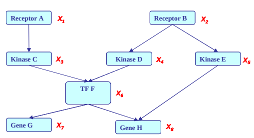
School of Computer Science
Carnegie Mellon

Towards Complex Graphical Models and Approximate Inference

Probabilistic Graphical Models (10-708)

Lecture 15, Nov 5, 2007

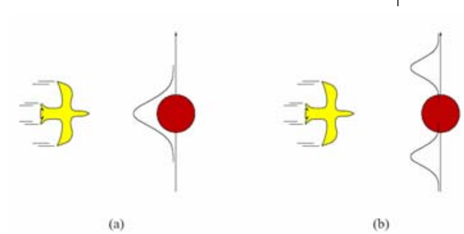


Eric Xing

Reading: posted papers

The need for complex dynamic models

- Complex dynamic systems:
  - Non-linearity
  - Non-Gaussianity
  - Multi-modality
  - ...



- Limitation of LDS

$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + K_{t+1}(\mathrm{y}_{t+1} - C\hat{\mathrm{x}}_{t+1|t})$$

$$P_{t+1|t+1} = P_{t+1|t} - KCP_{t+1|t}$$

  - defines only linearity evolving, unimodal, and Gaussian belief states
    - A Kalman filter will predict the location of the bird using a single Gaussian centered on the obstacle.
    - A more realistic model allows for the bird's evasive action, predicting that i side or the other.

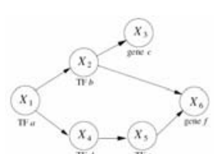
Representing complex dynamic processes

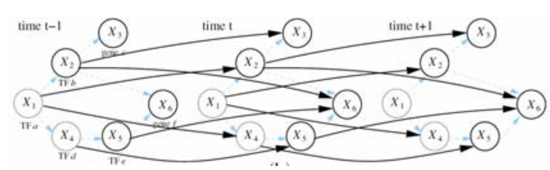
- The problem with HMMs
  - Suppose we want to track the state (e.g., the position) of D objects in an image sequence.
  - Let each object be in K possible states.
  - Then $X_t = (X_t^{(1)}, \ldots, X_t^{(D)})$ can have K^D possible values.

⇒ Inference takes time and space.

⇒ $P(X_t|X_{t-1})$ need parameters to specify.

Dynamic Bayesian Network





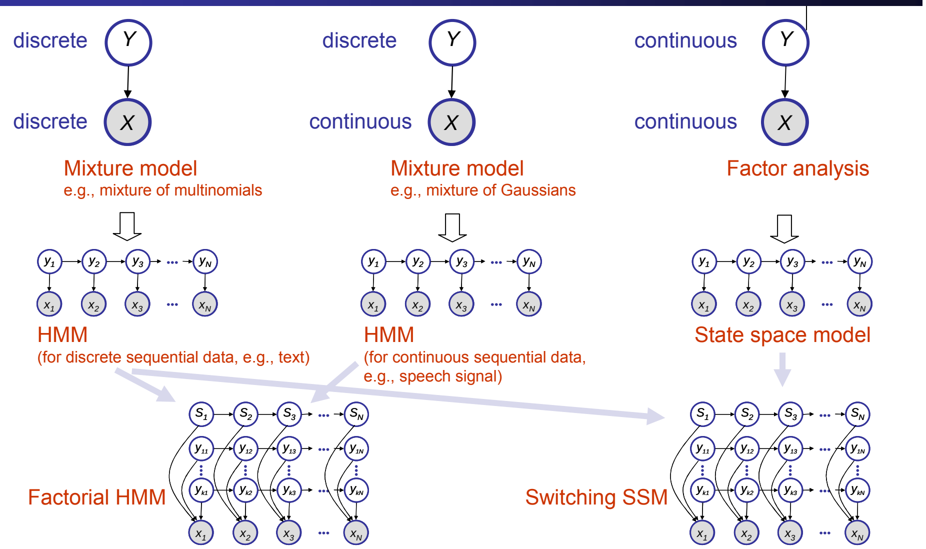
- A DBN represents the state of the world at time *t* using a set of random variables, $X_t^{(1)}, \ldots, X_t^{(D)}$ (factored/ distributed representation).
- A DBN represents $P(X_t|X_{t-1})$ in a compact way using a parameterized graph.

