Lecture 13

Learning Bayesian Networks from Data

Thursday, October 7, 1999

William H. Hsu
Department of Computing and Information Sciences, KSU
http://www.cis.ksu.edu/~bhsu

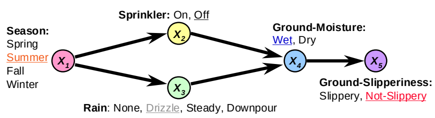
Readings:
Sections 6.11-6.13, Mitchell
"A Theory of Inferred Causation", Pearl and Verma
"A Tutorial on Learning Bayesian Networks", Heckerman

Lecture Outline

- Readings: 6.11-6.13, Mitchell; Pearl and Verma; Heckerman Tutorial
- More Bayesian Belief Networks (BBNs)
  - Inference: applying CPTs
  - Learning: CPTs from data, elicitation
  - In-class exercises
    - *Hugin*, *BKD* demos
    - CPT elicitation, application
- Learning BBN Structure
  - *K2* algorithm
  - Other probabilistic scores and search algorithms
- Causal Discovery: Learning Causality from Observations
- Incomplete Data: Learning and Inference (Expectation-Maximization)
- Next Week: BBNs Concluded; Review for Midterm (10/14/1999)
- After Midterm: EM Algorithm, Unsupervised Learning, Clustering

Bayesian Networks: Quick Review

- Recall: Conditional Independence (CI) Assumptions
- Bayesian Network: Digraph Model
  - Vertices (nodes): denote events (each a random variable)
  - Edges (arcs, links): denote conditional dependencies
- Chain Rule for (Exact) Inference in BBNs $P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid parents(X_i))$
  - Arbitrary Bayesian networks: $\mathcal{NP}$-complete
  - Polytrees: linear time
- Example ("Sprinkler" BBN)



P(*Summer*, *Off*, *Drizzle*, *Wet*, *Not-Slippery*) = *P*(*S*) · *P*(*O* | *S*) · *P*(*D* | *S*) · *P*(*W* | *O*, *D*) · *P*(*N* | *W*)

- MAP, ML Estimation over BBNs $h_{ML} = \arg\max_{h \in H} P(D \mid h)$

Learning Distributions in BBNs: Quick Review

- Learning Distributions
  - Shortcomings of Naïve Bayes
  - Making *judicious CI assumptions*
  - Scaling up to BBNs: need to learn a CPT for all parent sets
  - Goal: *generalization*
    - Given *D* (e.g., {1011, 1001, 0100})
    - Would like to know *P*(*schema*): e.g., $P(11**) \equiv P(x_1 = 1, x_2 = 1)$
- Variants
  - Known or unknown structure
  - Training examples may have missing values
- Gradient Learning Algorithm
  - Weight update rule

$$w_{ijk} \leftarrow w_{ijk} + r \sum_{x \in D} \frac{P_h(y_{ij}, u_{ik} \mid x)}{w_{ijk}}$$

  - Learns CPTs given data points *D*

Learning Structure

- Problem Definition
  - Given: data *D* (tuples or vectors containing observed values of variables)
  - Return: directed graph (*V*, *E*) expressing *target CPTs* (or commitment to acquire)
- Benefits
  - Efficient learning: more accurate models with less data - *P*(*A*), *P*(*B*) vs. *P*(*A*, *B*)
  - Discover structural properties of the domain (causal relationships)
- Acccurate Structure Learning: Issues
  - Superfluous arcs: more parameters to fit; wrong assumptions about causality
  - Missing arcs: cannot compensate using CPT learning; ignorance about causality
- Solution Approaches
  - Constraint-based: enforce consistency of network with observations
  - Score-based: optimize degree of match between network and observations
- Overview: Tutorials
  - [Friedman and Goldszmidt, 1998] http://robotics.Stanford.EDU/people/nir/tutorial/
  - [Heckerman, 1999] http://www.research.microsoft.com/~heckerman

Learning Structure: Constraints Versus Scores

- Constraint-Based
  - Perform tests of conditional independence
  - Search for network consistent with observed dependencies (or lack thereof)
  - Intuitive; closely follows definition of BBNs
  - Separates construction from form of CI tests
  - Sensitive to errors in individual tests
- Score-Based
  - Define scoring function (*aka* score) that evaluates how well (in)dependencies in a structure match observations
  - Search for structure that maximizes score
  - Statistically and information theoretically motivated
  - Can make compromises
- Common Properties
  - Soundness: with sufficient data and computation, both learn correct structure
  - Both learn structure from observations and *can incorporate knowledge*

