

CMPE 58K - Lecture 8.

Bayesian Statistics and Machine Learning

Exact Inference, Junction Tree



Department of Computer Engineering,
Boğaziçi University, Istanbul, Turkey
Instructor: A. Taylan Cemgil

Fall 2008-2009

Outline

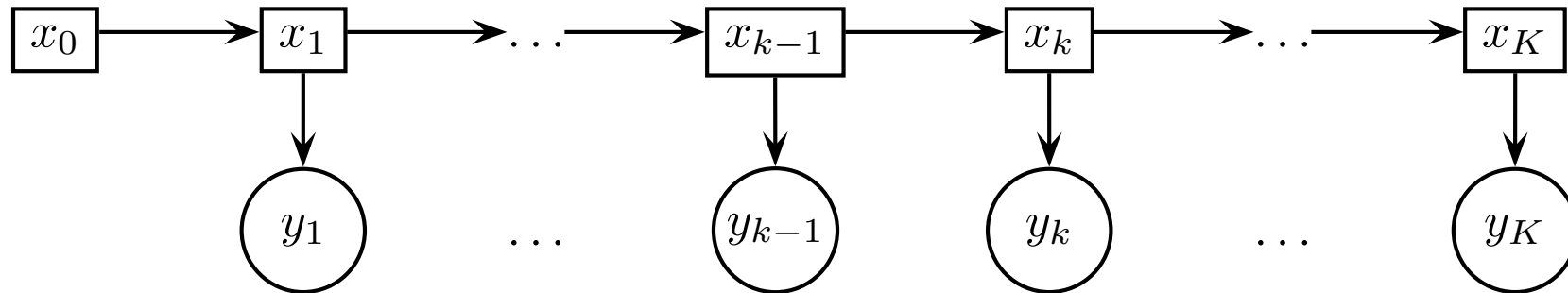
- Hidden Markov Model Review
 - Two filter smoother (forward backward)
 - Correction smoother
- Junction Tree Algorithm
- Summary

Exact Inference

- For a restricted class of models, we can exploit the factorisation of the posterior to compute exact marginals
 - Hidden Markov models with small number of hidden states
 - Gaussian Random fields
 - Bayesian Networks with discrete (multinomial) probability tables
- It is possible to generate inference code automatically for a large class of queries.

Hidden Markov Model [2]