⇒ A DBN may have exponentially fewer parameters than its corresponding HMM.
⇒ Inference in a DBN may be exponentially faster than in the corresponding HMM.

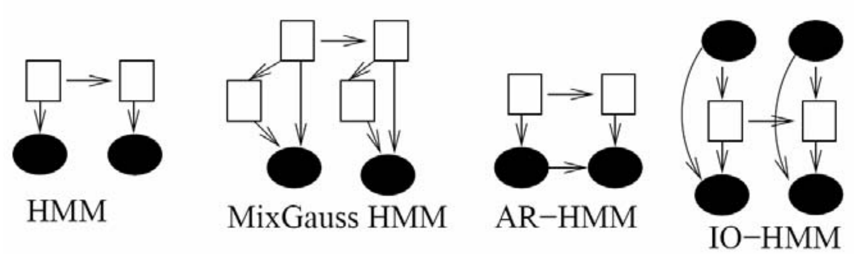
DBNs are a kind of graphical model

- In a graphical model, nodes represent random variables, and (lack of) arcs represents conditional independencies.
- DBNs are Bayes nets for dynamic processes.
- Informally, an arc from X_t^i to X_{t+1}^j means X_i "causes" X_j.
- Can "resolve" cycles in a "static" BN

A road map to complex dynamic models



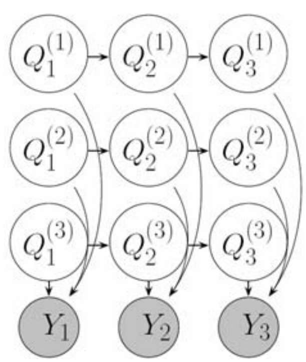
HMM variants represented as DBNs



- The same code (standard forward-backward, viterbi, and Baum-Welsh) can do inference and learning in all of these models.

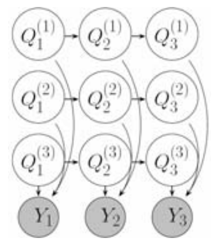
Factorial HMM

- The belief state at each time is $X_t = \{Q_t^{(1)}, \ldots, Q_t^{(k)}\}$ and in the most general case has a state space $O(d^k)$ for k d-nary chains
- The common observed child Y_t couples all the parents (explaining away).
- But the parameterization cost for fHMM is $O(kd^2)$ for k chain-specific transition models $p(Q_t^{(i)} | Q_{t-1}^{(i)})$ rather than $O(d^{2k})$ for $p(X_t | X_{t-1})$



Factorial HMMs vs HMMs

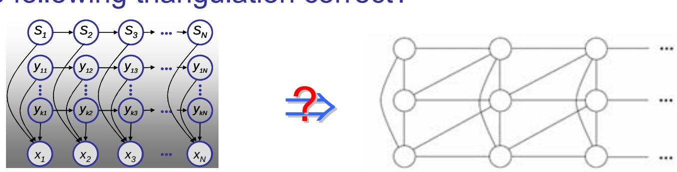
- Let us compare a factorial HMM with D chains, each with K values, to its *equivalent* HMM.



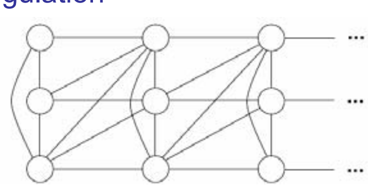
- Num. parameters to specify $p(X_t \mid X_{t-1})$
  - HMM:
  - fHMM:
- Computational complexity of exact inference:
  - HMM
  - fHMM:

Triangulating fHMM

- Is the following triangulation correct?

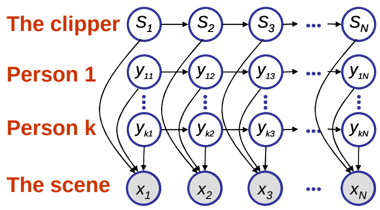


- Here is a triangulation



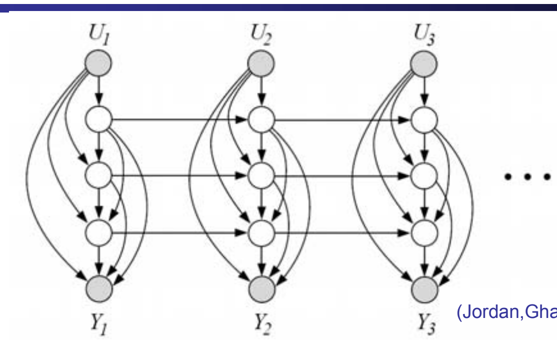
- We have created cliques of size k+1, and there are O(kT) of them. The junction tree algorithm is not efficient for factorial HMMs.

