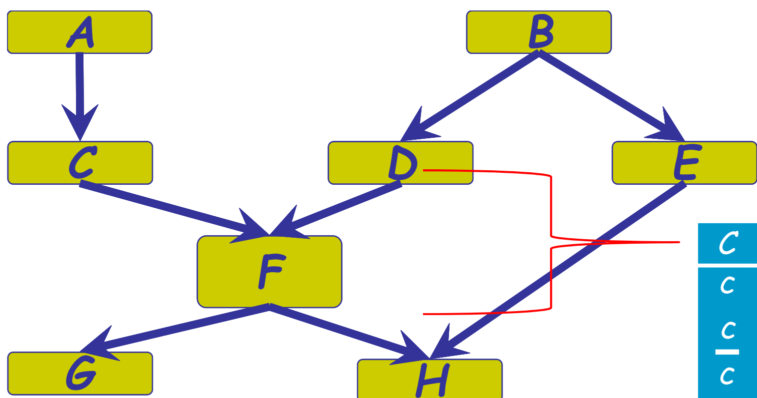
- Why we may favor a PGM?
  - Representation cost: how many probability statements are needed?

    2+2+4+4+4+8+4+8=36, an 8-fold reduction from 2^8!

  - Algorithms for systematic and efficient inference/learning computation
    - Exploring the graph structure and probabilistic (e.g., Bayesian, Markovian) semantics
  - Incorporation of domain knowledge and causal (logical) structures

Specification of a ~~BN~~ GM

- There are two components to any GM:
  - the *qualitative* specification
  - the *quantitative* specification



C	D	P(F | C,D)	
c	d	0.9	0.1
c	$\bar{d}$	0.2	0.8
$\bar{c}$	d	0.9	0.1
$\bar{c}$	$\bar{d}$	0.01	0.99

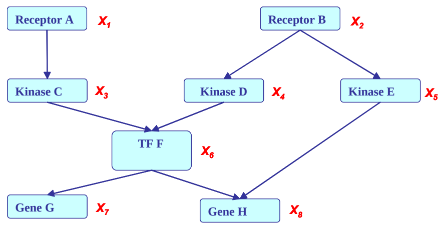
Qualitative Specification

- Where does the qualitative specification come from?
  - Prior knowledge of causal relationships
  - Prior knowledge of modular relationships
  - Assessment from experts
  - Learning from data
  - We simply link a certain architecture (e.g. a layered graph)
  - …

Two types of GMs

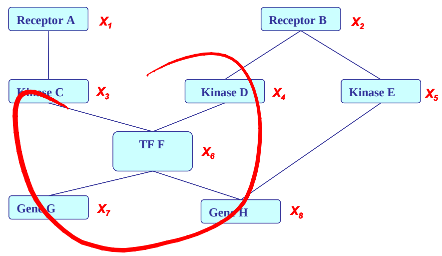
- **Directed edges give causality relationships (Bayesian Network or Directed Graphical Model):**

$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$
$$= P(X_1)\, P(X_2)\, P(X_3 | X_1)\, P(X_4 | X_2)\, P(X_5 | X_2)\, P(X_6 | X_3, X_4)\, P(X_7 | X_6)\, P(X_8 | X_5, X_6)$$



- **Undirected edges simply give correlations between variables (Markov Random Field or Undirected Graphical model):**

$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$
$$= 1/Z \exp\{E(X_1)+E(X_2)+E(X_3, X_1)+E(X_4, X_2)+E(X_5, X_2) + E(X_6, X_3, X_4)+E(X_7, X_6)+E(X_8, X_5, X_6)\}$$



Bayesian Network:

- A BN is a directed graph whose nodes represent the random variables and whose edges represent direct influence of one variable on another.

- It is a data structure that provides the skeleton for representing **a joint distribution** compactly in a **factorized** way;

- It offers a compact representation for **a set of conditional independence assumptions** about a distribution;

- We can view the graph as encoding a generative sampling process executed by nature, where the value for each variable is selected by nature using a distribution that depends only on its parents. In other words, each variable is a stochastic function of its parents.

$p(x_1, x_2) = p(x_1)\, p(x_2|x_1)$ $\quad p(x_1 \ldots x_n)$ $\quad X_1 \perp X_2$

Bayesian Network: Factorization Theorem

$X_1 \perp X_2 \mid X_3$

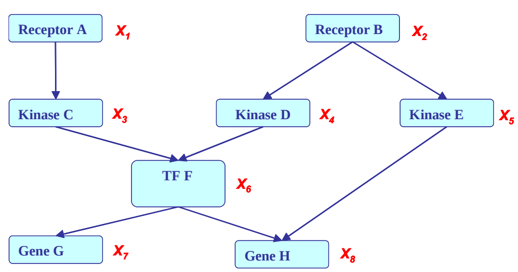
- **Theorem:**

  Given a DAG, The most general form of the probability distribution that is consistent with the graph factors according to "node given its parents":

$$P(\mathbf{X}) = \prod_{i=1:d} P(X_i \mid \mathbf{X}_{\pi_i})$$

  where $\mathbf{X}_{\pi_i}$ is the set of parents of X_i, d is the number of nodes (variables) in the graph.

G $\quad P\{V_i\}$



$$P(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8)$$
$$= P(X_1)\, P(X_2)\, P(X_3 | X_1)\, P(X_4 | X_2)\, P(X_5 | X_2)\, P(X_6 | X_3, X_4)\, P(X_7 | X_6)\, P(X_8 | X_5, X_6)$$

Bayesian Network: Conditional Independence Semantics

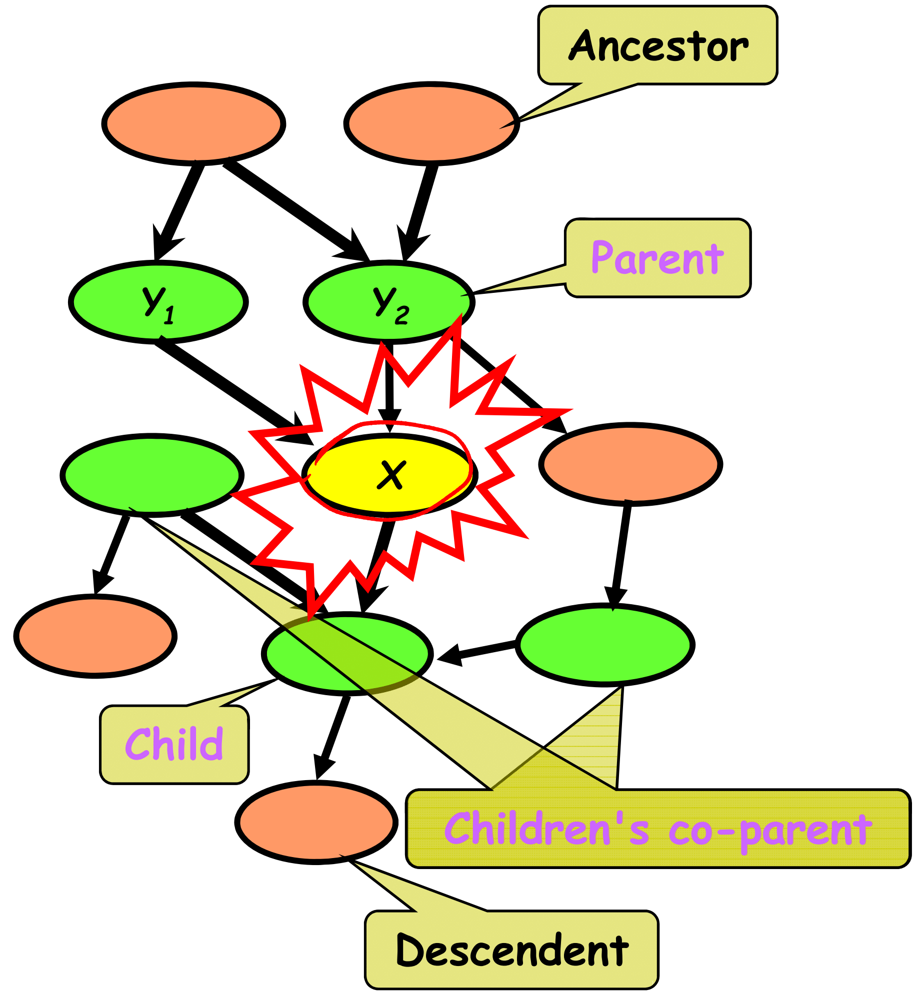
$P(X_i | X_{-i}) = P(X_i | X_{MB_i})$

Structure: *DAG*

- **Meaning: a node is conditionally independent of every other node in the network outside its Markov blanket**

- **Local conditional distributions (CPD) and the DAG completely determine the joint dist.**

- **Give causality relationships, and facilitate a generative process**

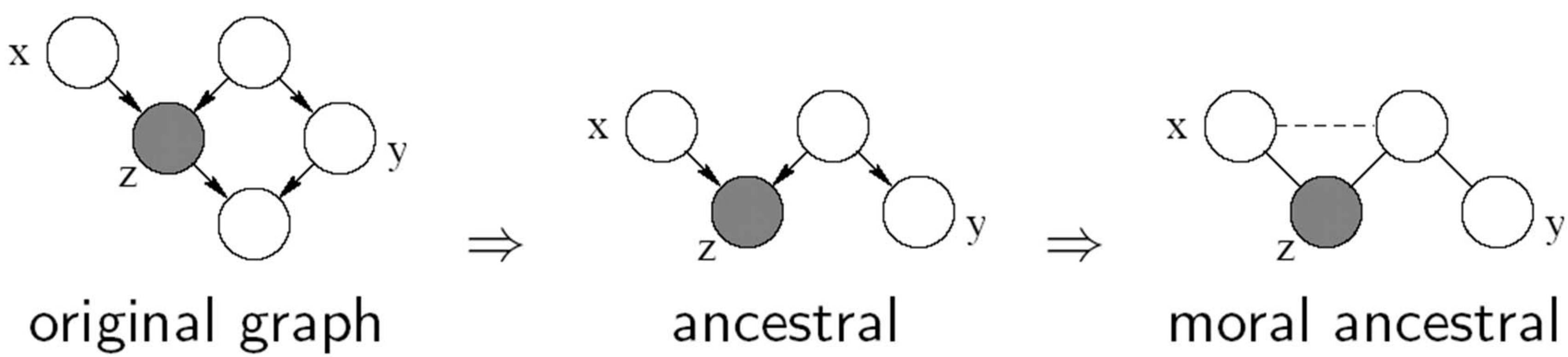


Graph separation criterion

- D-separation criterion for Bayesian networks (D for Directed edges):

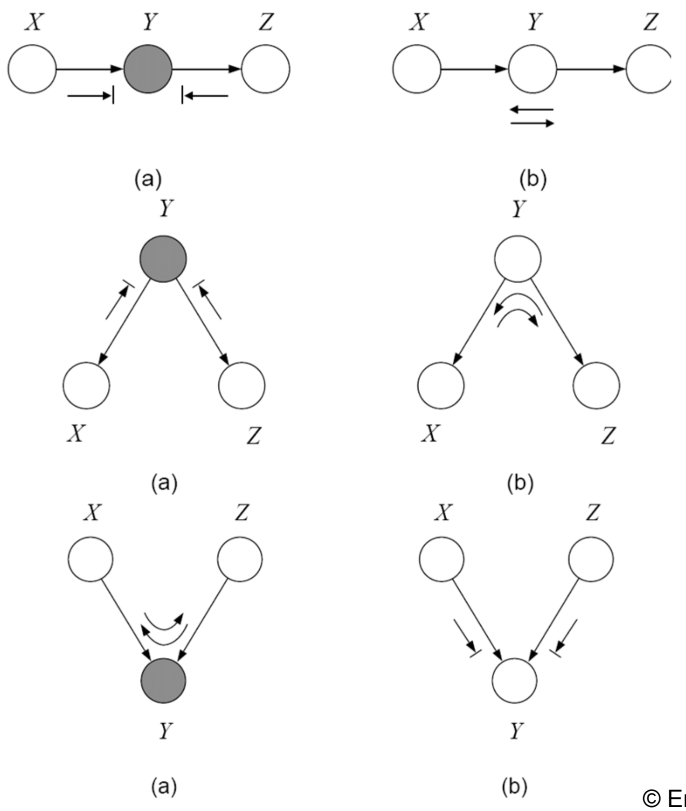
  Definition: variables x and y are *D-separated* (conditionally independent) given z if they are separated in the *moralized* ancestral graph

- Example:



Global Markov properties of DAGs

- X is **d-separated** (directed-separated) from Z given Y if we can't send a ball from any node in X to any node in Z using the "*Bayes-ball*" algorithm illustrated bellow (and plus some boundary conditions):



- **Defn: *I*(*G*)=all independence properties that correspond to d-separation:**

$$I(G) = \left\{ X \perp Z \middle| Y : \mathrm{dsep}_G(X; Z \middle| Y) \right\}$$

- **D-separation is sound and complete**

Towards quantitative specification of probability distribution

- Separation properties in the graph imply independence properties about the associated variables
- For the graph to be useful, any conditional independence properties we can derive from the graph should hold for the probability distribution that the graph represents

- **The Equivalence Theorem**

  For a graph G,

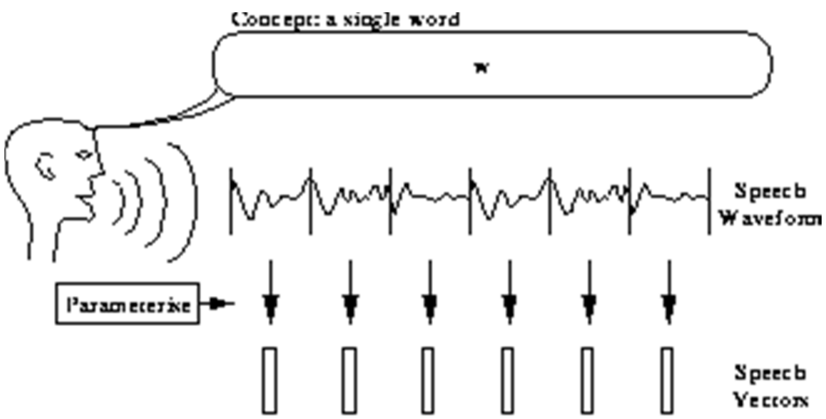
  Let $\mathcal{D}_1$ denote the family of all distributions that satisfy I(G),

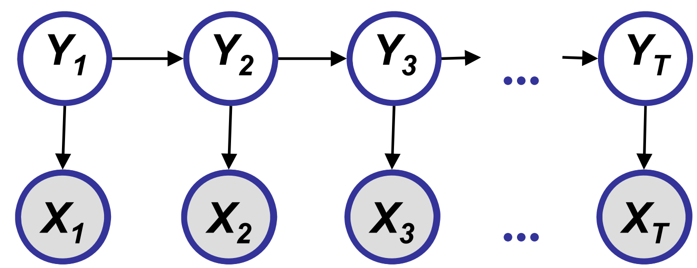
  Let $\mathcal{D}_2$ denote the family of all distributions that factor according to G,

  Then $\mathcal{D}_1 \equiv \mathcal{D}_2$.