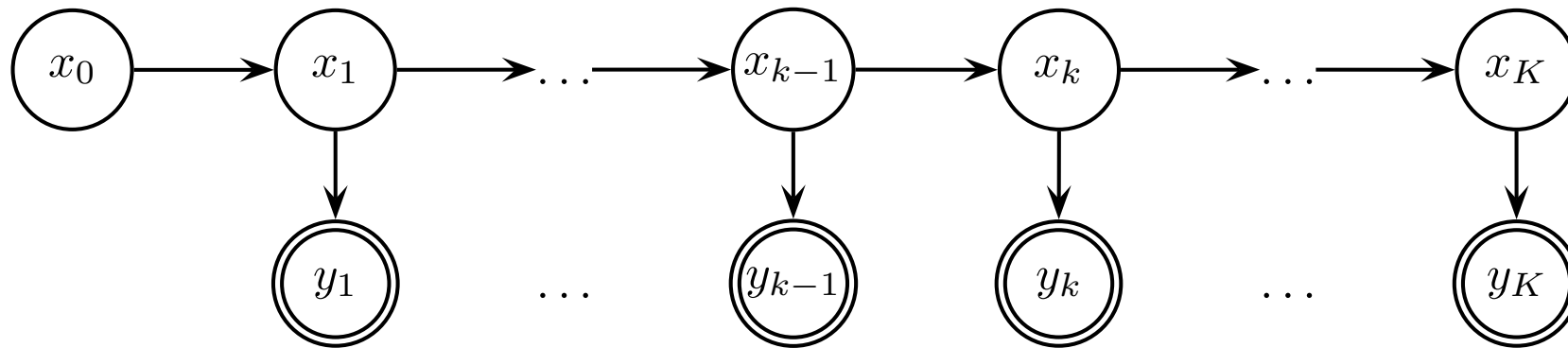
- Mixture model evolving in time



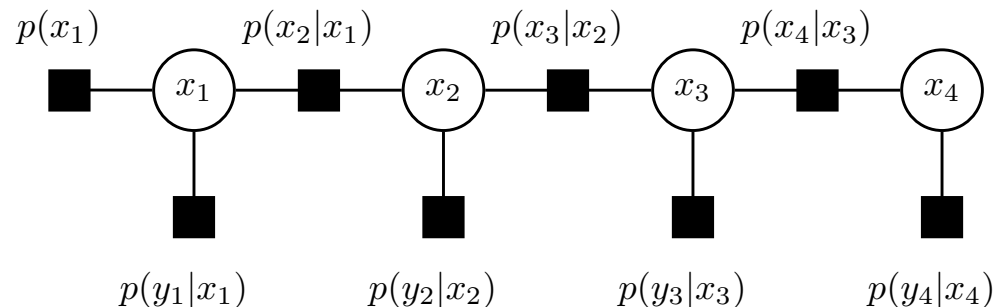
- Observations y_k are continuous or discrete
- Latent variables x_k are discrete
 - Represents the fading memory of the process
- Exact inference possible if x_k has a “small” number of states

Offline Inference, Terminology

- **Smoothing** $p(x_{0:K} | y_{1:K})$,
Most likely trajectory – Viterbi path $\arg \max_{x_{0:K}} p(x_{0:K} | y_{1:K})$



Exact Inference in HMM, Forward/Backward Algorithm



- Forward Pass

$$\begin{aligned}
 p(y_{1:K}) &= \sum_{x_{1:K}} p(y_{1:K}|x_{1:K})p(x_{1:K}) \\
 &= \underbrace{\sum_{x_K} p(y_K|x_K) \sum_{x_{K-1}} p(x_K|x_{K-1}) \cdots \sum_{x_2} p(x_3|x_2) p(y_2|x_2)}_{\alpha_K} \underbrace{\sum_{x_1} p(x_2|x_1) p(y_1|x_1)}_{\alpha_2} \underbrace{p(x_1)}_{\alpha_1}
 \end{aligned}$$

- Backward Pass

$$p(y_{1:K}) = \sum_{x_1} p(x_1)p(y_1|x_1) \cdots \underbrace{\sum_{x_{K-1}} p(x_{K-1}|x_{K-2})p(y_{K-1}|x_{K-1})}_{\beta_{K-2}} \underbrace{\sum_{x_K} p(x_K|x_{K-1})p(y_K|x_K)}_{\beta_{K-1}} \underbrace{1}_{\beta_K}$$

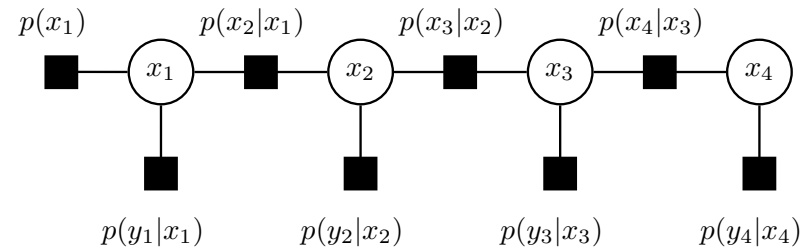
Two filter smoother

- Forward pass: Compute α -messages ($p(x_t, y_{1:t})$)
- Backward pass: Compute β -messages ($p(y_{t+1:T}|x_t)$)
- Smoothing: Compute the product

$$\gamma_t = \alpha_t \times \beta_{t|t+1}$$

and normalise

Forward pass



- Predict

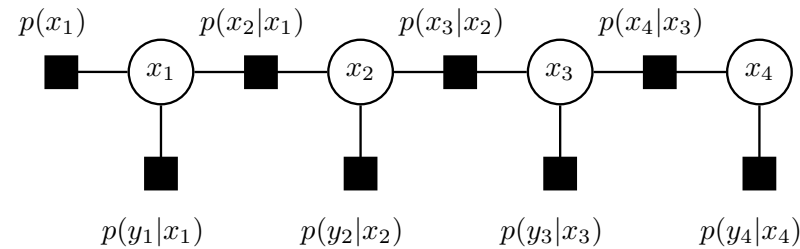
$$\begin{aligned}\alpha_{k|k-1}(x_k) &= p(y_{1:k-1}, x_k) = \sum_{x_{k-1}} p(x_k | x_{k-1}) p(y_{1:k-1}, x_{k-1}) \\ &= \sum_{x_{k-1}} p(x_k | x_{k-1}) \alpha_{k-1|k-1}(x_{k-1})\end{aligned}$$