Special case: switching HMM



- Different chains have different state space and different semantics
- The exact calculation is intractable and we must use approximate inference methods

Hidden Markov decision trees



(Jordan,Ghahramani,&Saul,197)

- A combination of decision trees with factorial HMMs
- This gives a "command structure" to the factorial representation
- Appropriate for multi-resolution time series
- Again, the exact calculation is intractable and we must use approximate inference methods

Recall State Space Models (SSMs)

- Also known as linear dynamical system, dynamic linear model, Kalman filter model, etc.

$X_t \in R^D$, $Y_t \in R^M$ and

$$\begin{aligned} P(X_t|X_{t-1} &= \mathcal{N}(X_t; AX_{t-1}, Q) \\ P(Y_t|X_t) &= \mathcal{N}(Y_t; BX_t, R) \end{aligned}$$

- The Kalman lter can compute $P(X_t \mid Y_{1:t})$ in O(min{M^3;D^2}) operations per time step.

Factored linear-Gaussian models produce sparse matrices

- Directed arc from X_{t-1}^i to X_t^j iff $A(i,j)$ >0
  (undirected arc between X_t^i to X_t^j iff $\Sigma^{-1}(i,j)$ >0

- e.g., consider a 2-chain factorial SSM with

$$P(X_t^i \mid X_{t-1}^i) = \mathcal{N}(X_t^i; A^i X_{t-1}^i, Q^i)$$

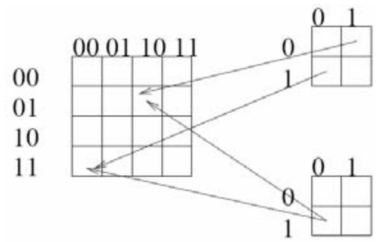
$$P(X_t^1, X_t^2 \mid X_{t-1}^1, X_{t-1}^2) =$$

Discrete-state models

- Factored discrete-state models do NOT produce sparse transition matrices

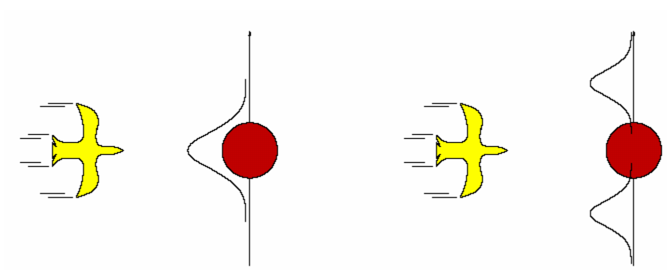
- e.g., consider a 2-chain factorial HMM

$$P(X_t^1, X_t^2 \mid X_{t-1}^1, X_{t-1}^2) = P(X_t^1 \mid X_{t-1}^1) P(X_t^2 \mid X_{t-1}^2)$$



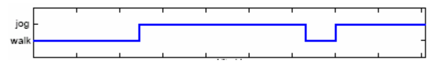
Problems with SSMs

- linearity
- Gaussianity
- Uni-modality

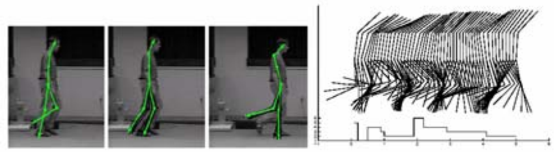


Switching SMM

- Possible world:
  - multiple motion state:

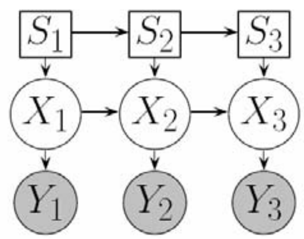


- Task:
  - Trajectory prediction



- Model:
  - Combination of HMM and SSM

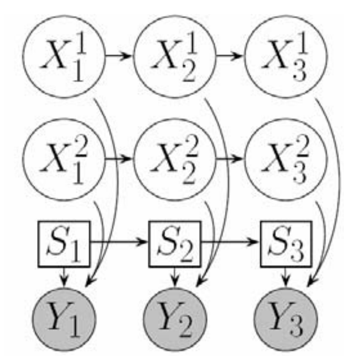
$$p(X_t = x_t \mid X_{t-1} = x_{t-1}, S_t = i) = \mathcal{N}(x_t; A_i x_{t-1}, Q_i)$$
$$p(Y_t = y_t \mid X_t = x_t) = \mathcal{N}(t_t; C x_t, R)$$
$$p(S_t = j \mid S_{t-1} = i) = M(i, j)$$