Example

- Speech recognition





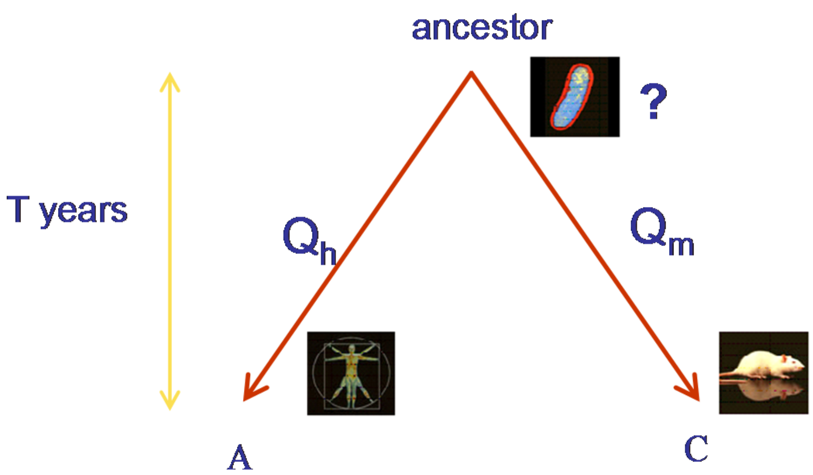
Hidden Markov Model

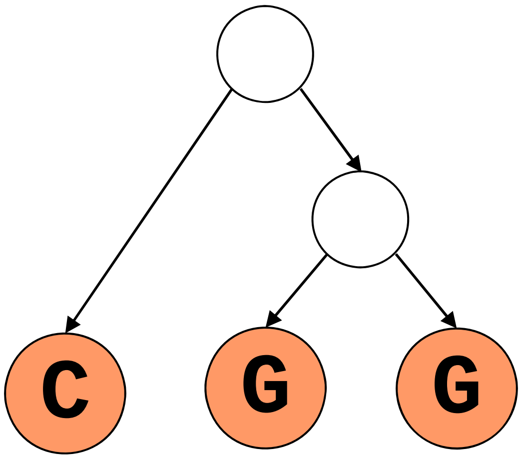
Knowledge Engineering

- Picking variables
  - Observed
  - Hidden
- Picking structure
  - CAUSAL
  - Generative
- Picking Probabilities
  - Zero probabilities
  - Orders of magnitudes
  - Relative values

Example, con'd

- Evolution





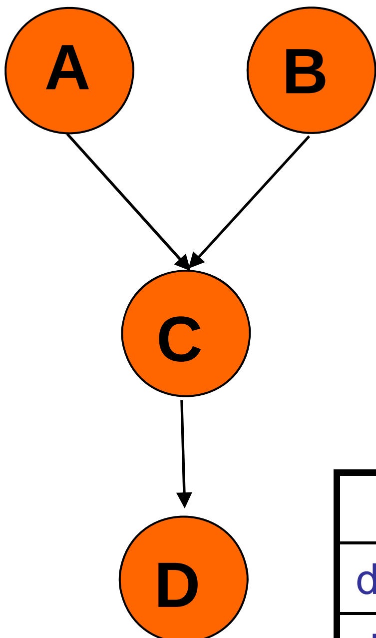
Tree Model

Conditional probability tables (CPTs)

a^0	0.75
a^1	0.25

b^0	0.33
b^1	0.67

$$P(a,b,c.d) = P(a)P(b)P(c|a,b)P(d|c)$$



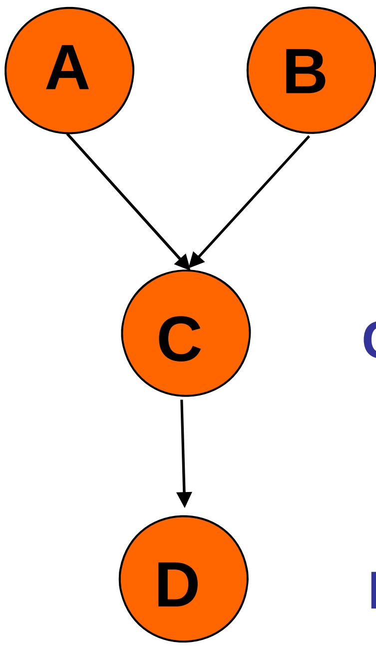
	a^0b^0	a^0b^1	a^1b^0	a^1b^1
c^0	0.45	1	0.9	0.7
c^1	0.55	0	0.1	0.3

	c^0	c^1
d^0	0.3	0.5
d^1	07	0.5

Conditional probability density func. (CPDs)

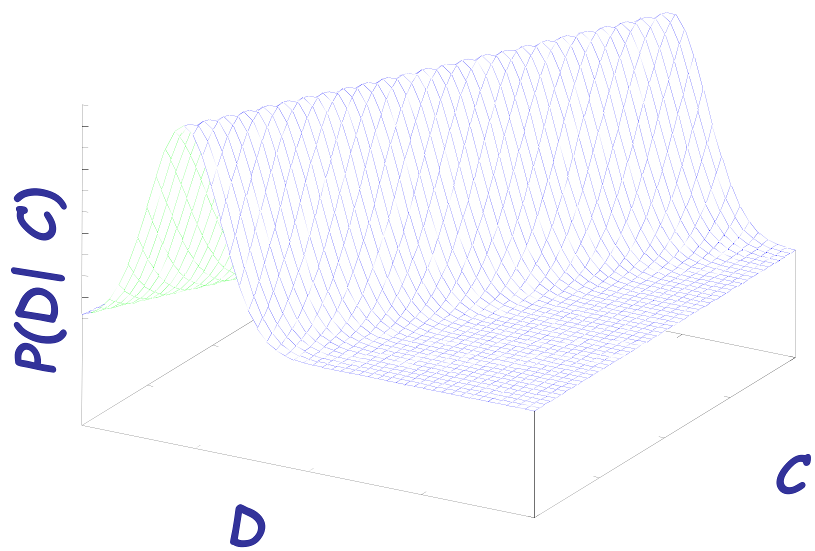
$A \sim N(\mu_a, \Sigma_a)$ $B \sim N(\mu_b, \Sigma_b)$

$$P(a,b,c.d) = P(a)P(b)P(c|a,b)P(d|c)$$

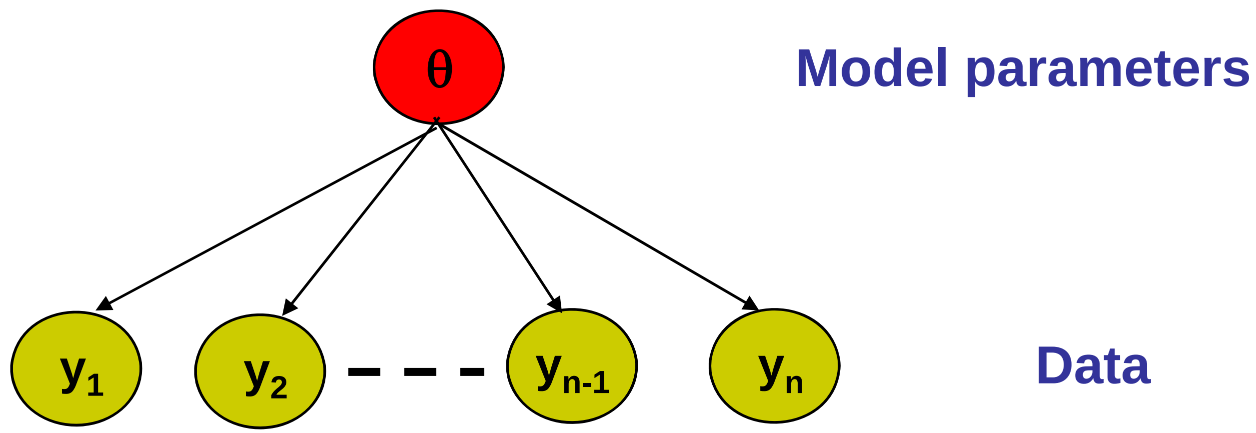


$C \sim N(A+B, \Sigma_c)$

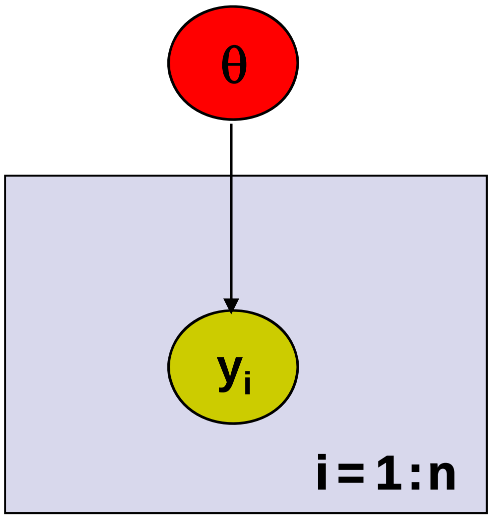
$D \sim N(\mu_a + C, \Sigma_a)$



Conditionally Independent Observations



"Plate" Notation



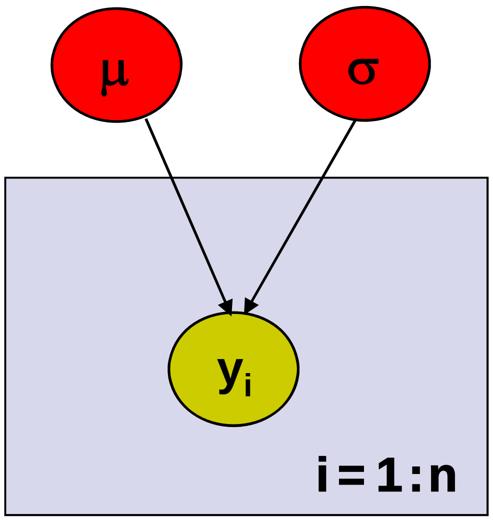
Model parameters

Data = $\{y_1,\ldots y_n\}$

Plate = rectangle in graphical model

variables within a plate are replicated
in a conditionally independent manner

Example: Gaussian Model



Generative model:

$$p(y_1,\ldots y_n \mid \mu, \sigma) = \prod p(y_i \mid \mu, \sigma)$$

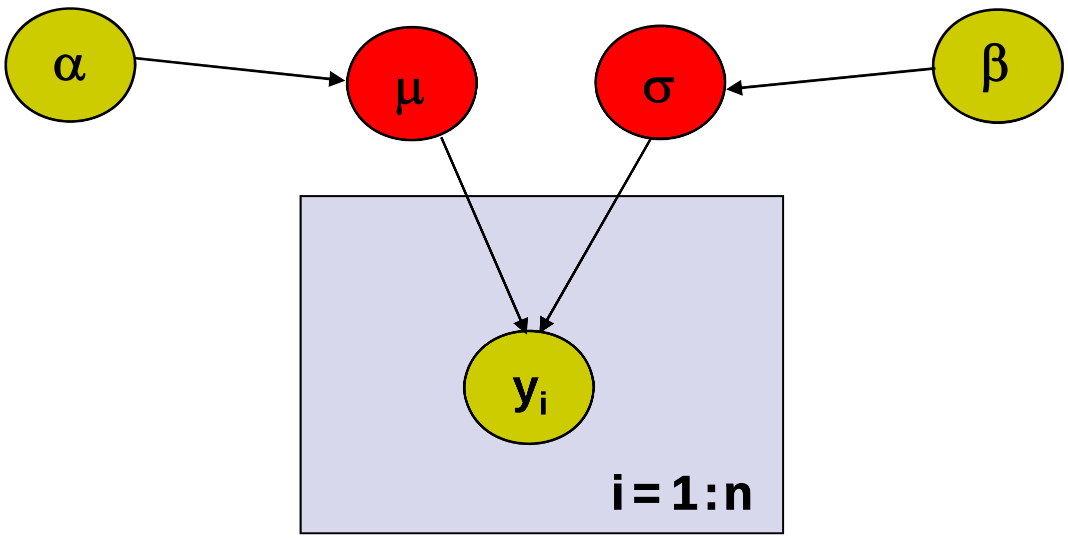
$$= \text{p(data | parameters)}$$

$$= p(D \mid \theta)$$

where $\theta = \{\mu, \sigma\}$

- **Likelihood** = p(data | parameters)
  $= p(D \mid \theta)$
  $= L (\theta)$
- **Likelihood tells us how likely the observed data are conditioned on a particular setting of the parameters**
  - Often easier to work with log $L (\theta)$

Example: Bayesian Gaussian Model

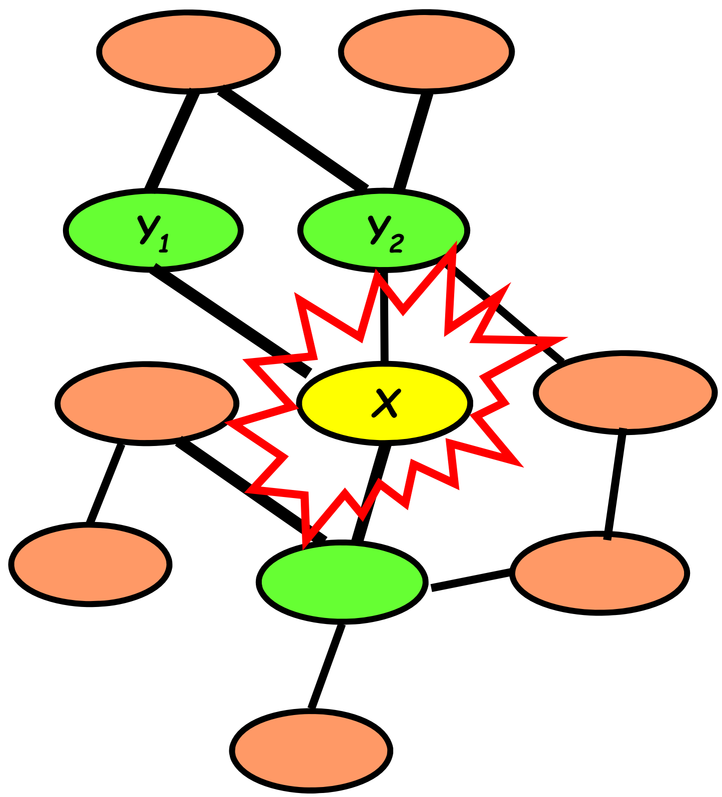


Note: priors and parameters are assumed independent here

Markov Random Fields

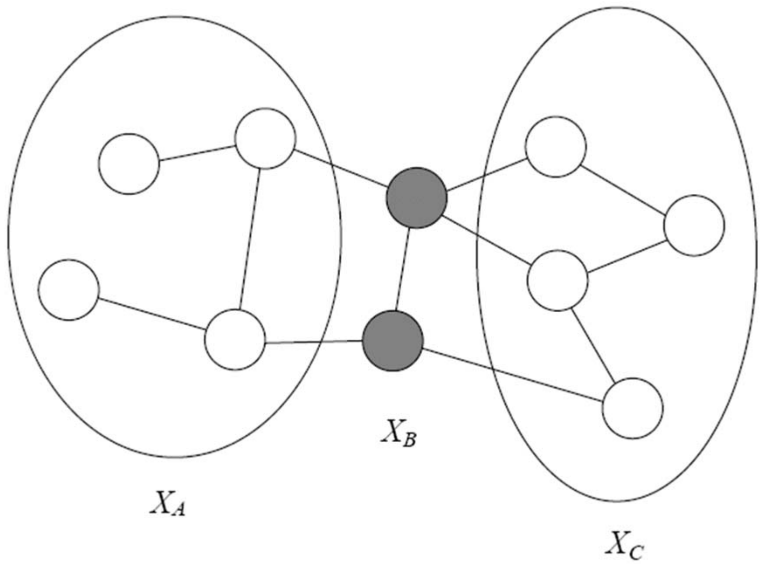
Structure: an *undirected graph*

- Meaning: a node is conditionally independent of every other node in the network given its Directed neighbors
- Local contingency functions (potentials) and the cliques in the graph completely determine the joint dist.
- Give correlations between variables, but no explicit way to generate samples



Global Markov property

- Let H be an undirected graph:



- B ***separates*** A and C if every path from a node in A to a node in C passes through a node in B: $\mathrm{sep}_H(A; C \mid B)$
- A probability distribution satisfies the ***global Markov property*** if for any disjoint A, B, C, such that B separates A and C, A is independent of C given B: $\mathrm{I}(H) = \{A \perp C \mid B) : \mathrm{sep}_H(A; C \mid B)\}$

Representation

- Defn: an undirected graphical model represents a distribution $P(X_1,\ldots,X_n)$ defined by an undirected graph H, and a set of positive ***potential functions*** y_c associated with cliques of H, s.t.

$$P(x_1,\ldots,x_n) = \frac{1}{Z}\prod_{c\in C}\psi_c(\mathbf{x}_c)$$

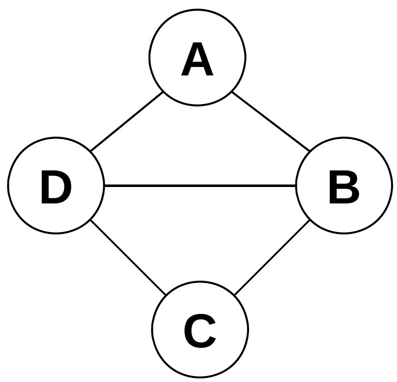
where Z is known as the partition function:

$$Z = \sum_{x_1,\ldots,x_n}\prod_{c\in C}\psi_c(\mathbf{x}_c)$$

- Also known as Markov Random Fields, Markov networks …
- The ***potential function*** can be understood as an contingency function of its arguments assigning "pre-probabilistic" score of their joint configuration.