- Update

$$\begin{aligned}\alpha_{k|k}(x_k) &= p(y_{1:k}, x_k) = p(y_k | x_k) p(y_{1:k-1}, x_k) \\ &= p(y_k | x_k) \alpha_{k|k-1}(x_k)\end{aligned}$$

$$\begin{aligned}
p(y_{1:K}) &= \sum_{x_{1:K}} p(y_{1:K}|x_{1:K})p(x_{1:K}) \\
&= \sum_{x_K} p(y_K|x_K) \sum_{x_{K-1}} p(x_K|x_{K-1}) \cdots \sum_{x_2} p(x_3|x_2)p(y_2|x_2) \sum_{x_1} p(x_2|x_1) \underbrace{p(y_1|x_1) p(x_1)}_{\alpha_{1|1}}^{\alpha_{1|0}} \\
&= \sum_{x_K} p(y_K|x_K) \sum_{x_{K-1}} p(x_K|x_{K-1}) \cdots \sum_{x_2} p(x_3|x_2)p(y_2|x_2) \sum_{x_1} p(x_2|x_1) \alpha_{1|1}(x_1) \\
&= \sum_{x_K} p(y_K|x_K) \sum_{x_{K-1}} p(x_K|x_{K-1}) \cdots \sum_{x_2} p(x_3|x_2)p(y_2|x_2) \alpha_{2|1}(x_2) \\
&= \sum_{x_K} p(y_K|x_K) \sum_{x_{K-1}} p(x_K|x_{K-1}) \cdots \sum_{x_2} p(x_3|x_2) \alpha_{2|2}(x_2) \\
&= \sum_{x_K} p(y_K|x_K) \sum_{x_{K-1}} p(x_K|x_{K-1}) \cdots \alpha_{3|2}(x_3)
\end{aligned}$$

Backward pass



- “Postdict”

$$\begin{aligned}\beta_{k|k+1}(x_k) &= p(y_{k+1:K}|x_k) = \sum_{x_{k+1}} p(x_{k+1}|x_k)p(y_{k+1:K}|x_{k+1}) \\ &= \sum_{x_{k+1}} p(x_{k+1}|x_k)\beta_{k+1|k+1}(x_{k+1})\end{aligned}$$

- Update

$$\begin{aligned}\beta_{k|k}(x_k) &= p(y_{k:K}|x_k) = p(y_k|x_k)p(y_{k+1:K}|x_k) \\ &= p(y_k|x_k)\beta_{k|k+1}(x_k)\end{aligned}$$

$$\begin{aligned}
p(y_{1:K}) &= \sum_{x_1} p(x_1)p(y_1|x_1) \cdots \sum_{x_{K-1}} p(x_{K-1}|x_{K-2})p(y_{K-1}|x_{K-1}) \sum_{x_K} p(x_K|x_{K-1})p(y_K|x_K) \underbrace{\mathbf{1}}_{\beta_{K|K+1}} \\
&= \sum_{x_1} p(x_1)p(y_1|x_1) \cdots \sum_{x_{K-1}} p(x_{K-1}|x_{K-2})p(y_{K-1}|x_{K-1}) \sum_{x_K} p(x_K|x_{K-1})\beta_{K|K} \\
&= \sum_{x_1} p(x_1)p(y_1|x_1) \cdots \sum_{x_{K-1}} p(x_{K-1}|x_{K-2})p(y_{K-1}|x_{K-1})\beta_{K-1|K} \\
&= \sum_{x_1} p(x_1)p(y_1|x_1) \cdots \sum_{x_{K-1}} p(x_{K-1}|x_{K-2})\beta_{K-1|K-1} \\
&= \sum_{x_1} p(x_1)p(y_1|x_1) \cdots \beta_{K-2|K-1}
\end{aligned}$$

Two filter Smoother

$$\begin{aligned} p(y_{1:K}, x_k) &= p(y_{1:k}, x_k) p(y_{k+1:K} | x_k) \\ &= \alpha_{k|k}(x_k) \beta_{k|k+1}(x_k) \\ &\equiv \gamma_k(x_k) \end{aligned}$$