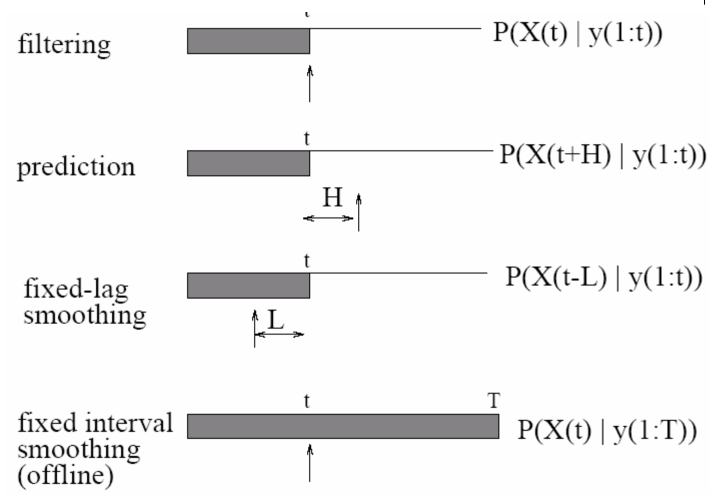
  - Belief state has O(k^t) Gaussian modes:

Data association (correspondence problem)

- Optimal belief state has O(k^t) modes.
- Common to use nearest neighbor approximation.
- For each time slice, can enforce that at most one source causes each observation
- Correspondence problem also arises in shape matching and stereo vision.

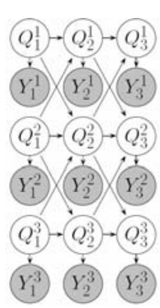


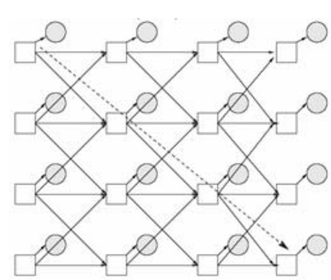
Kinds of inference for DBNs



Complexity of inference in DBN

- Even with local connectivity, everything becomes correlated due to shared common influences in the past.
- E.g. coupled HMM (cHMM)

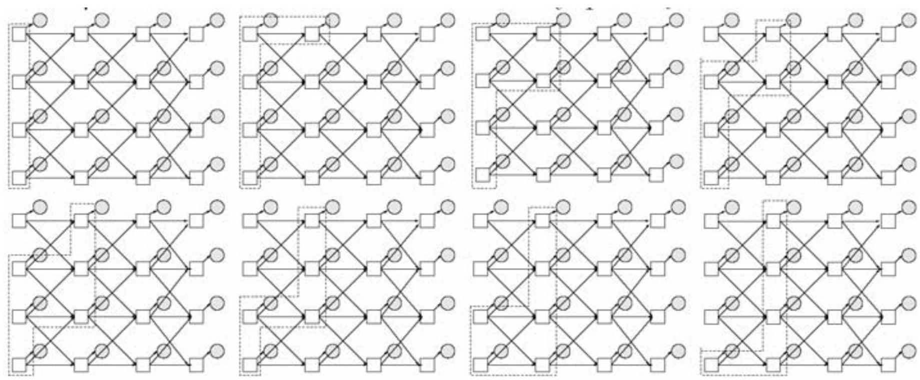




- Even though CHMMs are sparse, all nodes eventually become correlated, so $P(X_t|y_{1:t})$ has size $O(2^N)$.

Junction tree for coupled HMMs

- Cliques form a frontier that snakes from X_{t-1} to X_t.