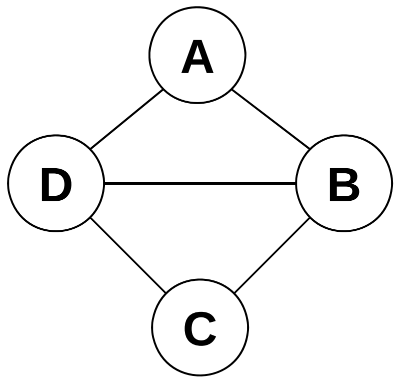
Cliques

- For *G*={*V*,*E*}, a complete subgraph (clique) is a subgraph *G'*={*V'*Í*V*,*E'*Í*E*} such that nodes in *V'* are fully interconnected
- A (maximal) clique is a complete subgraph s.t. any superset *V''*É*V'* is not complete.
- A sub-clique is a not-necessarily-maximal clique.



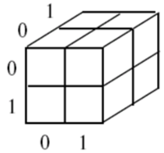
- Example:
  - max-cliques = {*A*,*B*,*D*}, {*B*,*C*,*D*},
  - sub-cliques = {*A*,*B*}, {*C*,*D*}, …→ all edges and singletons

Example UGM – using max cliques



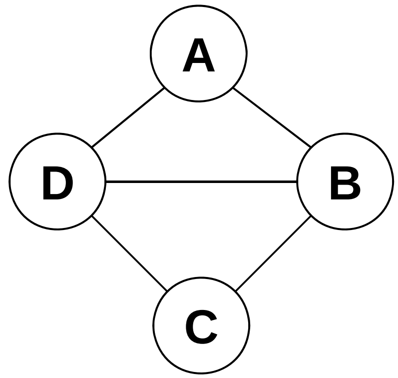
$$P(x_1, x_2, x_3, x_4) = \frac{1}{Z}\psi_c(\mathbf{x}_{124}) \times \psi_c(\mathbf{x}_{234})$$

$$Z = \sum_{x_1, x_2, x_3, x_4} \psi_c(\mathbf{x}_{124}) \times \psi_c(\mathbf{x}_{234})$$

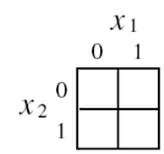


- For discrete nodes, we can represent $P(X_{1:4})$ as two 3D tables instead of one 4D table

Example UGM – using subcliques



$$P(x_1, x_2, x_3, x_4) = \frac{1}{Z} \prod_{ij} \psi_{ij}(\mathbf{x}_{ij})$$



$$= \frac{1}{Z} \psi_{12}(\mathbf{x}_{12}) \psi_{14}(\mathbf{x}_{14}) \psi_{23}(\mathbf{x}_{23}) \psi_{24}(\mathbf{x}_{24}) \psi_{34}(\mathbf{x}_{34})$$

$$Z = \sum_{x_1, x_2, x_3, x_4} \prod_{ij} \psi_{ij}(\mathbf{x}_{ij})$$

- For discrete nodes, we can represent $P(X_{1:4})$ as 5 2D tables instead of one 4D table

Exponential Form

- Constraining clique potentials to be positive could be inconvenient (e.g., the interactions between a pair of atoms can be either attractive or repulsive). We represent a clique potential $\psi_c(\mathbf{x}_c)$ in an unconstrained form using a real-value "energy" function $\phi_c(\mathbf{x}_c)$:

$$\psi_c(\mathbf{x}_c) = \exp\{-\phi_c(\mathbf{x}_c)\}$$

For convenience, we will call $\phi_c(\mathbf{x}_c)$ a potential when no confusion arises from the context.

- This gives the joint a nice additive strcuture

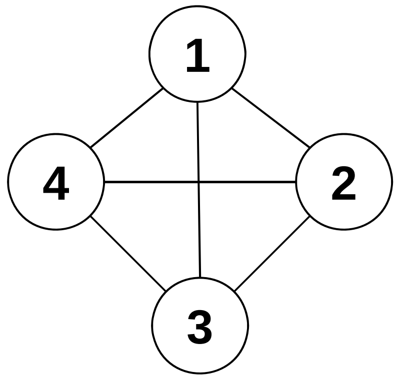
$$p(\mathbf{x}) = \frac{1}{Z}\exp\left\{-\sum_{c\in C}\phi_c(\mathbf{x}_c)\right\} = \frac{1}{Z}\exp\{-H(\mathbf{x})\}$$

where the sum in the exponent is called the "free energy":

$$H(\mathbf{x}) = \sum_{c\in C}\phi_c(\mathbf{x}_c)$$

- In physics, this is called the "Boltzmann distribution".
- In statistics, this is called a log-linear model.

Example: Boltzmann machines



- A fully connected graph with pairwise (edge) potentials on binary-valued nodes (for $x_i \in \{-1,+1\}$ or $x_i \in \{0,1\}$) is called a Boltzmann machine

$$P(x_1, x_2, x_3, x_4) = \frac{1}{Z} \exp\left\{ \sum_{ij} \phi_{ij}(x_{i,} x_j) \right\}$$

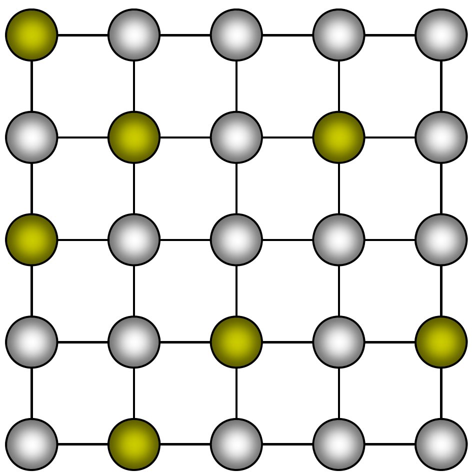
$$= \frac{1}{Z} \exp\left\{ \sum_{ij} \theta_{ij} x_i x_j + \sum_i \alpha_i x_i + C \right\}$$

- Hence the overall energy function has the form:

$$H(x) = \sum_{ij} (x_i - \mu) \Theta_{ij} (x_j - \mu) = (x - \mu)^T \Theta (x - \mu)$$

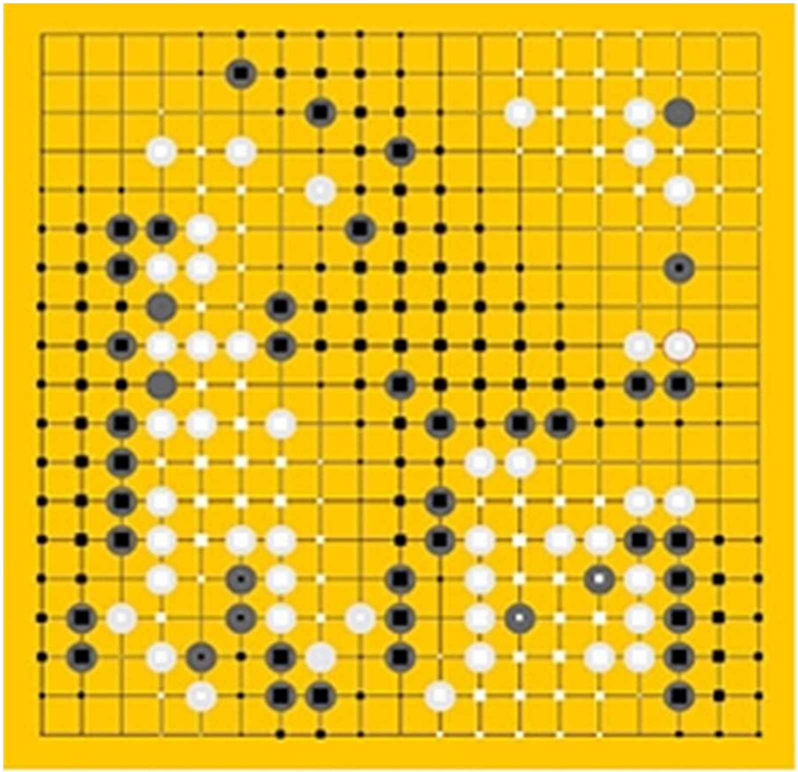
Example: Ising (spin-glass) models

- Nodes are arranged in a regular topology (often a regular packing grid) and connected only to their geometric neighbors.



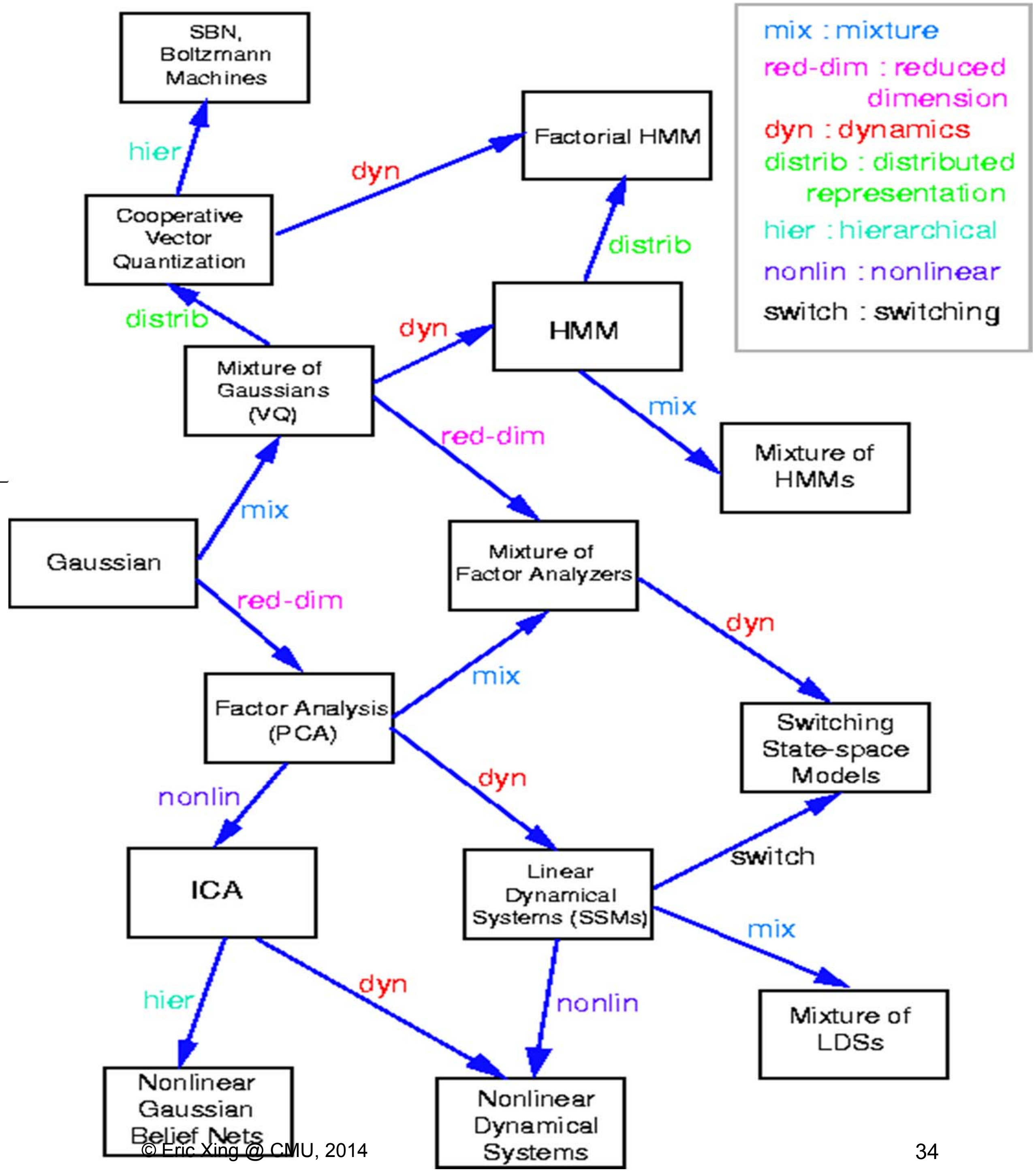
- Same as sparse Boltzmann machine, where $\theta_{ij} \neq 0$ iff i,j are neighbors.
  - e.g., nodes are pixels, potential function encourages nearby pixels to have similar intensities.
- Potts model: multi-state Ising model.

Example: Modeling Go



This is the middle position of a Go game.
Overlaid is the estimate for the probability of
becoming black or white for every intersection.
Large squares mean the probability is higher.

An (incomplete) genealogy of graphical models



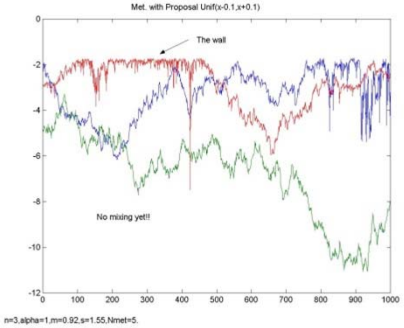
(Picture by Zoubin Ghahramani and Sam Roweis)

Advanced Introduction to Machine Learning

Markov Chain Monte Carlo

Eric Xing

Lecture 14, October 20, 2014



Reading:

Approaches to inference

- Exact inference algorithms
  - The elimination algorithm
  - Belief propagation
  - The junction tree algorithms (but will not cover in detail here)
- Approximate inference techniques
  - Variational algorithms
  - Stochastic simulation / sampling methods
  - Markov chain Monte Carlo methods

Monte Carlo methods

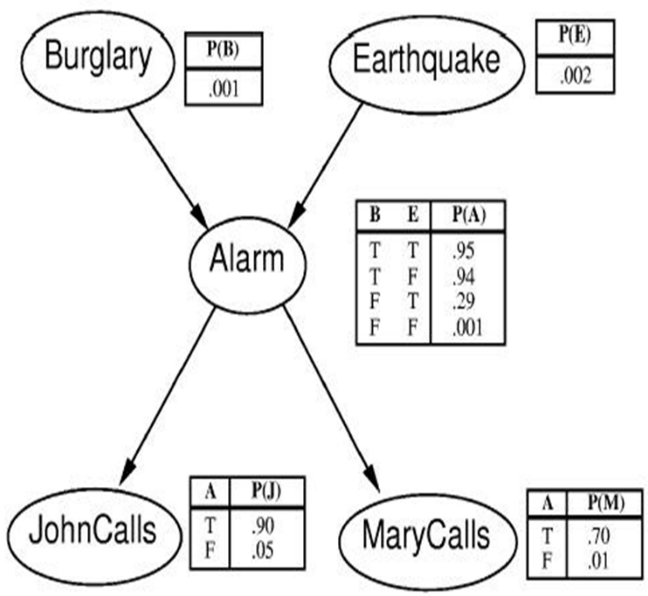
- Draw random samples from the desired distribution
- Yield a stochastic representation of a complex distribution
  - marginals and other expections can be approximated using sample-based averages

$$E[f(x)] = \frac{1}{N}\sum_{t=1}^{N} f(x^{(t)})$$

- **Asymptotically** exact and easy to apply to arbitrary models
- Challenges:
  - how to draw samples from a given dist. (not all distributions can be trivially sampled)?
  - how to make better use of the samples (not all sample are useful, or eqally useful, see an example later)?
  - how to know we've sampled enough?