Learning Structure: Maximum Weight Spanning Tree (Chow-Liu)

- Algorithm *Learn-Tree-Structure-I* (*D*)
  - Estimate $P(x)$ and $P(x, y)$ for all single RVs, pairs; $I(X; Y) = D(P(X, Y) \,||\, P(X) \cdot P(Y))$
  - Build *complete* undirected graph: variables as vertices, $I(X; Y)$ as edge weights
  - $T \leftarrow$ *Build-MWST* ($V \times V$, *Weights*) // Chow-Liu algorithm: weight function ≡ I
  - Set directional flow on *T* and place the CPTs on its edges (gradient learning)
  - RETURN: tree-structured BBN with CPT values
- Algorithm *Build-MWST-Kruskal* ($E \subseteq V \times V$, *Weights*: $E \rightarrow \mathbf{R}^+$)
  - $H \leftarrow$ *Build-Heap* (*E*, *Weights*) // *aka* priority queue — $O(|E|)$
  - $E' \leftarrow \emptyset$; *Forest* $\leftarrow \{\{v\} \mid v \in V\}$ // *E'*: set; *Forest*: union-find — $O(|V|)$
  - WHILE *Forest.Size* > 1 DO — $O(|E|)$
    - $e \leftarrow$ *H.Delete-Max*() // $e \equiv$ new edge from *H* — $O(\lg |E|)$
    - IF (($T_S \leftarrow$ *Forest.Find*(*e.Start*)) $\neq$ ($T_E \leftarrow$ *Forest.Find*(*e.End*))) THEN — $O(\lg^* |E|)$
      - *E'.Union*(*e*) // append edge *e*; *E'.Size*++ — $O(1)$
      - *Forest.Union* (T_S, T_E) // *Forest.Size*-- — $O(1)$
  - RETURN *E'* — $O(1)$
- Running Time: $O(|E| \lg |E|) = O(|V|^2 \lg |V|^2) = O(|V|^2 \lg |V|) = O(n^2 \lg n)$

Learning Structure: Overfitting Prevention and Avoidance

- "Classic" Issue in Machine Learning
  - *h'* worse than *h* on D_{train}
  - *h'* better than *h* on D_{test}
- Standard Approaches
  - Prevention: restricted hypothesis space *H*
    - Limits overfitting capability
    - Examples: *restrict number of parents*, number of parameters
  - Avoidance: Minimum Description Length (MDL)
    - Description length $MDL(h) = -BIC(h) = -\lg P(D \mid h) - \lg P(h)$ measures complexity
    - Choose model that compactly describes *D*
  - Avoidance: Bayesian methods (cf. BOC)
    - Average over all possible values of BBN parameters Θ
    - Use prior knowledge
- Other Approaches
  - Holdout, cross-validation (CV), leave-one-out
  - Structural risk minimization: penalize $H' \subseteq H$ based on their VC dimension

Scores for Learning Structure: The Role of Inference

- General-Case BBN Structure Learning: *Use Inference to Compute Scores*
- Recall: Bayesian Inference *aka* Bayesian Reasoning
  - Assumption: $h \in H$ are mutually exclusive and exhaustive
  - Optimal strategy: *combine predictions of hypotheses in proportion to likelihood*
    - Compute conditional probability of hypothesis *h* given observed data *D*
    - i.e., compute expectation over unknown *h* for unseen cases
    - Let $h \equiv$ structure, parameters $\Theta \equiv$ CPTs

$$P\left(\vec{x}^{(m+1)} \mid D\right) = P\left(x_1, x_2, \ldots, x_n \mid \vec{x}^{(1)}, \vec{x}^{(2)}, \ldots, \vec{x}^{(m)}\right) = \sum_{h \in H} P\left(\vec{x}^{(m+1)} \mid D, h\right) \cdot P(h \mid D)$$

Posterior Score — Marginal Likelihood — Prior over Parameters

$$\underbrace{P(h \mid D)}_{\text{Posterior Score}} \propto \underbrace{P(D \mid h)}_{\text{Marginal Likelihood}} \cdot P(h) = \underbrace{P(h)}_{\text{Prior over Structures}} \cdot \int \underbrace{P(D \mid h, \Theta)}_{\text{Likelihood}} \cdot \underbrace{P(\Theta \mid h)}_{\text{Prior over Parameters}} d\Theta$$