Approximate Filtering

- Many possible representations for belief state $\alpha_t \equiv P(X_t \mid Y_{1:t})$:
  - Discrete distribution (histogram)
  - Gaussian
  - Mixture of Gaussians
  - Set of samples (particles)

Belief state = discrete distribution

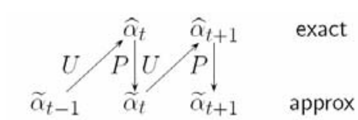
- Discrete distribution is non-parametric (flexible), but intractable.
- Only consider k most probable values --- Beam search.
- Approximate joint as product of factors (ADF/BK approximation)

$$\alpha_t \approx \tilde{\alpha}_t = \prod_{i=1}^{C} P(X_t^i \mid Y_{1:t})$$

Example: Assumed density filtering (ADF)

- ADF forces the belief state to live in some restricted family $\mathcal{F}$, e.g., product of histograms, Gaussian.
- Given a prior $\tilde{\alpha}_{t-1} \in \mathcal{F}$, do one step of exact Bayesian updating to get $\hat{\alpha}_t \notin \mathcal{F}$. Then do a projection step to find the closest approximation in the family:

$$\tilde{\alpha}_t \in \arg\min_{q \in \mathcal{F}} \mathrm{KL}(\hat{\alpha}_t \,\|\, q)$$

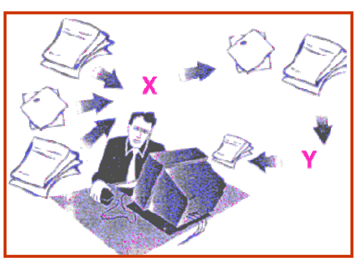


- The Boyen-Koller (BK) algorithm is ADF applied to a DBN
  - e.g., let $\mathcal{F}$ be a product of (singleton) marginals:
- This is also a variational method, and the updating step can still be intractable

Approximate smoothing (off-line)

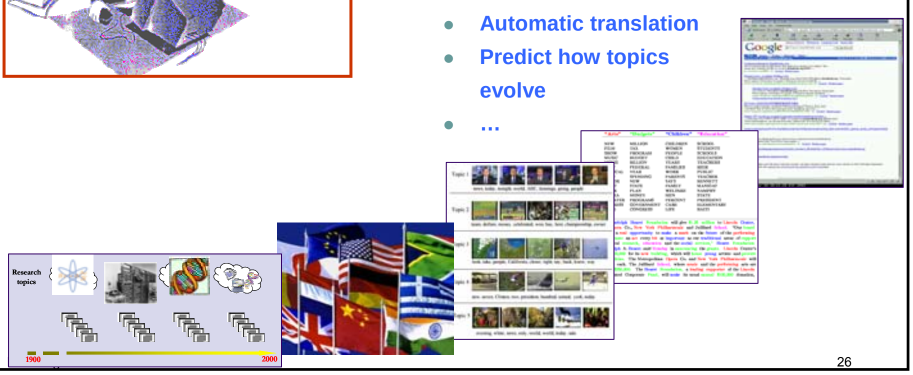
- Two-lter smoothing
- Loopy belief propagation
- Variational methods
- Gibbs sampling
- Can combine exact and approximate methods
- Used as a subroutine for learning

NLP and Data Mining



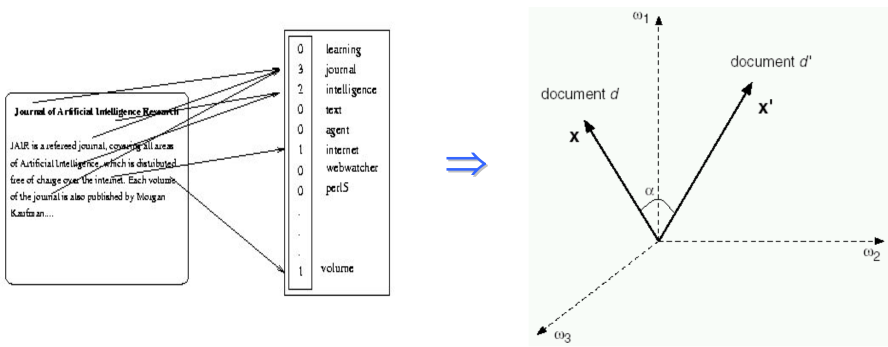
We want:

- **Semantic-based search**
- **infer topics and categorize documents**
- **Multimedia inference**
- **Automatic translation**
- **Predict how topics evolve**
- **…**