Example: naive sampling

- Construct samples according to probabilities given in a BN.



Alarm example: (Choose the right sampling sequence)
1) Sampling:P(B)=<0.001, 0.999> suppose it is false, B0. Same for E0. P(A|B0, E0)=<0.001, 0.999> suppose it is false...
2) Frequency counting: In the samples right, P(J|A0)=P(J,A0)/P(A0)=<1/9, 8/9>.

E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E1	B0	A1	M1	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0

Example: naive sampling

- Construct samples according to probabilities given in a BN.

Alarm example: (Choose the right sampling sequence)

3) what if we want to compute P(J|A1) ?
we have only one sample ...
P(J|A1)=P(J,A1)/P(A1)=<0, 1>.

4) what if we want to compute P(J|B1) ?
No such sample available!
P(J|A1)=P(J,B1)/P(B1) can not be defined.

For a model with hundreds or more variables, rare events will be very hard to garner evough samples even after a long time or sampling ...

E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E1	B0	A1	M1	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0

Monte Carlo methods (cond.)

- Direct Sampling
  - We have seen it.
  - Very difficult to populate a high-dimensional state space
- Rejection Sampling
  - Create samples like direct sampling, only count samples which is consistent with given evidences.
- Likelihood weighting, ...
  - Sample variables and calculate evidence weight. Only create the samples which support the evidences.
- Markov chain Monte Carlo (MCMC)
  - Metropolis-Hasting
  - Gibbs

Markov chain Monte Carlo (MCMC)

- Construct a Markov chain whose stationary distribution is the target density $= P(X|e)$.

- Run for T samples (burn-in time) until the chain converges/mixes/reaches stationary distribution.
- Then collect M (correlated) samples x_m .
- Key issues:
  - Designing proposals so that the chain mixes rapidly.
  - Diagnosing convergence.

Markov Chains

- **Definition:**
  - Given an n-dimensional state space
  - Random vector $\mathbf{X} = (x_1,\ldots,x_n)$
  - $\mathbf{x}^{(t)} = \mathbf{x}$ at time-step t
  - $\mathbf{x}^{(t)}$ transitions to $\mathbf{x}^{(t+1)}$ with prob

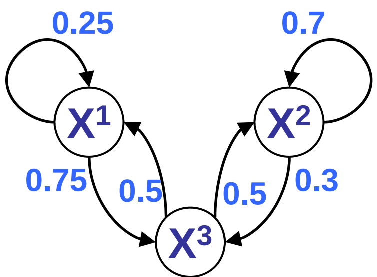
$$P(\mathbf{x}^{(t+1)} \mid \mathbf{x}^{(t)},\ldots,\mathbf{x}^{(1)}) = T(\mathbf{x}^{(t+1)} \mid \mathbf{x}^{(t)}) = T(\mathbf{x}^{(t)} \rightarrow \mathbf{x}^{(t+1)})$$

- **Homogenous**: chain determined by state $\mathbf{x}^{(0)}$, fixed *transition kernel* T (rows sum to 1)
- **Equilibrium**: $\pi(\mathbf{x})$ is a *stationary (equilibrium) distribution* if

$$\pi(\mathbf{x}') = \Sigma_{\mathbf{x}} \pi(\mathbf{x})\, T(\mathbf{x} \rightarrow \mathbf{x}').$$

i.e., is a left eigenvector of the transition matrix $\pi^T T = \pi^T T$.

$$\begin{pmatrix} 0.2 & 0.5 & 0.3 \end{pmatrix} = \begin{pmatrix} 0.2 & 0.5 & 0.3 \end{pmatrix} \begin{pmatrix} 0.25 & 0 & 0.75 \\ 0 & 0.7 & 0.3 \\ 0.5 & 0.5 & 0 \end{pmatrix}$$



Markov Chains

- An MC is ***irreducible*** if transition graph connected
- An MC is ***aperiodic*** if it is not trapped in cycles
- An MC is ***ergodic*** (regular) if you can get from state x to x' in a finite number of steps.
- **Detailed balance**: prob($\mathbf{x}^{(t)} \rightarrow \mathbf{x}^{(i-1)}$) = prob($\mathbf{x}^{(t-1)} \rightarrow \mathbf{x}^{(t)}$)

$$p(\mathbf{x}^{(t)})T(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}) = p(\mathbf{x}^{(t-1)})T(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)})$$

summing over $\mathbf{x}^{(t-1)}$

$$p(\mathbf{x}^{(t)}) = \sum_{\mathbf{x}^{(t-1)}} p(\mathbf{x}^{(t-1)})T(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)})$$

- Detailed bal → stationary dist exists

Metropolis-Hastings

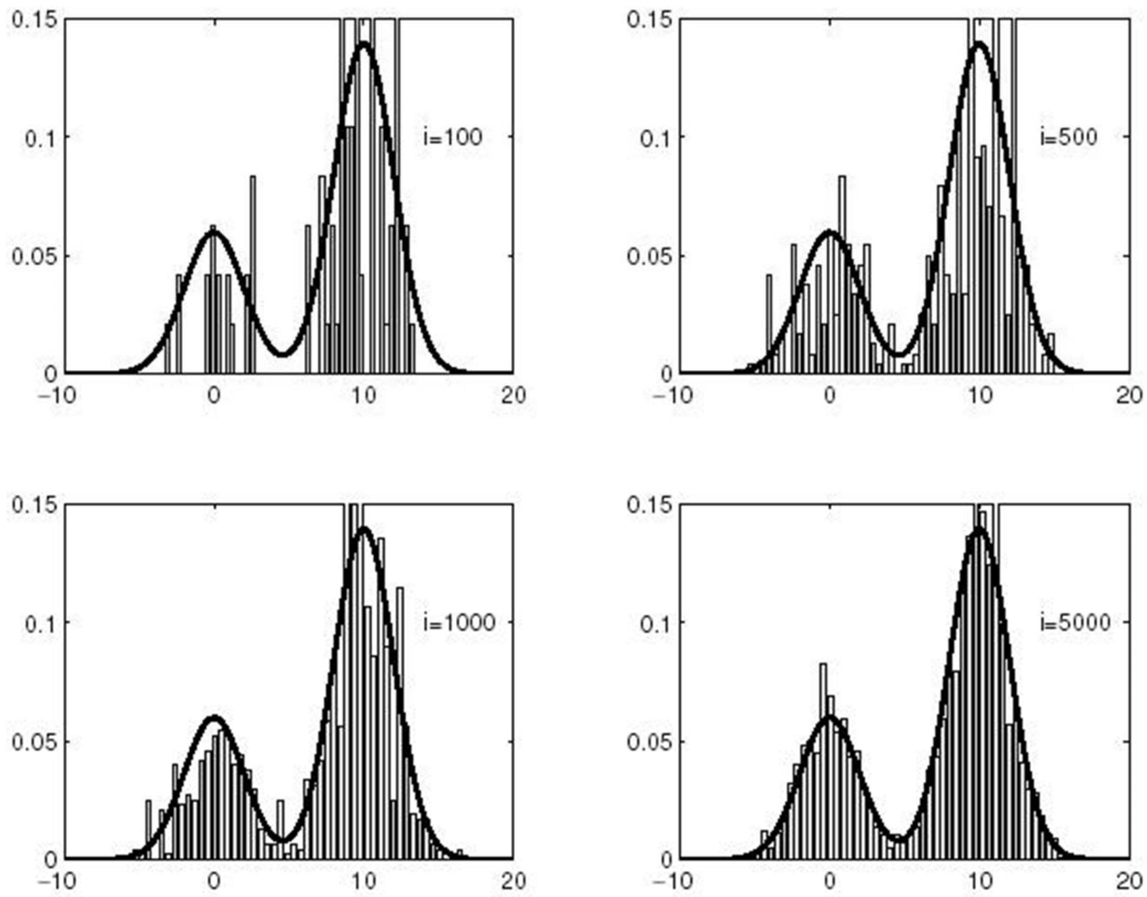
- Treat the target distribution as stationary distribution
- Sample from an easier proposal distribution, followed by an acceptance test
- This induces a transition matrix that satisfies detailed balance
  - MH proposes moves according to $Q(x'|x)$ and accepts samples with probability $A(x'|x)$.
  - The induced transition matrix is
  $$T(x \to x') = Q(x'|x)A(x'|x)$$
  - Detailed balance means
  $$\pi(x)Q(x'|x)A(x'|x) = \pi(x')Q(x|x')A(x|x')$$
  - Hence the acceptance ratio is
  $$A(x'|x) = \min\left(1, \frac{\pi(x')Q(x|x')}{\pi(x)Q(x'|x)}\right)$$

Metropolis-Hastings

1. Initialize $x^{(0)}$
2. While not mixing // burn-in
   - $x = x^{(t)}$
   - t += 1,
   - sample $u \sim \text{Unif}(0,1)$
   - sample $x^* \sim Q(x^*|x)$
     - if $u < A(x^* | x) = \min\left(1, \dfrac{\pi(x^*)Q(x \mid x^*)}{\pi(x)Q(x^* \mid x)}\right)$
       - $x^{(t)} = x^*$ // transition
     - else
       - $x^{(t)} = x$ // stay in current state
- Reset t=0, for t =1:N
  - $x(t+1)) \leftarrow$ Draw sample (x(t))

Function
Draw sample (x(t))

MCMC example

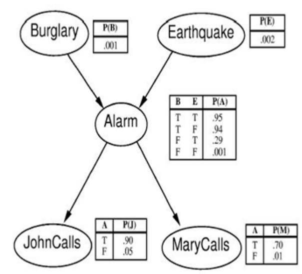


$q(x^*|x) \sim N(x^{(i)}, 100)$

$p(x) \sim 0.3 \exp(-0.2x^2) + 0.7 \exp(-0.2(x-10)^2)$

Gibbs sampling

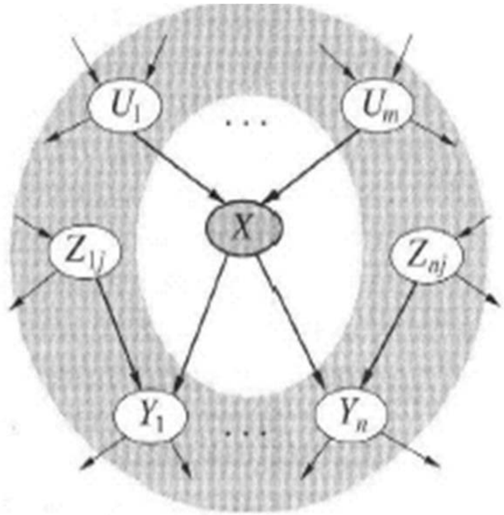
- Gibbs sampling is an MCMC algorithm that is especially appropriate for inference in graphical models.



- The procedue
  - we have variable set $\boldsymbol{X}=\{x_1, x_2, x_3,\ldots x_N\}$ for a GM
  - at each step one of the variables X_i is selected (at random or according to some fixed sequences), denote the remaining variables as $\boldsymbol{X}_{-i}$, and its current value as $\boldsymbol{x}_{-i}^{(t-1)}$
    - Using the "alarm network" as an example, say at time *t* we choose X_E, and we denote the current value assignments of the remaining variables, $\boldsymbol{X}_{-E}$, obtained from previous samples, as $x_{-E}^{(t-1)} = \left\{x_B^{(t-1)}, x_A^{(t-1)}, x_J^{(t-1)}, x_M^{(t-1)}\right\}$
  - the conditonal distribution $p(X_i| \boldsymbol{x}_{-i}^{(t-1)})$ is computed
  - a value $\boldsymbol{x}_i^{(t)}$ is sampled from this distribution
  - the sample $\boldsymbol{x}_i^{(t)}$ replaces the previous sampled value of X_i in $\boldsymbol{X}$.
    - i.e., $x^{(t)} = x_{-E}^{(t-1)} \cup x_E^{(t)}$

Markov Blanket

- **Markov Blanket in BN**
  - **A variable is independent from others, given its parents, children and children's parents (d-separation).**

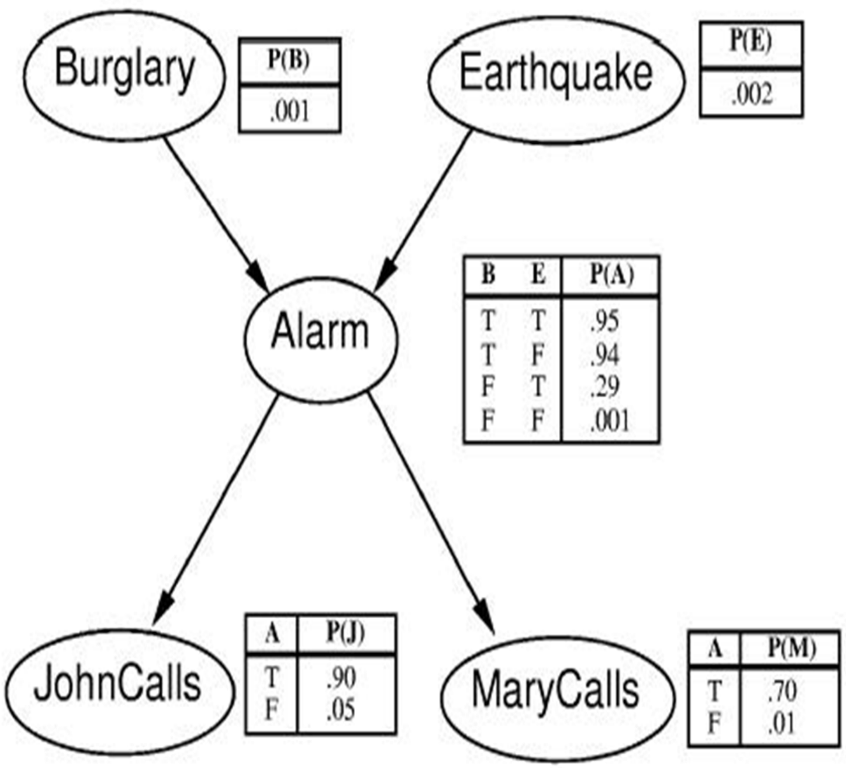


- **MB in MRF**
  - **A variable is independent all its non-neighbors, given all its direct neighbors.**

    $\Rightarrow p(X_i | X_{-i}) = p(X_i | \mathrm{MB}(X_i))$

- **Gibbs sampling**
  - **Every step, choose one variable and sample it by P(X|MB(X)) based on previous sample.**