Scores for Learning Structure: Prior over Parameters

- Likelihood $L(\Theta : D)$
  - Definition: $L(\Theta : D) \equiv P(D \mid \Theta) = \prod_{x \in D} P(x \mid \Theta)$
  - General BBN (i.i.d data *x*): $L(\Theta : D) \equiv \prod_{x \in D} \prod_i P(x_i \mid Parents(x_i) \sim \Theta) = \prod_i L(\Theta_i : D)$
    - NB: Θ specifies CPTs for $Parents(x_i)$
    - Likelihood decomposes *according to the structure of the BBN*
- Estimating Prior over Parameters: $P(\Theta \mid D) \propto P(D) \cdot P(D \mid \Theta) \equiv P(D) \cdot L(\Theta : D)$
  - Example: *Sprinkler*
    - Scenarios *D* = {(*Season*(*i*), *Sprinkler*(*i*), *Rain*(*i*), *Moisture*(*i*), *Slipperiness*(*i*))}
    - *P*(*Su*, *Off*, *Dr*, *Wet*, *NS*) = *P*(*S*) · *P*(*O* | *S*) · *P*(*D* | *S*) · *P*(*W* | *O*, *D*) · *P*(*N* | *W*)
  - MLE for multinomial distribution (e.g., {Spring, Summer, Fall, Winter}): $\hat{\Theta}_k = \dfrac{N_k}{\sum_l N_l}$
  - Likelihood for multinomials $L(\Theta : D) = \prod_{k=1}^{K} \Theta_k^{N_k}$
  - Binomial case: N_1 = # heads, N_2 = # tails ("frequency is ML estimator")

Learning Structure: Dirichlet (Bayesian) Score and *K2* Algorithm

- Dirichlet Prior
  - Definition: a Dirichlet prior with hyperparameters $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ is a distribution

$$P(\Theta) \propto \prod_{k=1}^{K} \Theta_k^{\alpha_k - 1} \text{ for legal } \Theta_k$$

  - Posterior has the same score, with hyperparameters $\{\alpha_1 - N_1, \alpha_2 - N_2, \ldots, \alpha_k - N_k\}$

$$P(\Theta \mid D) \propto P(\Theta) \cdot P(D \mid \Theta) \propto \prod_{k=1}^{K} \Theta_k^{\alpha_k - 1} \cdot \prod_{k=1}^{K} \Theta_k^{N_k} = \prod_{k=1}^{K} \Theta_k^{\alpha_k + N_k - 1}$$

- Bayesian Score (*aka* Dirichlet Score) for Marginal Likelihood *P*(*D* | *h*)

$$P(D \mid h) \propto \prod_{i=1}^{n} \left[\prod_{Pa_i^h} \frac{\Gamma\left(\alpha\left(Pa_i^h\right)\right)}{\Gamma\left(\alpha\left(Pa_i^h\right) + N\left(Pa_i^h\right)\right)} \cdot \prod_{X_i = x_i} \frac{\Gamma\left(\alpha\left(x_i, Pa_i^h\right) + N\left(x_i, Pa_i^h\right)\right)}{\Gamma\left(\alpha\left(x_i, Pa_i^h\right)\right)} \right]$$

where $x_i \equiv x_{ij} \equiv$ particular value of X_i, $Pa_i^h \equiv Pa_{ik}^h \equiv$ particular value of $Parents_h(x_i)$,

$\Gamma(i) = (i-1)!$ for $i \in \mathbb{Z}^+$

- *K2*: Algorithm for General Case Structure Learning
  - Greedy, Bayesian score-based
  - See: http://wilma.cs.brown.edu/research/ai/dynamics/tutorial/

Learning Structure: *K2* Algorithm and *ALARM*

- Algorithm *Learn-BBN-Structure-K2* (*D*, *Max-Parents*)

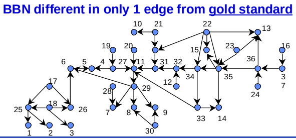
  FOR *i* ← 1 to *n* DO // arbitrary ordering of variables $\{x_1, x_2, \ldots, x_n\}$