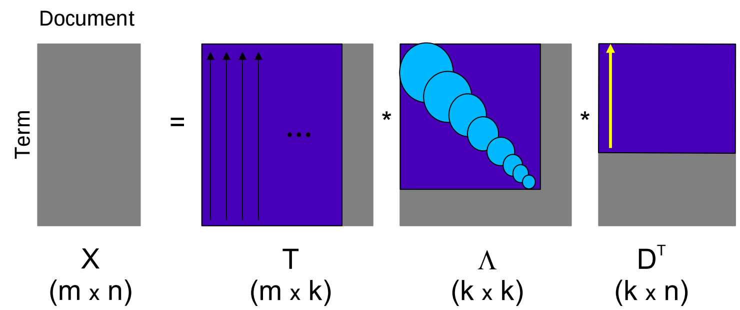


The Vector Space Model

- Represent each document by a high-dimensional vector in the space of words



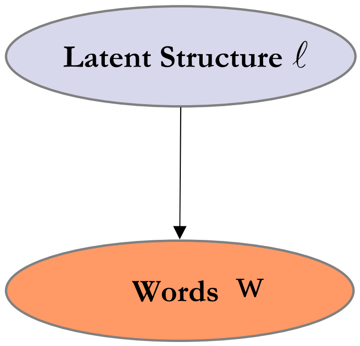
Latent Semantic Indexing



$$\vec{w} = \sum_{k=1}^{K} d_k \lambda_k \vec{T}_k$$

- LSA does not define a properly normalized probability distribution of observed and latent entities
  - Does not support probabilistic reasoning under uncertainty and data fusion

Latent Semantic Structure



Distribution over words

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

Inferring latent structure

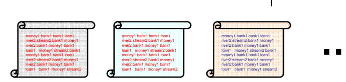
$$P(\ell \mid \mathrm{w}) = \frac{P(\mathrm{w} \mid \ell) P(\ell)}{P(\mathrm{w})}$$

Prediction

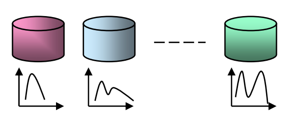
$$P(w_{n+1} \mid \mathrm{w}) = \ldots$$

Admixture Models

- Objects are bags of elements



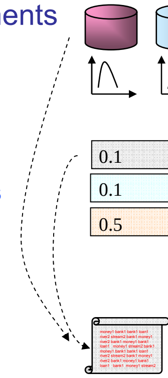
- Mixtures are distributions over elements



- Objects have mixing vector θ
  - Represents each mixtures' contributions

0.1	0.1	…..	0.5
0.1	0.5	…..	0.1
0.5	0.1	…..	0.1

- Object is generated as follows:
  - Pick a mixture component from θ
  - Pick an element from that component



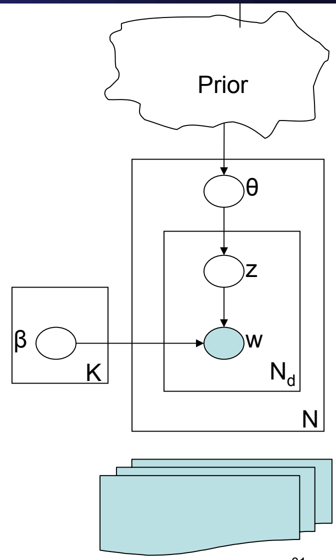
Topic Models =Admixture Models

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 $multinomial(\beta_{z_n})$



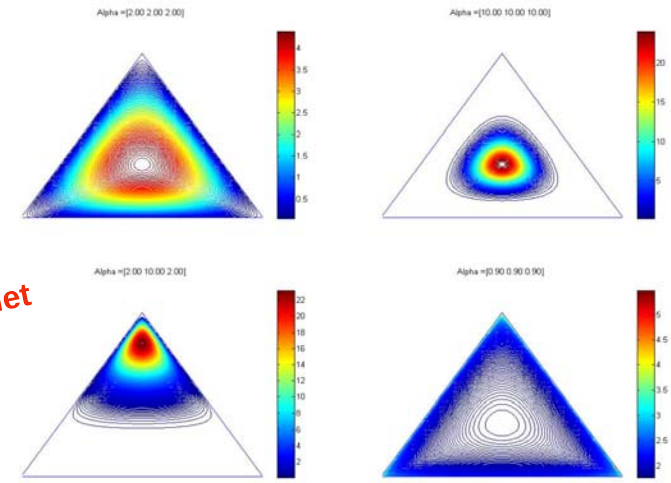
Which prior to use?