Correction smoother

- We will derive a recursive algorithm to compute the marginals $p(x_t|y_{1:T})$

$$\begin{aligned} p(x_t|y_{1:T}) &= \sum_{x_{t+1}} p(x_t, x_{t+1}|y_{1:T}) = \sum_{x_{t+1}} p(x_t|x_{t+1}, y_{1:T})p(x_{t+1}|y_{1:T}) \\ &= \sum_{x_{t+1}} p(x_t|x_{t+1}, y_{1:t})p(x_{t+1}|y_{1:T}) \\ &= \sum_{x_{t+1}} p(x_t|x_{t+1}, y_{1:t}) \frac{p(x_{t+1}|y_{1:t})}{p(x_{t+1}|y_{1:t})} p(x_{t+1}|y_{1:T}) \\ &= \sum_{x_{t+1}} \frac{p(x_t, x_{t+1}|y_{1:t})}{p(x_{t+1}|y_{1:t})} p(x_{t+1}|y_{1:T}) \\ &= \sum_{x_{t+1}} \frac{p(x_{t+1}|x_t)p(x_t|y_{1:t})}{p(x_{t+1}|y_{1:t})} p(x_{t+1}|y_{1:T}) \end{aligned}$$

Correction smoother

$$p(x_t, x_{t+1} | y_{1:T}) = \frac{p(x_t, x_{t+1} | y_{1:t})}{p(x_{t+1} | y_{1:t})} p(x_{t+1} | y_{1:T})$$

$$\text{New Potential}_{t,t+1} = \frac{\text{Old Potential}_{t,t+1}}{\text{Old Marginal}_{t+1}} \times \text{New Marginal}_{t+1}$$

$$p(x_t | y_{1:T}) = \sum_{x_{t+1}} p(x_t, x_{t+1} | y_{1:T})$$

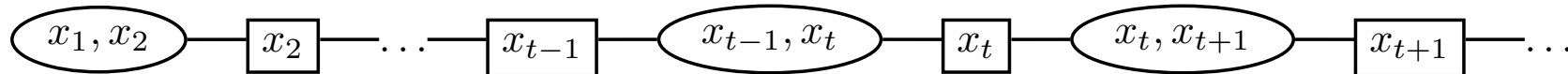
$$\text{New Marginal}_t = \sum_{x_{t+1}} \text{New Potential}_{t,t+1}$$



Correction smoother

At a deeper level, the correction smoother calculates a factorisation of the posterior of form

$$p(x_{1:T}|y_{1:T}) = \frac{\prod_{t=1}^{T-1} p(x_t, x_{t+1}|y_{1:T})}{\prod_{t=2}^{T-1} p(x_t|y_{1:T})}$$



$$\text{Posterior} = \frac{\text{Clique Potentials}}{\text{Separator Potentials}}$$

The general form of the update equation

$$\Psi^{\text{new}}(C) \leftarrow \frac{\Psi(C)}{\Psi(S)} \Psi^{\text{new}}(S)$$

C : Clique

$\Psi(C)$: Clique Potential

S : Separator

$\Psi(S)$: Separator Potential

Junction Tree Algorithms

- Generalise the correction smoother idea to arbitrary graphical models
- There are two similar but different Junction tree algorithms
 - HUGIN (we cover this)
 - Shafer-Shenoy

The junction tree algorithm

Compile time: Building the Inference engine

1. Construction of the junction tree JT :
 - (a) Construct the set of cliques $\mathcal{C} = \cup_{\alpha} C_{\alpha}$
 - (b) Build a weighted cluster graph over the cliques C_{α} .
 - (c) Choose JT to be a maximum-weight spanning tree.
2. Distribute the factors of the density to the cliques of JT
3. Pass messages to render the JT consistent

The junction tree algorithm

Run time: Inference

1. Set the evidence in the potentials of the density.
2. Pass messages according the chosen message passing schedule
3. Read out the desired marginals from the cluster potentials

Junction Tree

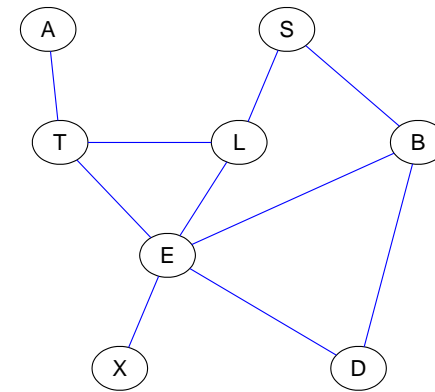
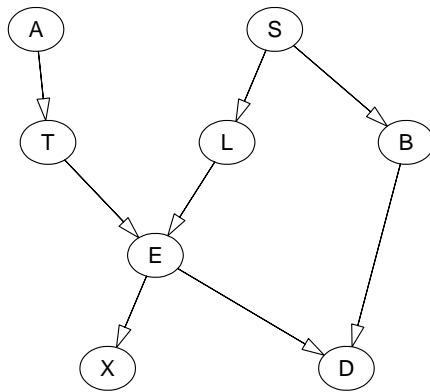
A junction tree JT is a tree of clusters that satisfies the following properties