  WHILE (*Parents*[x_i]*.Size* < *Max-Parents*) DO // find best candidate parent

  Best ← *argmax*$_{j>i}$ (*P*(*D* | $x_j \in$ *Parents*[x_i]) // max Dirichlet score

  IF (*Parents*[x_i] + *Best*)*.Score* > *Parents*[x_i]*.Score*) THEN *Parents*[x_i] += *Best*

  RETURN ({*Parents*[x_i] | $i \in$ {1, 2, …, *n*}})
- A Logical Alarm Reduction Mechanism [Beinlich *et al*, 1989]
  - BBN model for patient monitoring in surgical anesthesia
  - Vertices (37): findings (e.g., *esophageal intubation*), intermediates, observables
  - *K2*: found BBN different in only 1 edge from gold standard (elicited from expert)

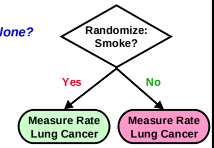


Learning Structure: (Score-Based) Hypothesis Space Search

- Learning Structure: Beyond Trees
  - Problem not as easy for more complex networks
  - Example
    - Allow two parents (even singly-connected case, *aka* polytree)
    - Greedy algorithms no longer guaranteed to find optimal network
    - In fact, *no efficient algorithm exists*
  - Theorem: finding network structure with maximal score, where *H* restricted to BBNs with at most k parents for each variable, is $\mathcal{NP}$-hard for $k > 1$
- Heuristic Search of Search Space *H*
  - Define *H*: elements denote possible structures, adjacency relation denotes transformation (e.g., arc addition, deletion, reversal)
  - Traverse this space looking for high-scoring structures
  - Algorithms
    - Greedy hill-climbing
    - Best-first search
    - Simulated annealing

Learning Structure: Causal Discovery

- Learning for Decision Support in Policy-Making
  - Does *smoking* cause *cancer*?
  - Does *ingestion of lead paint* decrease *IQ*?
  - Do *school vouchers* improve *education*?
  - Do *Microsoft business practices* harm *customers*?
- Causal Discovery: Inferring Existence, Direction of Causal Relationships
  - Methodology: by experiment
  - *Can discover causality from observational data alone?*
- What is "Causality" Anyway?
  - Probabilistic question
    - What is *P*(*lung cancer* | *yellow fingers*)?
  - Causal (mechanistic) question
    - What is *P*(*lung cancer* | set (*yellow fingers*))?



- Constraint-Based Methods for Causal Discovery
  - Require: *no unexplained correlations, no accidental independencies* (cause $\wedge$ CI)
  - Find: plausible topologies under local CI tests (cause $\Leftrightarrow$ $\neg$CI)

In-Class Exercise: *Hugin* Demo

- *Hugin*
  - Commercial product for BBN inference: http://www.hugin.com
  - First developed at University of Aalborg, Denmark
- Applications
  - Popular research tool for inference and learning
  - Used for real-world decision support applications
    - Safety and risk evaluation: http://www.hugin.com/serene/
    - Diagnosis and control in unmanned subs: http://advocate.e-motive.com
    - Customer support automation: http://www.cs.auc.dk/research/DSS/SACSO/
- Capabilities
  - Lauritzen-Spiegelhalter algorithm for inference (clustering *aka* clique reduction)
  - Object Oriented Bayesian Networks (OOBNs): structured learning and inference
  - Influence diagrams for decision-theoretic inference (utility + probability)
  - See: http://www.hugin.com/doc.html

In-Class Exercise: *Hugin* and CPT Elicitation

- *Hugin* Tutorials
  - Introduction: *causal reasoning for diagnosis in decision support* (toy problem)
    - http://www.hugin.com/hugintro/bbn_pane.html
    - Example domain: explaining low yield (drought versus disease)
  - Tutorial 1: constructing a simple BBN in *Hugin*
    - http://www.hugin.com/hugintro/bbn_tu_pane.html
    - Eliciting CPTs (or collecting from data) and entering them
  - Tutorial 2: constructing a simple influence diagram (decision network) in *Hugin*
    - http://www.hugin.com/hugintro/id_tu_pane.html
    - Eliciting utilities (or collecting from data) and entering them
- Other Important BBN Resources
  - Microsoft Bayesian Networks: http://www.research.microsoft.com/dtas/msbn/
  - XML BN (Interchange Format): http://www.research.microsoft.com/dtas/bnformat/
  - BBN Repository (more data sets)
    http://www-nt.cs.berkeley.edu/home/nir/public_html/Repository/index.htm