Choice of Prior

- Dirichlet (LDA) (Blei et al. 2003)
  - Conjugate prior means efficient inference
  - Can only capture variations in each topic's intensity independently
- Logistic Normal (CTM=LoNTAM) (Blei & Lafferty 2005, Ahmed & Xing 2006)
  - Capture the intuition that some topics are highly correlated and can rise up in intensity together
  - Not a conjugate prior implies hard inference

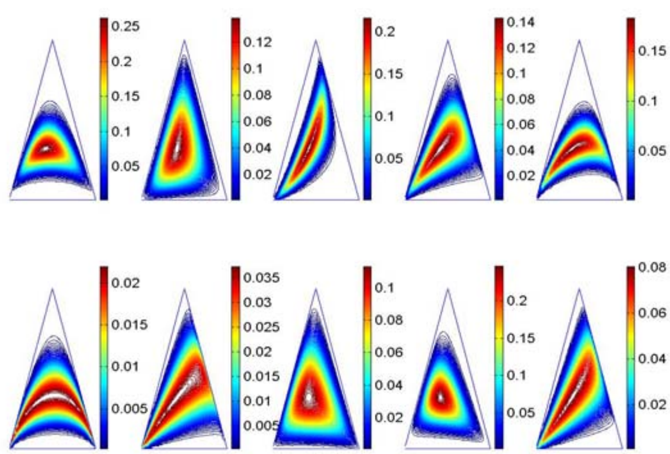
Logistic Normal Vs. Dirichlet

Dirichlet



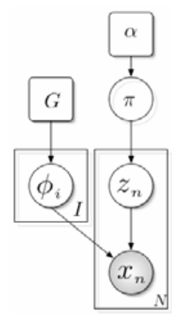
Logistic Normal Vs. Dirichlet

Logistic Normal

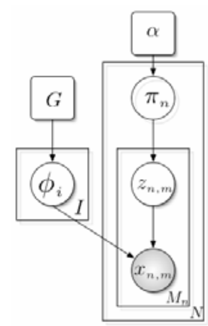


Mixed Membership Model (M³)

- Mixture versus admixture



⇒

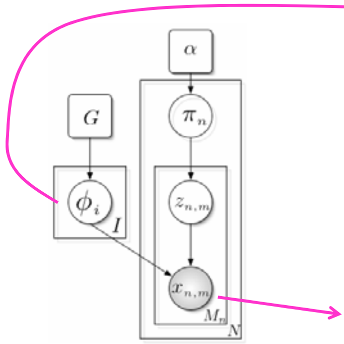


A Bayesian mixture model

A Bayesian admixture model: Mixed membership model

Latent Dirichlet Allocation: M³ in text mining

- A document is a bag of words each generated from a randomly selected topic

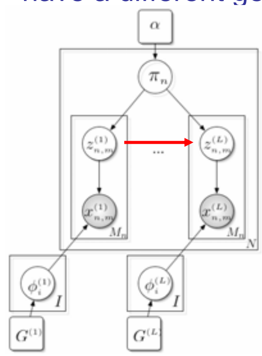


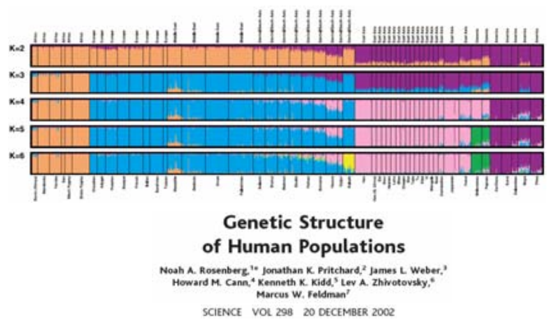
"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

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.