Gibbs sampling of the alarm network

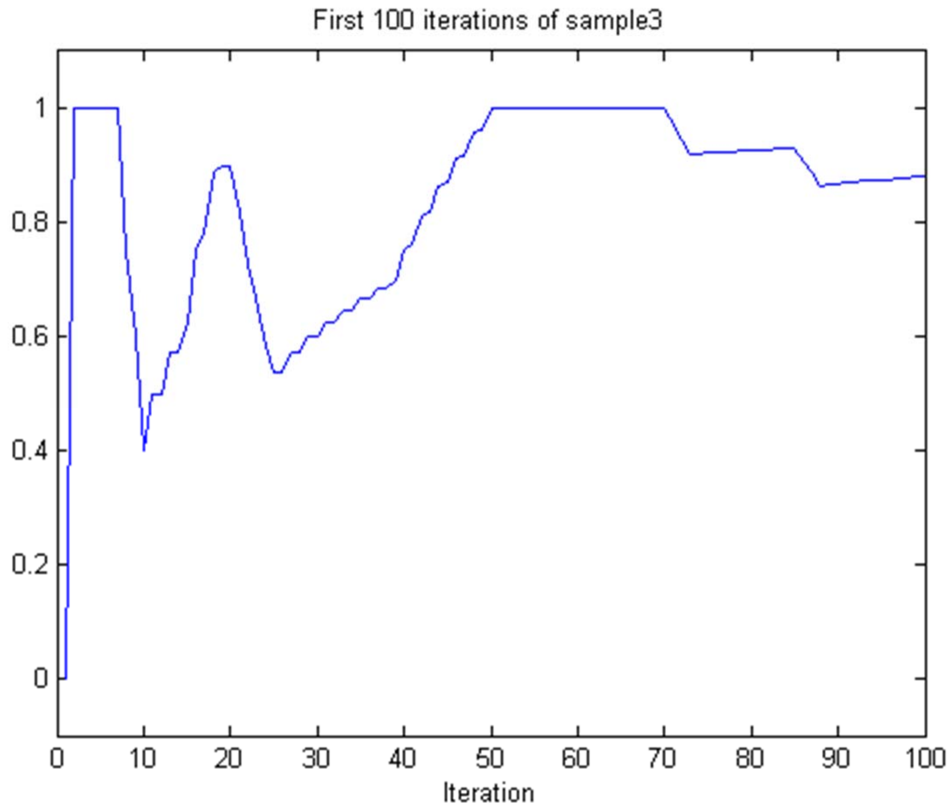


MB(A)={B, E, J, M}

MB(E)={A, B}

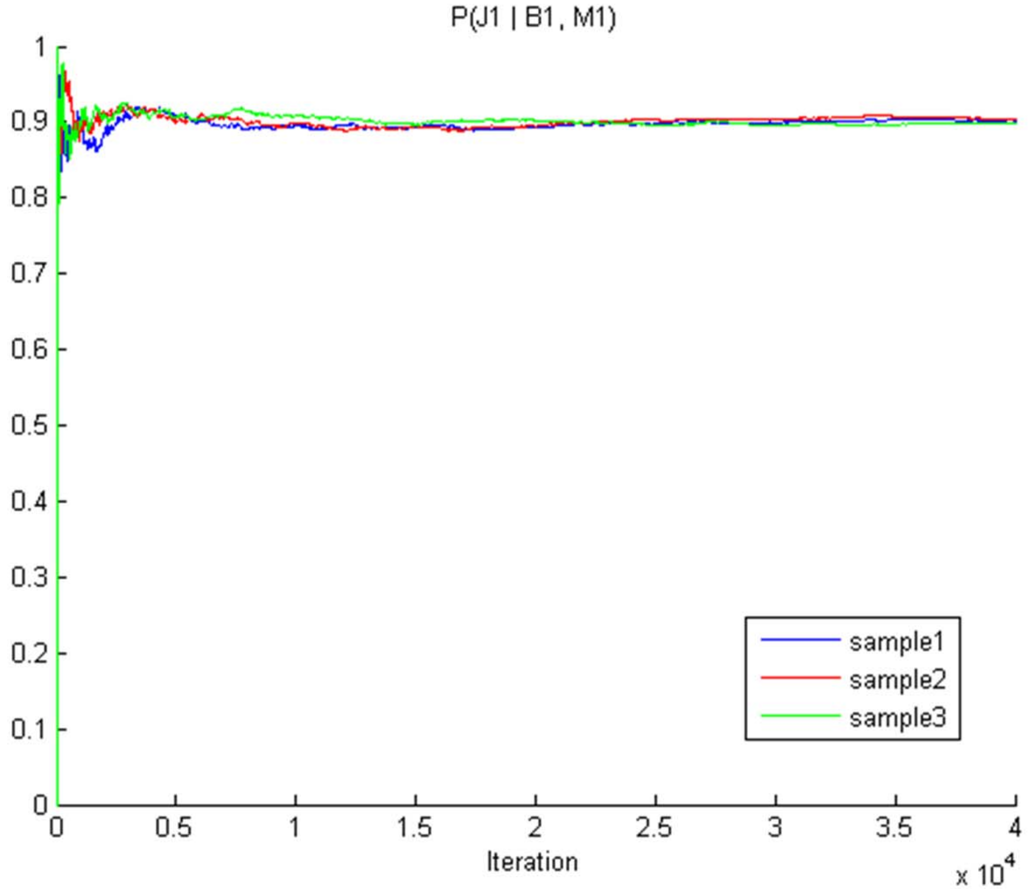
- To calculate P(J|B1,M1)
- Choose (B1,E0,A1,M1,J1) as a start
- Evidences are B1, M1, variables are A, E, J.
- Choose next variable as A
- Sample A by P(A|MB(A))=P(A|B1, E0, M1, J1) suppose to be false.
- (B1, E0, A0, M1, J1)
- Choose next random variable as E, sample E~P(E|B1,A0)
- …

Example



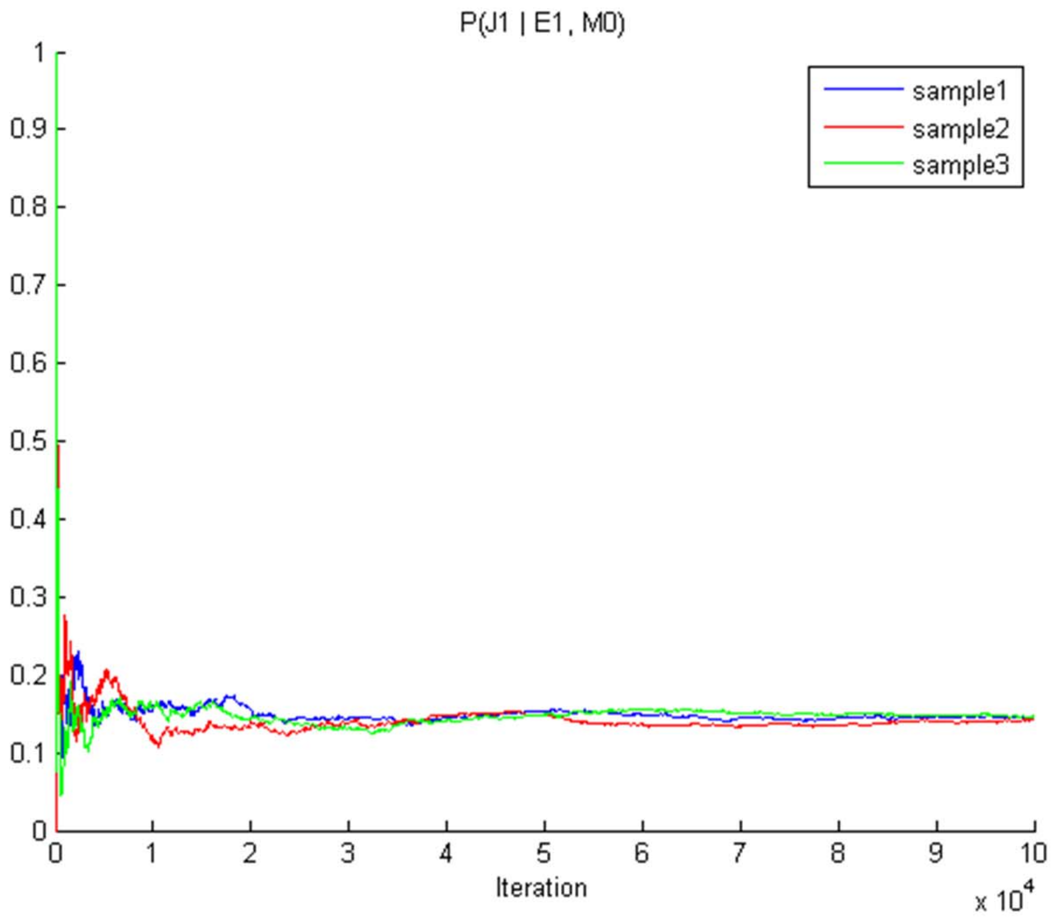
First 100 iterations of sample3
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
60
70
80
90
100
Iteration

Example:



P(J1 | B1, M1)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
3
3.5
4
Iteration
x 10^4
sample1
sample2
sample3

Example



P(J1 | E1, M0)
sample1
sample2
sample3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8
9
10
Iteration
x 10^4

Example

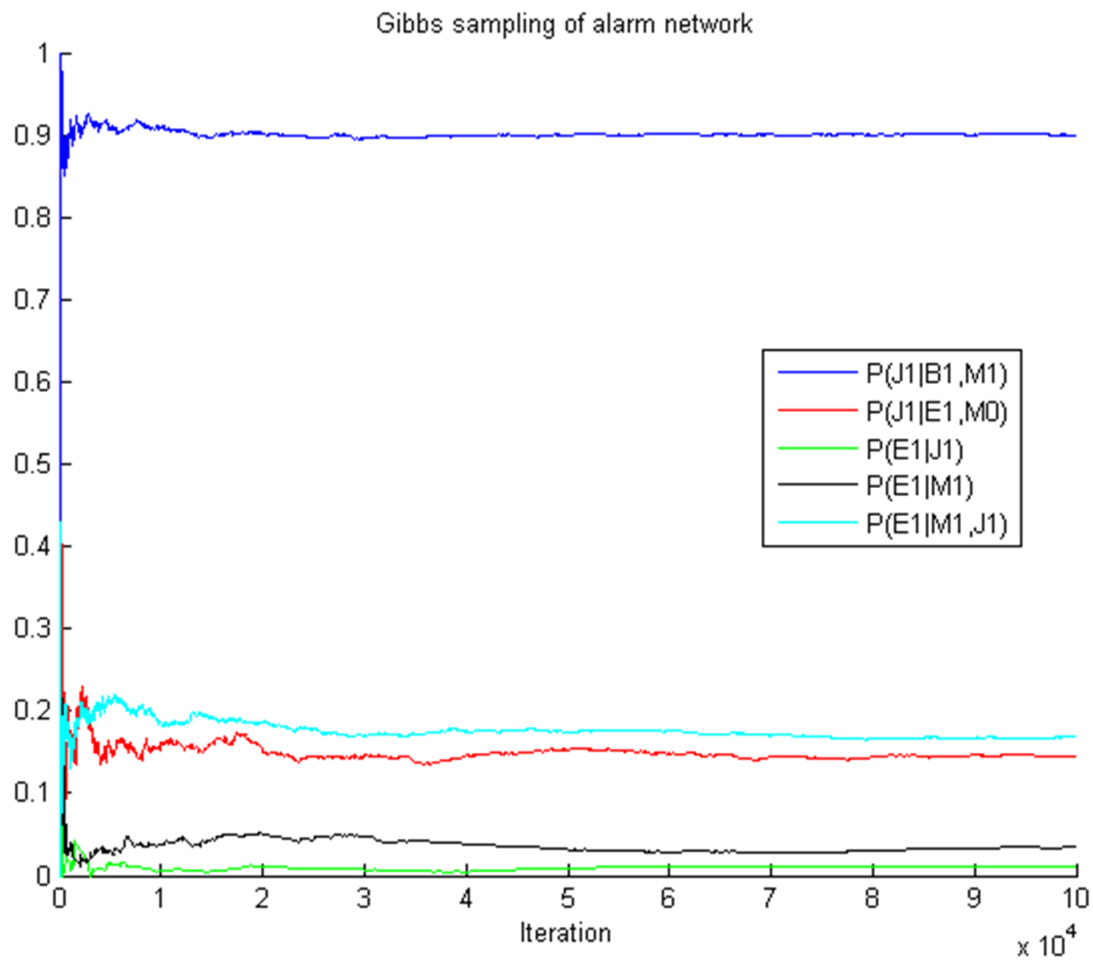
P(J1 | B1,M1) = 0.90

P(J1 | E1,M0) = 0.14

P(E1 | J1) = 0.01

P(E1 | M1) = 0.04

P(E1 | M1,J1) = 0.17



Gibbs sampling

- Gibbs sampling is a special case of MH
- The transition matrix updates each node one at a time using the following proposal:

$$Q\big((x_i, \mathbf{x}_{-i}) \rightarrow (x_i', \mathbf{x}_{-i})\big) = p(x_i' \mid \mathbf{x}_{-i})$$

- This is efficient since for two reasons
  - It leads to samples that is always accepted

$$A\big((x_i, \mathbf{x}_{-i}) \rightarrow (x_i', \mathbf{x}_{-i})\big) = \min\left(1, \frac{p(x_i', \mathbf{x}_{-i}) Q\big((x_i', \mathbf{x}_{-i}) \rightarrow (x_i, \mathbf{x}_{-i})\big)}{p(x_i, \mathbf{x}_{-i}) Q\big((x_i, \mathbf{x}_{-i}) \rightarrow (x_i', \mathbf{x}_{-i})\big)}\right)$$

$$= \min\left(1, \frac{p(x_i' \mid \mathbf{x}_{-i}) p(\mathbf{x}_{-i}) p(x_i \mid \mathbf{x}_{-i})}{p(x_i \mid \mathbf{x}_{-i}) p(\mathbf{x}_{-i}) p(x_i' \mid \mathbf{x}_{-i})}\right) = \min(1,1)$$

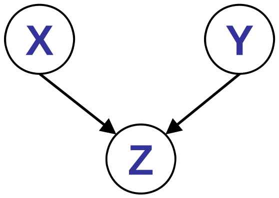
  Thus

$$T\big((x_i, \mathbf{x}_{-i}) \rightarrow (x_i', \mathbf{x}_{-i})\big) = p(x_i' \mid \mathbf{x}_{-i})$$

  - It is efficient since $p(x_i' \mid \mathbf{x}_{-i})$ only depends on the values in X_i's Markov blanket

Gibbs sampling

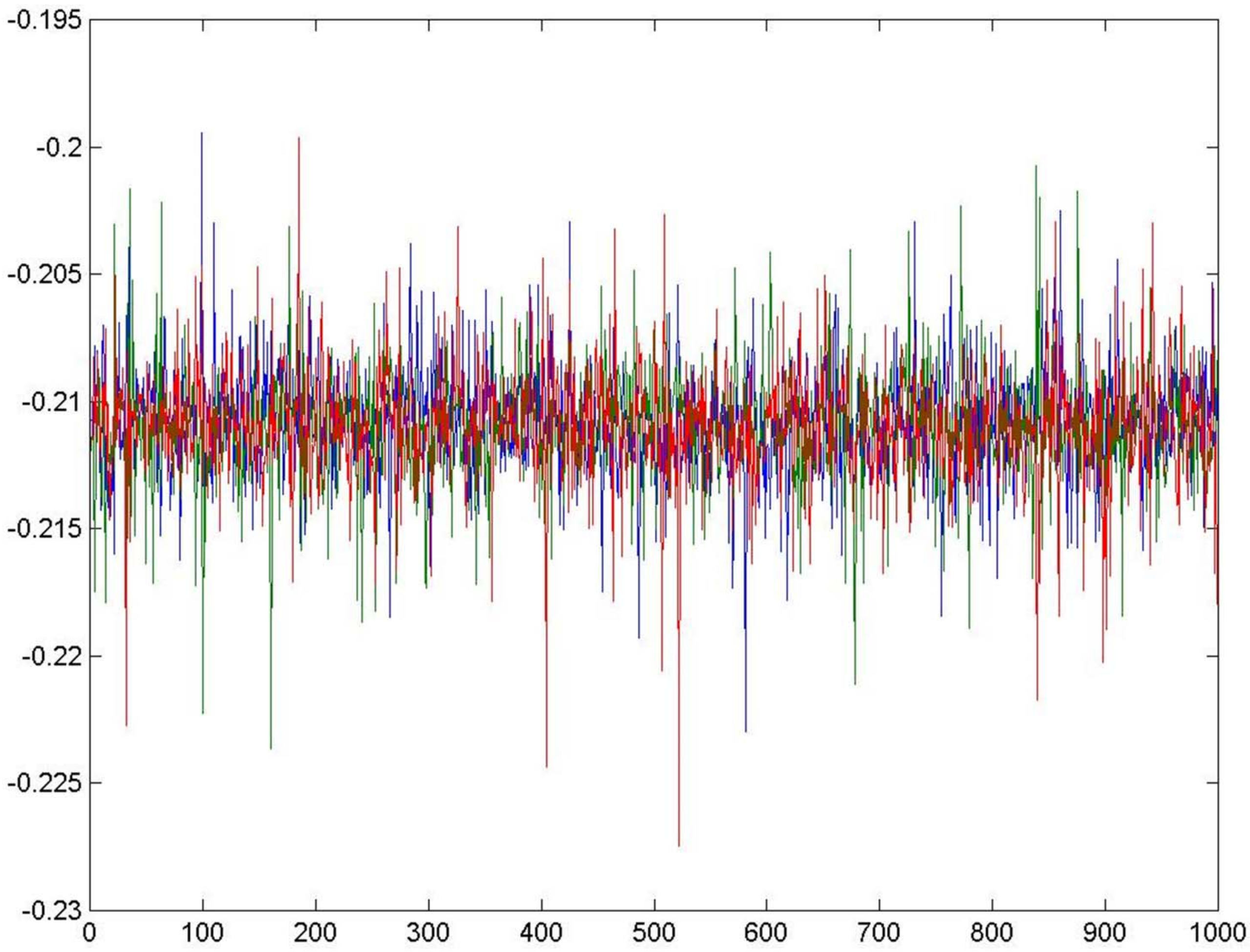
- Scheduling and ordering:
  - Sequential sweeping: in each "epoch" *t,* touch every r.v. in some order and yield an new sample, $x^{(t)}$, after every r.v. is resampled
  - Randomly pick an r.v. at each time step
- Blocking:
  - Large state space: state vector $\boldsymbol{X}$ comprised of many components (high dimension)
  - Some components can be correlated and we can sample components (i.e., subsets of r.v.,) one at a time
- Gibbs sampling can fail if there are deterministic constraint