In-Class Exercise: *Bayesian Knowledge Discoverer* (*BKD*) Demo

- *Bayesian Knowledge Discoverer* (*BKD*)
  - Research product for BBN structure learning: http://kmi.open.ac.uk/projects/bkd/
  - Bayesian Knowledge Discovery Project [Ramoni and Sebastiani, 1997]
    - Knowledge Media Institute (KMI), Open University, United Kingdom
    - Closed source, beta freely available for educational use
  - Handles missing data
  - Uses Branch and Collapse: Dirichlet score-based BOC approximation algorithm
    http://kmi.open.ac.uk/techreports/papers/kmi-tr-41.ps.gz
- Sister Product: Robust Bayesian Classifier (*RoC*)
  - Research product for BBN-based classification with missing data
    http://kmi.open.ac.uk/projects/bkd/pages/roc.html
  - Uses Robust Bayesian Estimator, a deterministic approximation algorithm
    http://kmi.open.ac.uk/techreports/papers/kmi-tr-79.ps.gz

Learning Structure: Conclusions

- Key Issues
  - Finding a criterion for inclusion or exclusion of an edge in the BBN
  - Each edge
    - "Slice" (axis) of a CPT *or* a commitment to acquire one
    - Positive statement of conditional dependency
- Other Techniques
  - Focus today: constructive (score-based) view of BBN structure learning
  - Other score-based algorithms
    - Heuristic search over space of addition, deletion, reversal operations
    - Other criteria (information theoretic, coding theoretic)
  - Constraint-based algorithms: *incorporating knowledge into causal discovery*
- Augmented Techniques
  - Model averaging: optimal Bayesian inference (integrate over structures)
  - Hybrid BBN/DT models: use a decision tree to record *P*(*x* | *Parents*(*x*))
- Other Structures: e.g., Belief Propagation with Cycles

Bayesian Network Learning: Related Fields and References

- ANNs: BBNs as Connectionist Models
- GAs: BBN Inference, Learning as Genetic Optimization, Programming
- Hybrid Systems (Symbolic / Numerical AI)
- Conferences
  - General (with respect to machine learning)
    - International Conference on Machine Learning (ICML)
    - American Association for Artificial Intelligence (AAAI)
    - International Joint Conference on Artificial Intelligence (IJCAI, biennial)
  - Specialty
    - International Joint Conference on Neural Networks (IJCNN)
    - Genetic and Evolutionary Computation Conference (GECCO)
    - Neural Information Processing Systems (NIPS)
    - Uncertainty in Artificial Intelligence (UAI)
    - Computational Learning Theory (COLT)
- Journals
  - General: *Artificial Intelligence*, *Machine Learning*, *Journal of AI Research*
  - Specialty: *Neural Networks*, *Evolutionary Computation*, etc.

Learning Bayesian Networks: Missing Observations

- Problem Definition
  - Given: data (*n*-tuples) with missing values, *aka* partially observable (PO) data
  - Kinds of missing values
    - Undefined, unknown (possible *new*)
    - Missing, corrupted (not properly collected)
  - Second case ("truly missing"): want to fill in *?* with expected value
- Solution Approaches
  - Expected = distribution over possible values
  - Use "best guess" BBN to estimate distribution
  - Expectation-Maximization (EM) algorithm can be used here
- Intuitive Idea
  - Want to find h_{ML} in PO case ($D \equiv$ *unobserved variables* ° *observed variables*)
  - Estimation step: calculate E[*unobserved variables* | h], assuming current h
  - Maximization step: update w_{ijk} to maximize $E[\lg P(D \mid h)]$, $D \equiv$ *all variables*

CIS 798: Intelligent Systems and Machine Learning
Kansas State University
Department of Computing and Information Sciences

Expectation-Maximization (EM)