Population admixture: M^3 in genetics

- The genetic materials of each modern individual are inherited from multiple ancestral populations, each DNA locus may have a different generic origin …

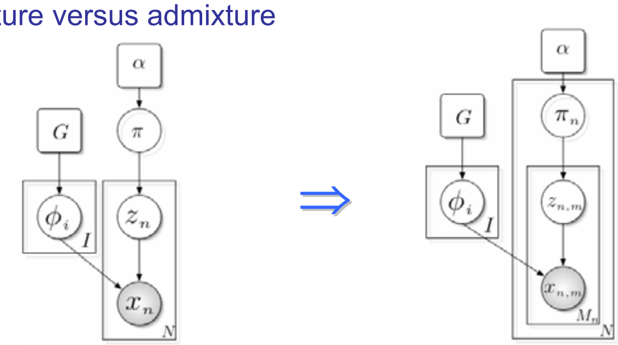




- Ancestral labels may have (e.g., Markovian) dependencies

Inference in Mixed Membership Models

- Mixture versus admixture



$$p(D) = \sum_{\{z_{n,m}\}} \int \cdots \int \left(\prod_n \left(\prod_m p(x_{n,m} \mid \phi_{z_n}) p(z_{n,m} \mid \pi_n) \right) p(\pi_n \mid \alpha) \right) p(\phi \mid G) d\pi_1 \cdots d\pi_N d\phi$$

- Inference is very hard in M^3, all hidden variables are coupled and not factorizable!

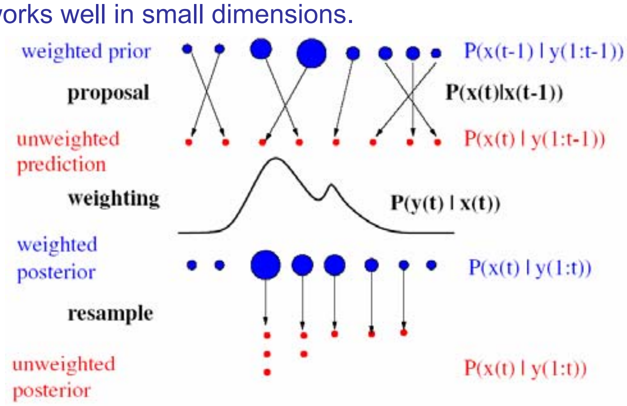
$$p(\pi_n \mid D) \sim \sum_{\{z_{n,m}\}} \int \left(\prod_n \left(\prod_m p(x_{n,m} \mid \phi_{z_n}) p(z_{n,m} \mid \pi_n) \right) p(\pi_n \mid \alpha) \right) p(\phi \mid G) d\pi_{-i} d\phi$$

Approaches to inference

- Exact inference algorithms
  - The elimination algorithm
  - The junction tree algorithms
- Approximate inference techniques
  - Monte Carlo algorithms:
    - Stochastic simulation / sampling methods
    - Markov chain Monte Carlo methods
  - Variational algorithms:
    - Belief propagation
    - Variational inference

Example: Particle filtering (sequential Monte Carlo)

- Represent belief state as weighted set of samples (non-parametric).
- Can handle nonlinear transition/emission and multi-modality.
- Easy to implement.
- Only works well in small dimensions.

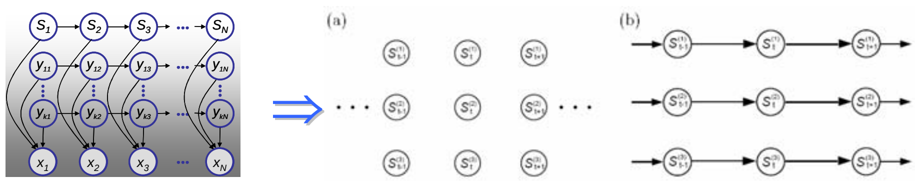


Example: Structured Variational approximation

- Finds an optimal $q^*()$ in a tractable family to approximate the original joint $p()$

$$q^*() \in \arg\min_{q \in \mathcal{T}} \mathrm{F}(q \| p)$$

- There can be many different choices of $\mathcal{T}$ and F().



fHMM   Mean field approx.   Structured variational approx.

Monte Carlo methods

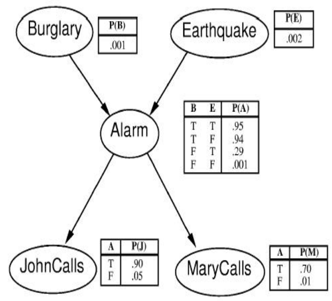
- Draw random samples from the desired distribution
- Yield a stochastic representation of a complex distribution
  - marginals and other expections can be approximated using sample-based averages

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

- **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