Z is xor

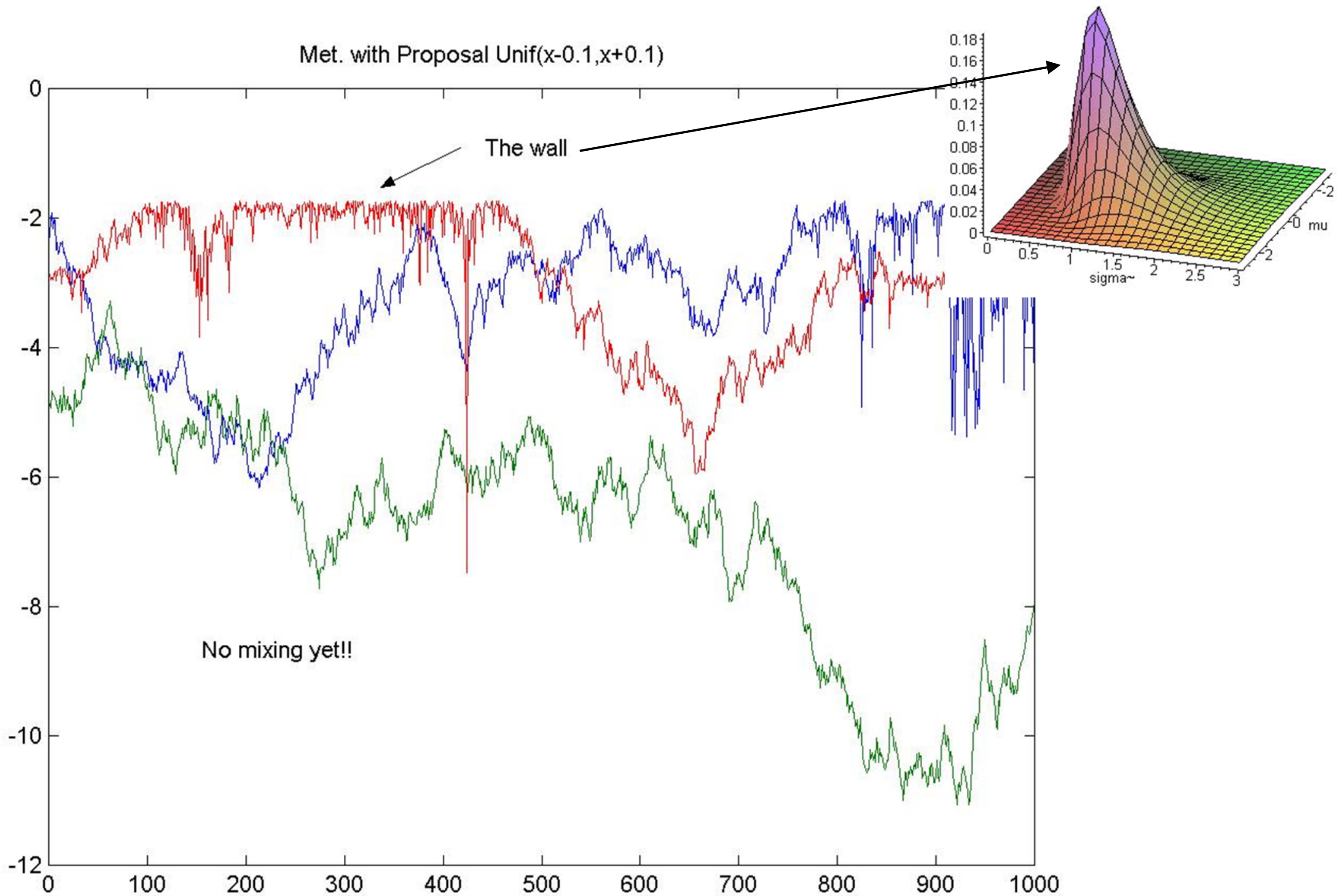
- **Suppose we observe $Z = 1$. The posterior has 2 modes: $P(X = 1, Y = 0|Z = 1)$ and $P(X = 0, Y = 1|Z = 1)$. if we start in mode 1, $P(X|Y = 0, Z = 1)$ leaves $X =$ 1, so we can't move to mode 2 (Reducible Markov chain).**
- **If all states have non-zero probability, the MC is guaranteed to be regular.**
- **Sampling blocks of variables at a time can help improve mixing.**

GOOD! Chains



-0.195
-0.2
-0.205
-0.21
-0.215
-0.22
-0.225
-0.23
0
100
200
300
400
500
600
700
800
900
1000

BAD! Chains



n=3,alpha=1,m=0.92,s=1.55,Nmet=5.

The Art of simulation

- Run several chains
- Start at over-dispersed points
- Monitor the log lik.
- Monitor the serial correlations
- Monitor acceptance ratios
- Re-parameterize (to get approx. indep.)
- Re-block (Gibbs)
- Collapse (int. over other pars.)
- Run with troubled pars. fixed at reasonable vals.

Summary

- Random walk through state space
- Can simulate multiple chains in parallel
- Much hinges on proposal distribution Q
  - Want to visit state space where $p(X)$ puts mass
  - Want $A(x^*|x)$ high in modes of $p(X)$
  - Chain mixes well
- Convergence diagnosis
  - How can we tell when burn-in is over?
  - Run multiple chains from different starting conditions, wait until they start "behaving similarly".
  - Various heuristics have been proposed.

Advanced Introduction to Machine Learning

10715, Fall 2014

Intro to Topic Models

Eric Xing

Lecture 15, October 20, 2014

Reading: Tutorial on Topic Model @ ACL12

We are inundated with data ...

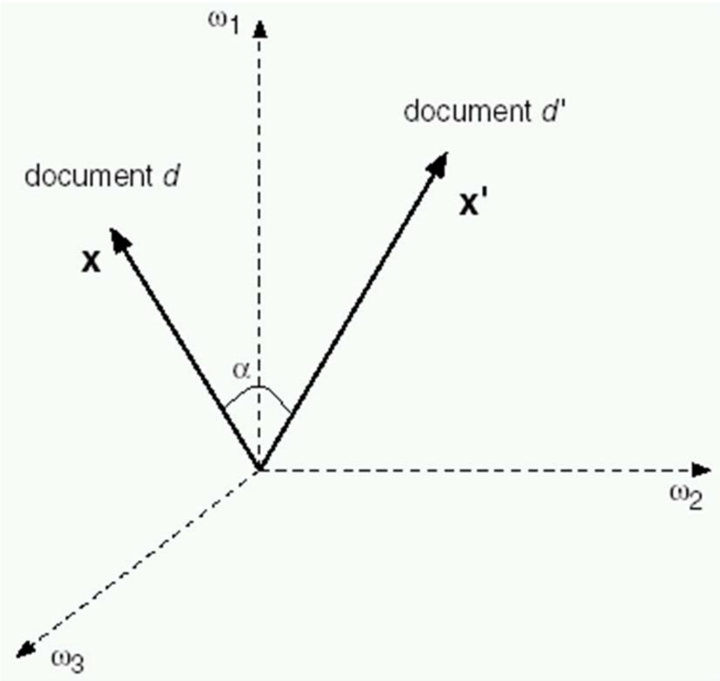
(from images.google.cn)

- Humans cannot afford to deal with (e.g., search, browse, or measure similarity) a huge number of text and media documents
- We need computers to help out …

A task:

- Say, we want to have a mapping …, so that

$\Rightarrow$



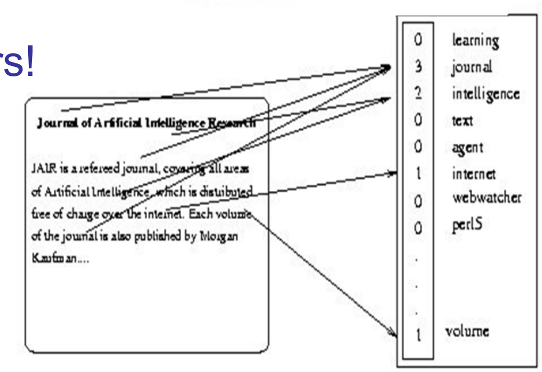
- Compare similarity
- Classify contents
- Cluster/group/categorize docs
- Distill semantics and perspectives
- ..

Representation:

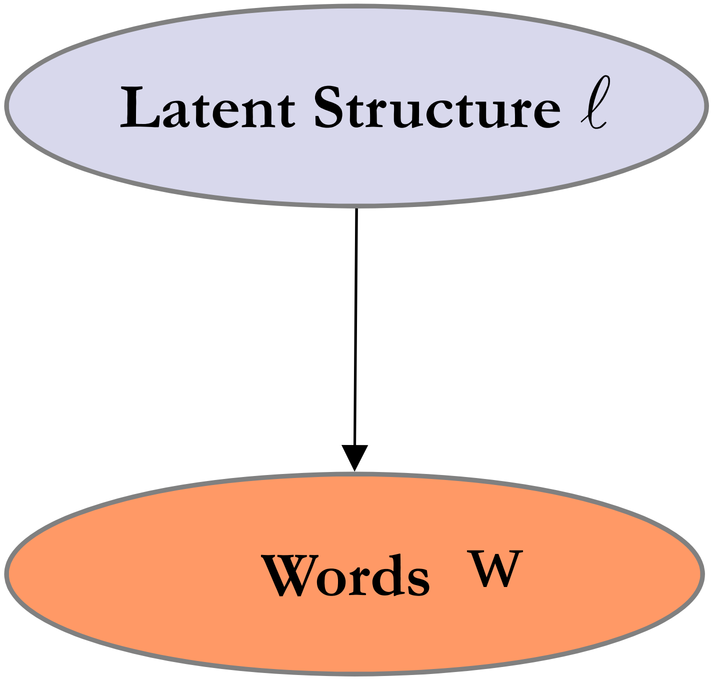
- Data: **Bag of Words Representation**

As for the Arabian and Palestinean voices that are against the current negotiations and the so-called peace process, they are not against peace per se, but rather for their well-founded predictions that Israel would NOT give an inch of the West bank (and most probably the same for Golan Heights) back to the Arabs. An 18 months of "negotiations" in Madrid, and Washington proved these predictions. Now many will jump on me saying why are you blaming israelis for no-result negotiations. I would say why would the Arabs stall the negotiations, what do they have to loose ?

- Each document is a vector in the word space
- Ignore the order of words in a document. Only count matters!
- A high-dimensional and sparse representation
  - Not efficient text processing tasks, e.g., search, document classification, or similarity measure
  - Not effective for browsing



Latent Semantic Structure in GM



Distribution over words

$$P(\mathbf{w}) = \sum_{\ell} P(\mathbf{w}, \ell)$$

Inferring latent structure

$$P(\ell \mid \mathrm{w}) = \frac{P(\mathrm{w} \mid \ell) P(\ell)}{P(\mathrm{w})}$$

How to Model Semantics?

- Q: What is it about?
- A: Mainly MT, with syntax, some learning

0.6	0.3	0.1
MT	Syntax	Learning

AdMixing Proportion

Source Target SMT Alignment Score BLEU	Parse Tree Noun Phrase Grammar CFG	likelihood EM Hidden Parameters Estimation argMax

Topics

Unigram over vocabulary

A Hierarchical Phrase-Based Model for Statistical Machine Translation

We present a statistical phrase-based Translation model that uses *hierarchical phrases*—phrases that contain sub-phrases. The model is formally a synchronous context-free grammar but is learned from a bitext without any syntactic information. Thus it can be seen as a shift to the *formal* machinery of syntax based translation systems without any *linguistic* commitment. In our experiments using BLEU as a metric, the hierarchical Phrase based model achieves a relative Improvement of 7.5% over Pharaoh, a state-of-the-art phrase-based system.

Topic Models

Why this is Useful?

- Q: What is it about?
- A: Mainly MT, with syntax, some learning

0.6	0.3	0.1
MT	Syntax	Learning

AdMixing Proportion

A Hierarchical Phrase-Based Model for Statistical Machine Translation

We present a statistical phrase-based Translation model that uses *hierarchical phrases*—phrases that contain sub-phrases. The model is formally a synchronous context-free grammar but is learned from a bitext without any syntactic information. Thus it can be seen as a shift to the *formal* machinery of syntax based translation systems without any *linguistic* commitment. In our experiments using BLEU as a metric, the hierarchical Phrase based model achieves a relative Improvement of 7.5% over Pharaoh, a state-of-the-art phrase-based system.