- Intuitive Idea
  - In fully observable case: $h_{ML} = \arg\max_{h \in H} \frac{\text{\# data cases with } \vec{n}, \vec{e}}{\text{\# data cases with } \vec{e}} = \arg\max_{h \in H} \frac{\sum_j I_{\vec{N}=\vec{n}, \vec{E}=\vec{e}}(\vec{x}_j)}{\sum_j I_{\vec{E}=\vec{e}}(\vec{x}_j)}$
  - $h \equiv$ BBN parameters (Θ), $N_i \equiv$ *unobserved variable*, $E_i \equiv$ *observed variable*
  - $I_{E_i = e_i}(\vec{X}_j) = \delta(e_i, X_{ji})$

$$= \begin{cases} 1 \text{ if } (X_{ji} \equiv E_i) = e_i \text{ in data case } \vec{X}_j \\ 0 \text{ otherwise} \end{cases}$$

- Partially Observable Case
  - I is unknown
  - Best estimate for I: $\hat{I}(\vec{n}, \vec{e} \mid \vec{x}) = P(\vec{n}, \vec{e} \mid \vec{x}, h_{ML})$, $h_{ML} \equiv \Theta_{ML}$ – unknown!
- Incomplete Data: Learning and Inference
  - Missing values: to be filled in given partial observations
  - Expectation-Maximization (EM): iterative refinement algorithm
    - Estimation step: use current parameters Θ to estimate missing $\{N_i\}$
    - Maximization (re-estimation) step: update Θ to maximize $P(N_i, E_j \mid D)$

CIS 798: Intelligent Systems and Machine Learning
Kansas State University
Department of Computing and Information Sciences

Continuing Research on Learning Bayesian Networks from Data

- Advanced Topics (Not Covered)
  - Continuous variables and hybrid (discrete/continuous) BBNs
  - Induction of hidden variables
  - Local structure: localized constraints and assumptions, e.g., Noisy-OR BBNs
  - Online learning
    - Incrementality (*aka* lifelong, situated, *in vivo* learning)
    - Ability to change network structure during inferential process
  - Structural EM
  - Polytree structure learning (tree decomposition): alternatives to Chow-Liu MWST
  - Hybrid quantitative and qualitative Inference ("simulation")
  - Complexity of learning, inference in restricted classes of BBNs
- Topics to Be Covered Later
  - Decision theoretic models: decision networks *aka* influence diagrams (briefly)
  - Control and prediction models: POMDPs (for reinforcement learning)
  - *Some* temporal models: Dynamic Bayesian Networks (DBNs)

CIS 798: Intelligent Systems and Machine Learning
Kansas State University
Department of Computing and Information Sciences

Terminology

- Bayesian Networks: Quick Review on Learning, Inference
  - Structure learning: determining the best topology for a graphical model from data
    - Constraint-based methods
    - Score-based methods: statistical or information-theoretic degree of match
    - Both can be global or local, exact or approximate
  - Elicitation of subjective probabilities
- Causal Modeling
  - Causality: "direction" from cause to effect among events (observable or not)
  - Causal discovery: learning causality from observations
- Incomplete Data: Learning and Inference
  - Missing values: to be filled in given partial observations
  - Expectation-Maximization (EM): iterative refinement clustering algorithm
    - Estimation step: use current parameters Θ to estimate missing $\{N_i\}$
    - Maximization (re-estimation) step: update Θ to maximize $P(N_i, E_j \mid D)$

CIS 798: Intelligent Systems and Machine Learning
Kansas State University
Department of Computing and Information Sciences

Summary Points

- Bayesian Networks: Quick Review on Learning, Inference
  - Learning, eliciting, applying CPTs
  - In-class exercise: *Hugin* demo; CPT elicitation, application
  - Learning BBN structure: constraint-based versus score-based approaches
  - *K2*, other scores and search algorithms
- Causal Modeling and Discovery: Learning Causality from Observations
- Incomplete Data: Learning and Inference (Expectation-Maximization)
- Tutorials on Bayesian Networks
  - Breese and Koller (AAAI '97, BBN intro): http://robotics.Stanford.EDU/~koller
  - Friedman and Goldszmidt (AAAI '98, Learning BBNs from Data): http://robotics.Stanford.EDU/people/nir/tutorial/
  - Heckerman (various UAI/IJCAI/ICML 1996-1999, Learning BBNs from Data): http://www.research.microsoft.com/~heckerman
- Next Week: BBNs Concluded; Review for Midterm (10/14/1999)
- After Midterm: More EM, Clustering, Exploratory Data Analysis

CIS 798: Intelligent Systems and Machine Learning
Kansas State University
Department of Computing and Information Sciences