- **Singly connected:** Ensures JT is a tree
- **Covering:** For each clique of G there is a cluster $C_\alpha \in \mathcal{C}$
- **running intersection property:** For each pair of clusters C_α and C_β containing V , all clusters on the unique path from C_α to C_β contain V

Construction of a Junction Tree

Given a directed acyclic graph G

- $MG = \text{Moralise}(G)$
 - For each node, draw an undirected edge between all parents of a node
 - Drop the directed edges

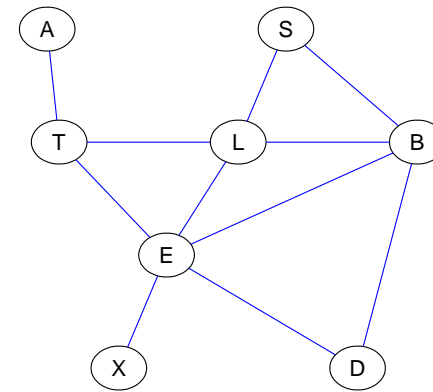
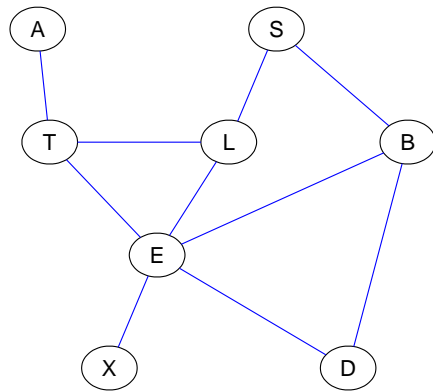


Construction of a Junction Tree

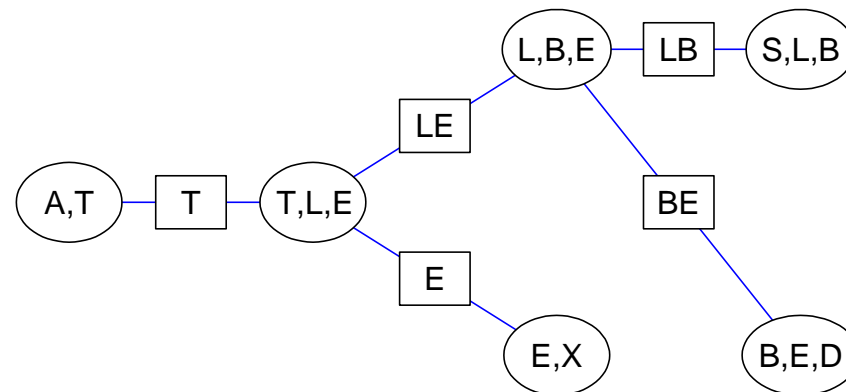
Optimal triangulation (finding smallest possible cliques) is NP-hard. We take a greedy approach.

- $TG = \text{Triangulate}(MG)$
 - Find an elimination sequence of nodes V :
Remove the node V and connect all its neighbors $N(V)$ to each other. Add the cluster $C = \{V, N(V)\}$ to the set of clusters \mathcal{C} , unless there is already a cluster $\bar{C} \in \mathcal{C}$ such that $C \subset \bar{C}$.
- $JT = \text{JTREE}(TG)$
 - Form a Cluster graph where the weight of each (undirected) edge $E(\alpha, \beta)$ is $|C_\alpha \cap C_\beta|$
 - The junction tree is the maximum spanning tree of the cluster graph

Construction of a Junction Tree



Construction of a Junction Tree



Inference Engine

- Distribute the potentials to cliques
- Choose a root node
- From the root node : Distribute Evidence
- From the leaf nodes : Collect Evidence

Summary

- Bayesian Inference
- Graphical models
- Exact Inference
- Approximate inference

Summary, Attributes of Probabilistic Inference

- **Exact** \leftrightarrow **Approximate**
- **Deterministic** \leftrightarrow **Stochastic**
- **Online** \leftrightarrow **Offline**
- **Centralized** \leftrightarrow **Distributed**