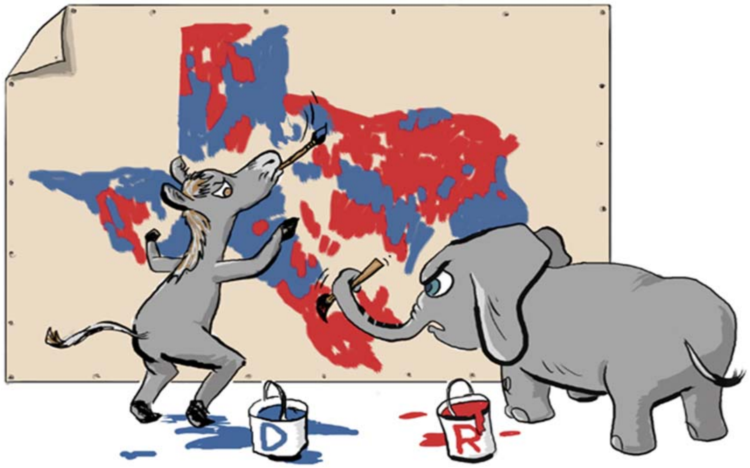
- Q: give me similar document?
  - Structured way of browsing the collection
- Other tasks
  - Dimensionality reduction
    - TF-IDF vs. topic mixing proportion
    - Classification, clustering, and more …

Words in Contexts

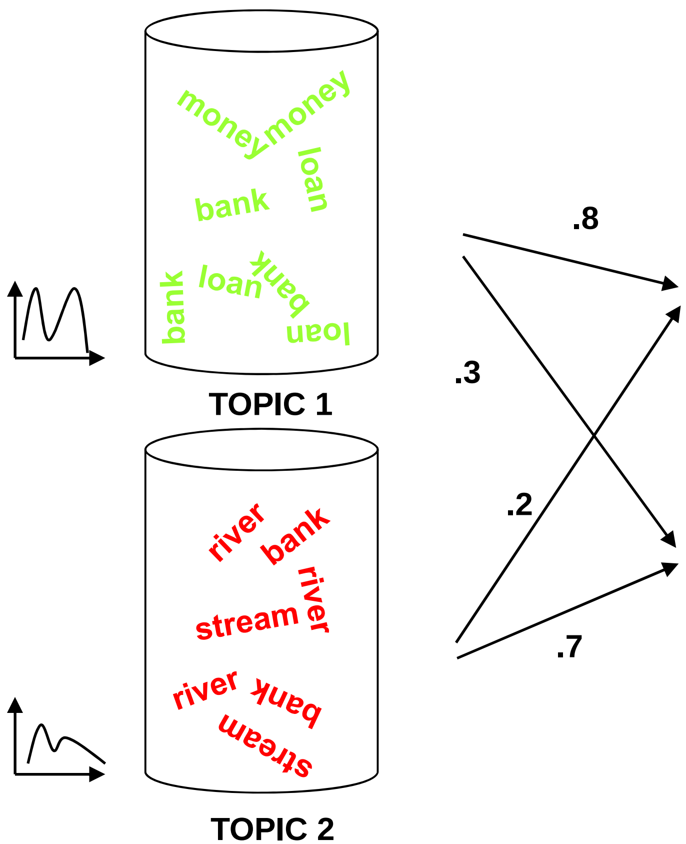
- “It was a nice **shot**. ”

Words in Contexts (con'd)

- the opposition Labor **Party** fared even worse, with a predicted 35 **seats**, seven less than last **election**.



A possible generative process of a document



DOCUMENT 1: money^1 bank^1 bank^1 loan^1 river^2 stream^2 bank^1 money^1 river^2 bank^1 money^1 bank^1 loan^1 money^1 stream^2 bank^1 money^1 bank^1 bank^1 loan^1 river^2 stream^2 bank^1 money^1 river^2 bank^1 money^1 bank^1 loan^1 bank^1 money^1 stream^2

DOCUMENT 2: river^2 stream^2 bank^2 stream^2 bank^2 money^1 loan^1 river^2 stream^2 loan^1 bank^2 river^2 bank^2 bank^1 stream^2 river^2 loan^1 bank^2 stream^2 bank^2 money^1 loan^1 river^2 stream^2 bank^2 stream^2 bank^2 money^1 river^2 stream^2 loan^1 bank^2 river^2 bank^2 money^1 bank^1 stream^2 river^2 bank^2 stream^2 bank^2 money^1

Mixture Components (distributions over elements)

admixing weight vector θ (represents all components' contributions)

Bayesian approach: use priors
Admixture weights ~ Dirichlet(α)
Mixture components ~ Dirichlet(Γ)

Topic Models = Mixed Membership Models = Admixture

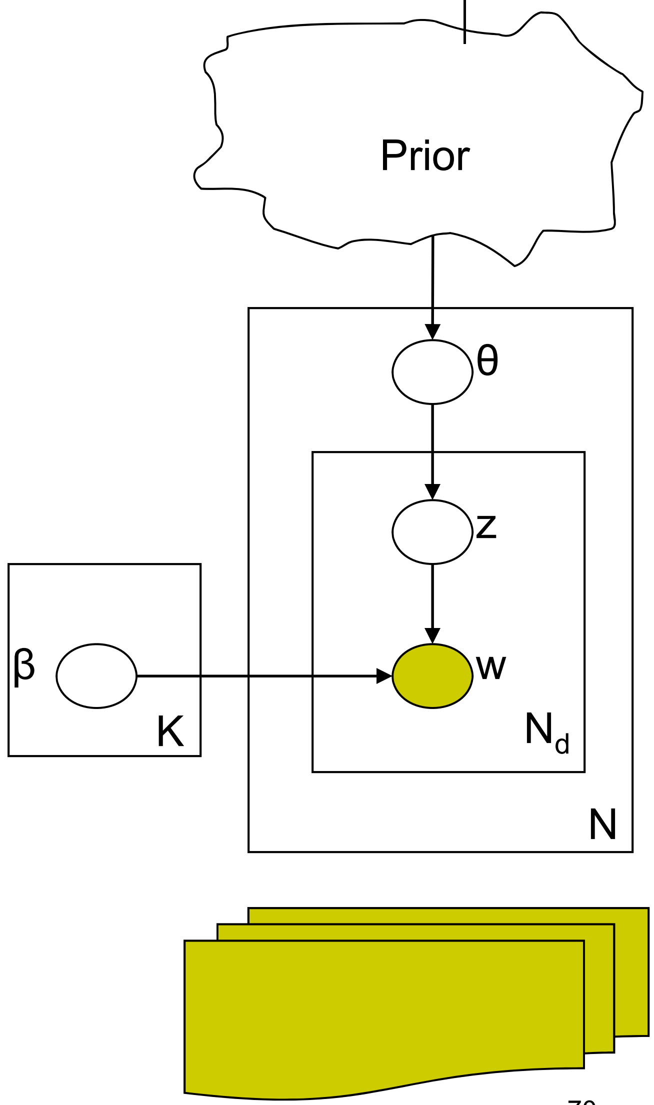
Generating a document

– *Draw* θ from the prior

For each word n

- Draw z_n from $multinomial(\theta)$
- Draw $w_n \mid z_n, \{\beta_{1:k}\}$ from $multinomial(\beta_{z_n})$

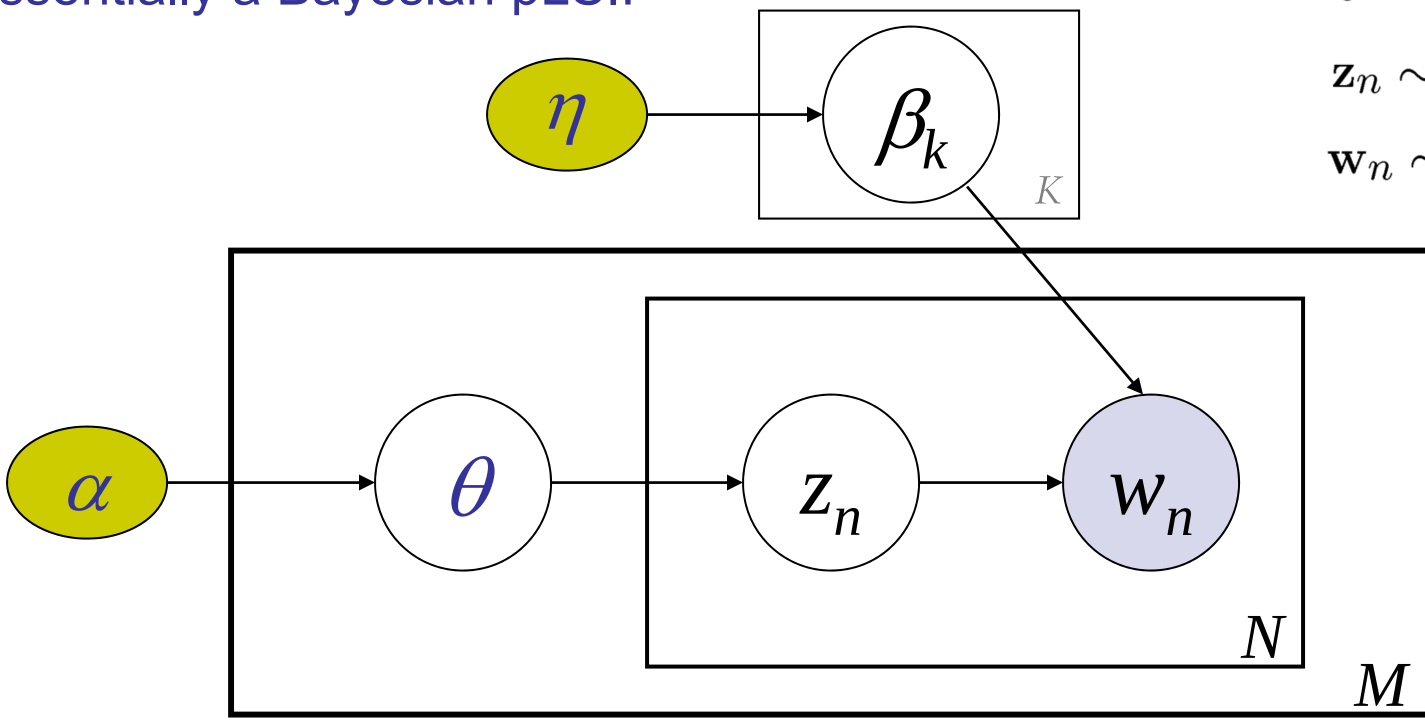
Which prior to use?



Latent Dirichlet Allocation

Blei, Ng and Jordan (2003)

Essentially a Bayesian pLSI:



$$\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})$$

$$\mathbf{z}_n \sim \mathrm{Mult}(\boldsymbol{\theta})$$

$$\mathbf{w}_n \sim p(\mathbf{w}_n | \mathbf{z}_n, \boldsymbol{\beta})$$

$$p(\mathbf{w}) = \sum_{\mathbf{z}} \int p(\theta) p(\beta) \left(\prod_{n=1}^{N} p(z_n | \theta) p(w_n | \beta_{z_n}) \right) d\theta \, d\beta$$

Outcomes from a topic model

- The "topics" β in a corpus:

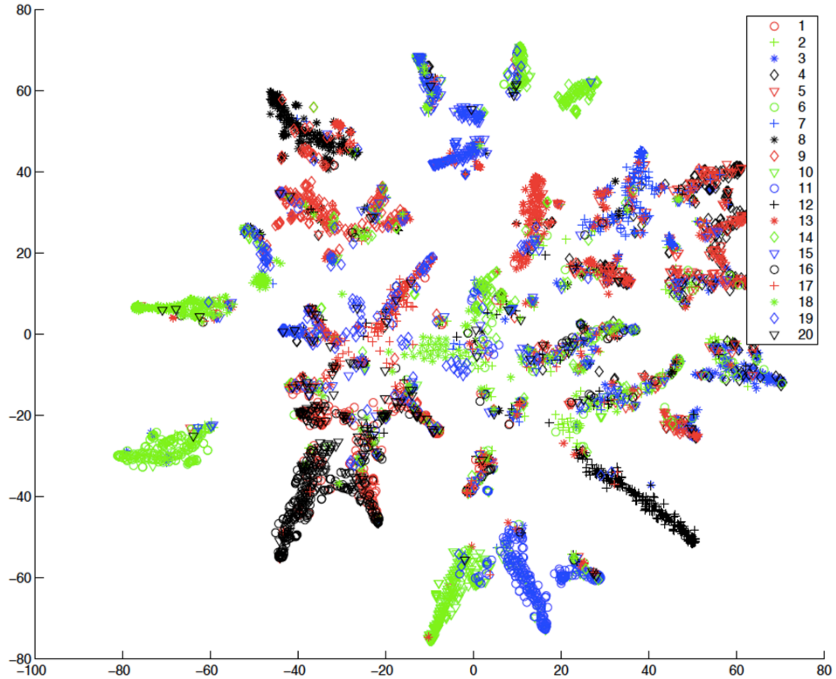
	T 59	T 104	T 31
comp.graphics	image jpeg color file gif images format bit files display	ftp pub graphics mail version tar file information send server	card monitor dos video apple windows drivers vga cards graphics
	T 30	**T 84**	**T 44**
sci.electronics	power ground wire circuit supply voltage current wiring signal cable	water energy air nuclear loop hot cold cooling heat temperature	sale price offer shipping sell interested mail condition email cd

	T 42	T 78	T 47
politics.mideast	israel israeli peace writes article arab war lebanese lebanon people	jews jewish israel israeli arab people arabs center jew nazi	armenian turkish armenians armenia turks genocide russian soviet people muslim
	T 44	**T 94**	**T 49**
misc.forsale	sale price offer shipping sell interested mail condition email cd	don mail call package writes send number ve hotel credit	drive scsi disk hard mb drives ide controller floppy system

  - There is no name for each "topic", you need to name it!
  - There is no objective measure of good/bad
  - The shown topics are the "good" ones, there are many many trivial ones, meaningless ones, redundant ones, … you need to manually prune the results
  - How many topics? …

Outcomes from a topic model

- The “topic vector” θ of each doc



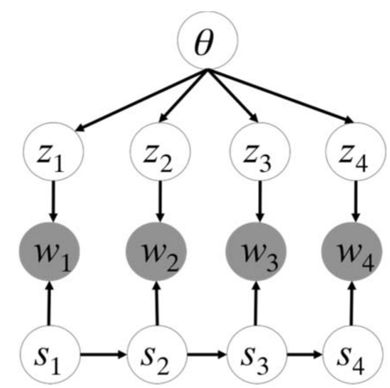
  - Create an embedding of docs in a “topic space”
  - Their no ground truth of θ to measure quality of inference
  - But on θ it is possible to define an “objective” measure of goodness, such as classification error, retrieval of similar docs, clustering, etc., of documents
  - But there is no consensus on whether these tasks bear the true value of topic models …

Outcomes from a topic model

- The per-word topic indicator z:

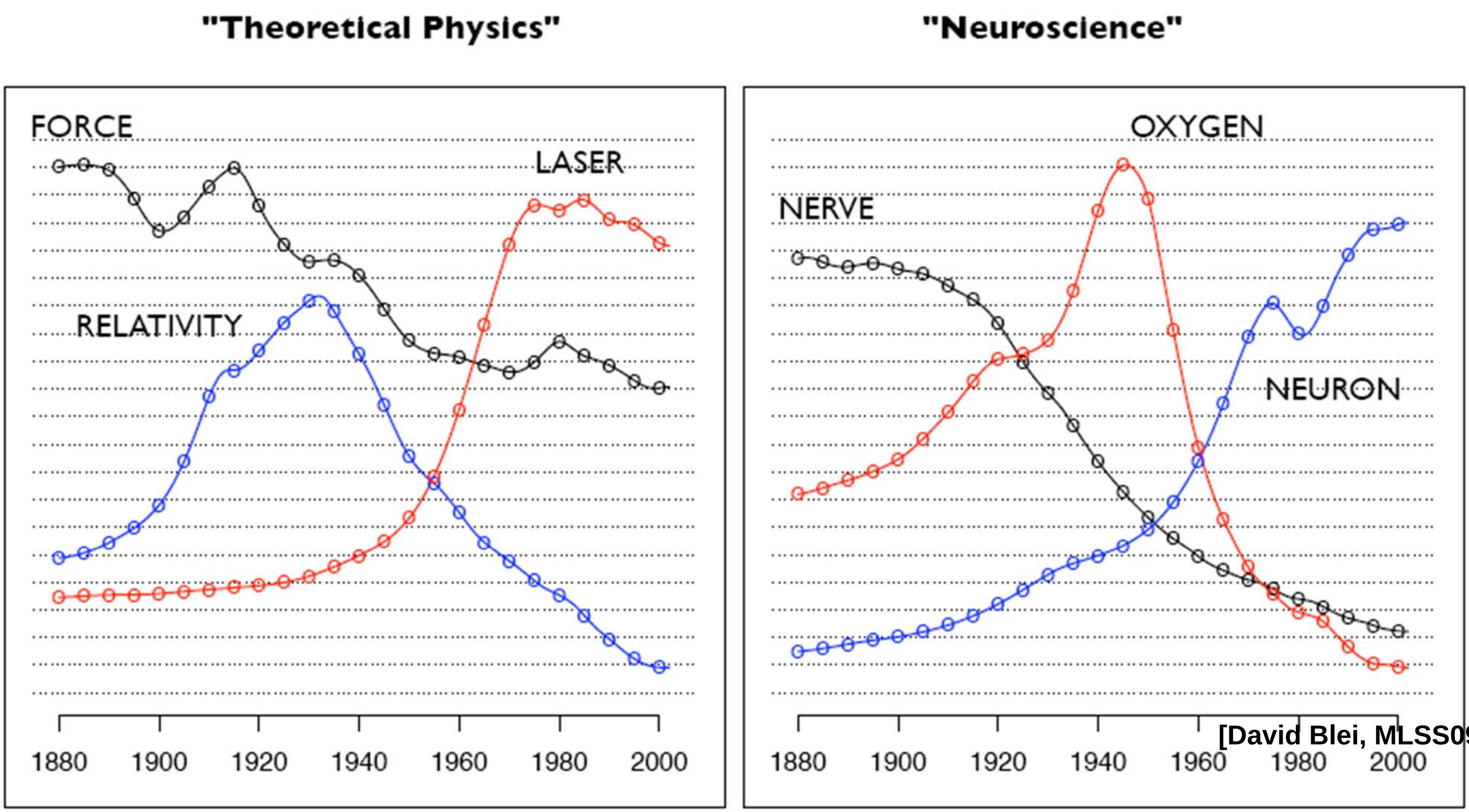
The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

- Not very useful under the bag of word representation, because of loss of ordering
- But it is possible to define simple probabilistic linguistic constraints (e.g, bi-grams) over z and get potentially interesting results [Griffiths, Steyvers, Blei, & Tenenbaum, 2004]



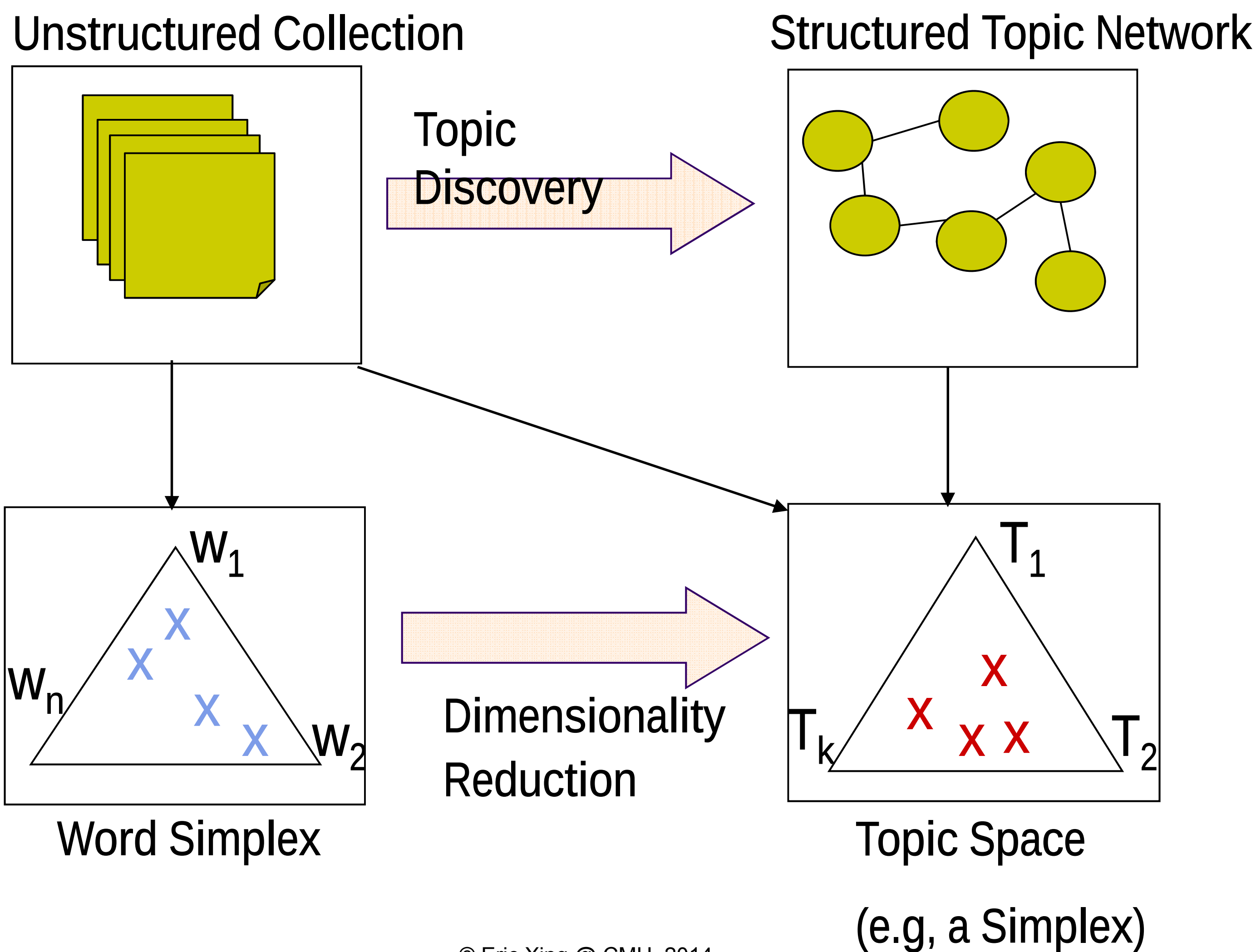
Outcomes from a topic model

- Topic change trends



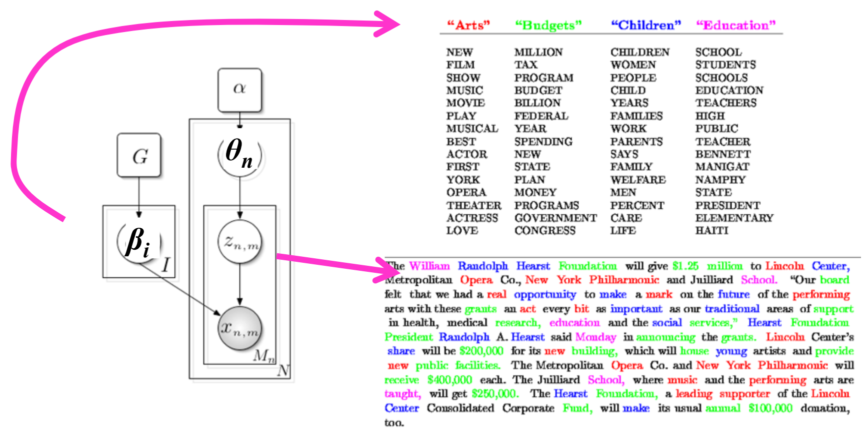
[David Blei, MLSS09]

The Big Picture



Computation on LDA

- Inference
  - Given a Document D
    - Posterior: P(Θ | μ,Σ, β ,D)
    - Evaluation: P(D| μ,Σ, β)



- Learning
  - Given a collection of documents $\{D_i\}$
    - Parameter estimation

$$\arg\max_{(\mu,\Sigma,\beta)} \sum \log\left(P\left(D_i \middle| \mu, \Sigma, \beta\right)\right)$$

Exact Bayesian inference on LDA is intractable

- A possible query:

$$p(\theta_n \mid D) = ?$$
$$p(z_{n,m} \mid D) = ?$$

- Close form solution?

$$p(\theta_n \mid D) = \frac{p(\theta_n, D)}{p(D)}$$
$$= \frac{\sum_{\{z_{n,m}\}} \int \left(\prod_n \left(\prod_m p(x_{n,m} \mid \beta_{z_n}) p(z_{n,m} \mid \theta_n) \right) p(\theta_n \mid \alpha) \right) p(\phi \mid G) d\theta_{-n} \, d\beta}{p(D)}$$

$$p(D) = \sum_{\{z_{n,m}\}} \int \cdots \int \left(\prod_n \left(\prod_m p(x_{n,m} \mid \beta_{z_n}) p(z_{n,m} \mid \theta_n) \right) p(\theta_n \mid \alpha) \right) p(\beta \mid G) d\theta_1 \cdots d\theta_N d\beta$$

- Sum in the denominator over T^n terms, and integrate over n k-dimensional topic vectors

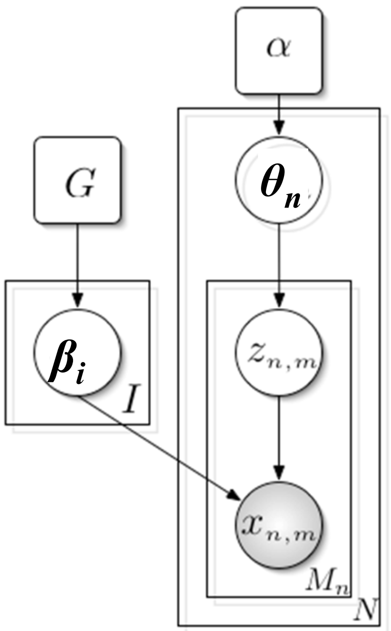
Approximate Inference

- Variational Inference
  - Mean field approximation (Blei et al)
  - Expectation propagation (Minka et al)
  - Variational 2^{nd}-order Taylor approximation (Ahmed and Xing)

- Markov Chain Monte Carlo
  - Gibbs sampling (Griffiths et al)

Collapsed Gibbs sampling
(Tom Griffiths & Mark Steyvers)

- Collapsed Gibbs sampling
  - Integrate out θ



For variables $\mathbf{z} = z_1, z_2, \ldots, z_n$

Draw $z_i^{(t+1)}$ from $P(z_i|\mathbf{z}_{-i}, \boldsymbol{w})$

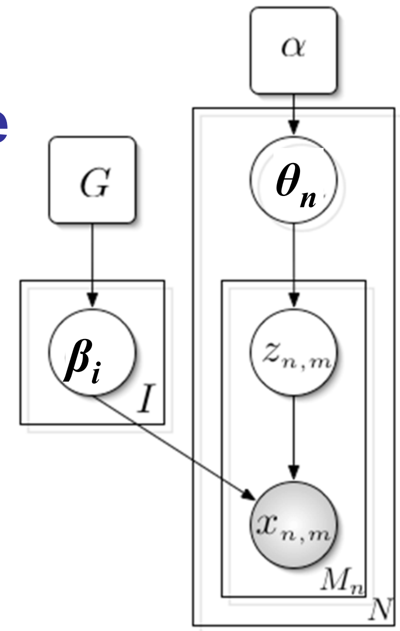
$\mathbf{z}_{-i} = z_1^{(t+1)}, z_2^{(t+1)}, \ldots, z_{i-1}^{(t+1)}, z_{i+1}^{(t)}, \ldots, z_n^{(t)}$

$$\{z^{(1)}, z^{(2)}, \ldots, z^{(T)}\}$$

$$\theta = \frac{1}{T}\sum_t z^{(t)}$$

Gibbs sampling

- Need full conditional distributions for variable
- Since we only sample z we need



$$P(z_i = j|\mathbf{z}_{-i}, \mathbf{w}) \propto P(w_i|z_i = j, \mathbf{z}_{-i}, \mathbf{w}_{-i})P(z_i = j|\mathbf{z}_{-i})$$

$$= \frac{n^{(w_i)}_{-i,j} + G}{n^{(\cdot)}_{-i,j} + WG} \frac{n^{(d_i)}_{-i,j} + \alpha}{n^{(d_i)}_{-i,\cdot} + T\alpha}$$

$n_j^{(w)}$ **number of times word *w* assigned to topic *j***

$n_j^{(d)}$ **number of times topic *j* used in document *d***

Gibbs sampling

i	w_i	d_i	iteration 1 z_i
1	MATHEMATICS	1	2
2	KNOWLEDGE	1	2
3	RESEARCH	1	1
4	WORK	1	2
5	MATHEMATICS	1	1
6	RESEARCH	1	2
7	WORK	1	2
8	SCIENTIFIC	1	1
9	MATHEMATICS	1	2
10	WORK	1	1
11	SCIENTIFIC	2	1
12	KNOWLEDGE	2	1
.	.	.	.
.	.	.	.
.	.	.	.
50	JOY	5	2

Gibbs sampling

			iteration 1	2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	?
2	KNOWLEDGE	1	2	
3	RESEARCH	1	1	
4	WORK	1	2	
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

Gibbs sampling

			iteration 1	2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	?
2	KNOWLEDGE	1	2	
3	RESEARCH	1	1	
4	WORK	1	2	
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \boldsymbol{G}}{n_{-i,j}^{(\cdot)} + W\boldsymbol{G}} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Gibbs sampling

			iteration 1	2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	?
2	KNOWLEDGE	1	2	
3	RESEARCH	1	1	
4	WORK	1	2	
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + G}{n_{-i,j}^{(\cdot)} + WG} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Gibbs sampling

			iteration 1	2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	2
2	KNOWLEDGE	1	2	?
3	RESEARCH	1	1	
4	WORK	1	2	
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + G}{n_{-i,j}^{(\cdot)} + WG} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Gibbs sampling

			iteration	
			1	2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	2
2	KNOWLEDGE	1	2	1
3	RESEARCH	1	1	?
4	WORK	1	2	
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + G}{n_{-i,j}^{(\cdot)} + WG} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Gibbs sampling

			iteration 1	2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	2
2	KNOWLEDGE	1	2	1
3	RESEARCH	1	1	1
4	WORK	1	2	?
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + G}{n_{-i,j}^{(\cdot)} + WG} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Gibbs sampling

			iteration 1	2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	2
2	KNOWLEDGE	1	2	1
3	RESEARCH	1	1	1
4	WORK	1	2	2
5	MATHEMATICS	1	1	?
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + G}{n_{-i,j}^{(\cdot)} + WG} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Gibbs sampling

			iteration 1	2	…	1000
i	w_i	d_i	z_i	z_i		z_i
1	MATHEMATICS	1	2	2		2
2	KNOWLEDGE	1	2	1		2
3	RESEARCH	1	1	1		2
4	WORK	1	2	2		1
5	MATHEMATICS	1	1	2		2
6	RESEARCH	1	2	2		2
7	WORK	1	2	2		2
8	SCIENTIFIC	1	1	1	…	1
9	MATHEMATICS	1	2	2		2
10	WORK	1	1	2		2
11	SCIENTIFIC	2	1	1		2
12	KNOWLEDGE	2	1	2		2
.	.	.	.	.		.
.	.	.	.	.		.
.	.	.	.	.		.
50	JOY	5	2	1		1

$$\theta = \frac{1}{T}\sum_t z^{(t)}$$

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + G}{n_{-i,j}^{(\cdot)} + WG} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Learning a TM

- Maximum likelihood estimation:

$$\{\beta_1, \beta_2, \ldots, \beta_K\}, \alpha = \underset{(\alpha,\beta)}{\arg\max} \sum \log\left(P\left(D_i \middle| \alpha, \beta\right)\right)$$

- Need statistics on topic-specific word assignment (due to z), topic vector distribution (due to θ), etc.
  - E.g,, this is the formula for topic k:

$$\beta_k = \frac{1}{\sum_d N_d} \sum_{d=1}^{D} \sum_{d_n=1}^{N_d} \delta(z_{d,d_n}, k) w_{d,d_n}$$

- These are hidden variables, therefore need an EM algorithm (also known as data augmentation, or DA, in Monte Carlo paradigm)

- This is a “reduce” step in parallel implementation

Conclusion

- GM-based topic models are cool
  - Flexible
  - Modular
  - Interactive
- There are many ways of implementing topic models
  - unsupervised
  - supervised
- Efficient Inference/learning algorithms
  - GMF, with Laplace approx. for non-conjugate dist.
  - MCMC
- Many applications
  - …
  - Word-sense disambiguation
  - Image understanding
  - Network inference