Summary of what we have mentioned

- Exact inference, Belief Propagation
- Approximate inference
 - Deterministic
 - * Variational Bayes,
 - * Expectation/Maximization (EM), Iterative Conditional Modes (ICM)
 - Stochastic
 - * Markov Chain Monte Carlo
 - * Importance Sampling,
 - * Particle filtering

Summary of what we have not mentioned

- Exact Inference (Junction Tree ...)
- Deterministic Inference
 - Assumed Density Filter (ADF), Extended Kalman Filter (EKF), Unscented Particle Filter
 - Structured Mean field
 - Loopy Belief Propagation, Expectation Propagation, Generalized Belief Propagation
 - Fractional Belief propagation, Bound Propagation, <your favorite name> Propagation
 - Graph cuts ...
- Stochastic
 - Unscented Particle Filter, Nonparametric Belief Propagation
 - Annealed Importance Sampling, Adaptive Importance Sampling
 - Hybrid Monte Carlo, Exact sampling, Coupling from the past

Bibliography

- General background about probability theory
- Graphical models
- Exact inference
- Variational Methods
- Markov Chain Monte Carlo
- Sequential Monte Carlo
- Applications

General background about probability theory

- Dimitri P. Bertsekas and John N. Tsitsiklis. Introduction to Probability. Athena Scientific, 2002
- Geoffrey Grimmett and David Stirzaker, Probability and Random Processes, (3rd Ed), Oxford, 2006

“Instant Classics” of Bayesian Machine Learning and Graphical Models

- Michael I. Jordan, Learning in Graphical Models, 1998
- David Mackay Information Theory, Learning and Inference Algorithms, 2003, Cambridge
- Chris Bishop, Machine Learning and Pattern Recognition, 2006, Springer

Further Reading, Variational Methods

- Jaakkola “Tutorial on variational approximation methods”, 2000
<http://people.csail.mit.edu/tommi/papers/Jaa-var-tutorial.ps>
- Wainwright and Jordan 2003 [3] Berkeley EECS Tech. Rep.
- Frey and Jojic, PAMI 2005 [1]
- Winn and Bishop “Variational Message Passing” 2005 JMLR [4]

Further Reading, MCMC and SMC tutorials and overviews

- Andrieu, de Freitas, Doucet, Jordan. *An Introduction to MCMC for Machine Learning*, 2001
- Andrieu. *Monte Carlo Methods for Absolute beginners*, 2004
- Doucet, Godsill, Andrieu. "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering", *Statistics and Computing*, vol. 10, no. 3, pp. 197-208, 2000

Books on Monte Carlo techniques

- Gilks, Richardson, Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman Hall, 1996
- Doucet, de Freitas, Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001
- Jun S. Liu, *Monte Carlo Strategies for Scientific computing*, Springer 2004
 - Short book
 - Covers almost everything we have mentioned on MCMC and SMC + more

Some Generic Software Packages

- Kevin Murphy's Matlab Bayesian Networks toolkit (BNT)
- Gilks, et. al. BUGS, WinBUGS – (Bayesian analysis Using Gibbs Sampling) A powerful program that compiles Gibbs Samplers from
- Winn, et. al, VIBES – Similar to BUGS but for variational inference

What Next?

- CMPE 58♠¹, Spring 2009

Monte Carlo methods for data analysis and scientific computing

Generating random variates, Basic principles, Rejection, Reweighting and variance reduction, Importance sampling, Rejection control, Sequential Monte Carlo, Metropolis algorithm, Reversible Jump, Gibbs sampler, Ising models and Boltzman machines, Clustering methods and Swendsen-Wang, General conditional sampling, Hybrid Monte Carlo, Multilevel sampling and Optimisation, Simulated annealing and Bridging, Population Monte Carlo, Markov chains and convergence, Propp-Wilson, Coupling from the Past

- If you liked CMPE 58K, you will like CMPE 58♠

¹Tentative course code, may change

What Next?

- CMPE 58[👂], Machine Listening
- Applications of probabilistic models to Music and Audio processing

References

- [1] B. J. Frey and N. Jovic. A comparison of algorithms for inference and learning in probabilistic graphical models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(9), 2005.
- [2] L. R. Rabiner. A tutorial in hidden Markov models and selected applications in speech recognition. Proc. of the IEEE, 77(2):257–286, 1989.
- [3] M. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Technical Report 649, Department of Statistics, UC Berkeley, September 2003.
- [4] J. Winn and C. Bishop. Variational message passing. Journal of Machine Learning Research, 6:661–694, 2005.