Structural EM – Learning Bayesian Networks and Parameters from Incomplete Data

Dan Li
University of Pittsburgh
Nov 16, 2005

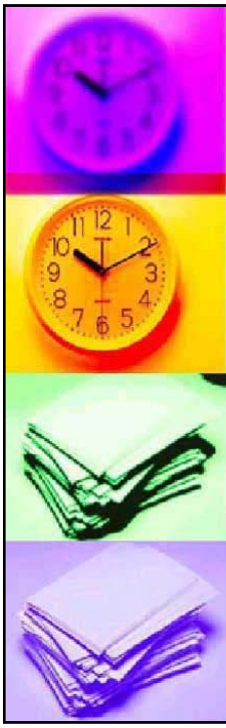
Papers

- Paper 1:
  - The Bayesian Structural EM Algorithm – by Nir Friedman
- Paper 2:
  - Learning Bayesian Networks from Incomplete Data – by Moninder Singh

The General Problem

- Learn the parameters for a fixed network with complete data
- Learn the parameters for a fixed network with incomplete data
- Learn both parameters and even the network structure from incomplete data – in the presence of missing values or hidden variables

Paper 1: The Bayesian Structural EM Algorithm

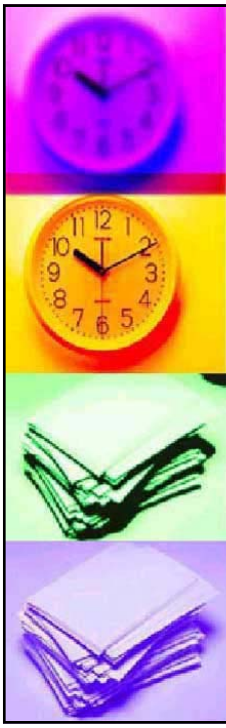


The Structural EM Algorithm

- In the previous paper:
  - Combines Standard EM algorithm which optimizes parameters and Structure search for model selection
  - Using penalized likelihood scores which includes BIC/MDL and various approximations to the Bayesian score
- In this paper, extended structural EM to deal with the Bayesian model selection

Introduction

- Current methods are successful at learning both the structure and the parameters from complete data
- Things are different when the data is incomplete
- It is unreasonable to require complete data to train the network while allowing inference based on incomplete data



Introduction

- The key idea in structural EM:
  - Use the best estimate of the distribution to complete the data and use procedures that work efficiently for complete data on this completed data.
  - Performs search in the joint space of (structure X parameters) for the best structure
  - In each step, it either find better parameters for the current structure or find a new structure

Preliminaries

- Factored Model

A *factored* model M (for $\mathbf{U} = \{X_1, \ldots, X_n\}$) is a parametric family with parameters $\Theta^M = \langle \Theta_1^M, \ldots, \Theta_k^M \rangle$ that defines a joint probability measure of the form:

$$\Pr(X_1, \ldots, X_n \mid M^h, \Theta^M) = \prod_i f_i^M(X_1, \ldots, X_n : \Theta_i^M),$$

where each f_i^M is a *factor* whose value depends on some (or all) of the variables $X_1, \ldots, X_n$. A factored model is *separable* if the space of legal choices of parameters is the cross product of the legal choices of parameters Θ_i^M for each f_i^M. In other words, if legal parameterization of different factors can be combined without restrictions.

Bayesian Learning

- Bayesian Learning attempts to make predictions by conditioning the prior on the observed data.
- The prediction of the probability of an event X after seeing the training data, can be written as:

$$
\begin{aligned}
\Pr(X \mid D) &= \sum_M \Pr(X \mid M^h, D) \Pr(M^h \mid D) \\
&= \sum_M \Pr(X \mid M^h, D) \frac{\Pr(D \mid M^h) \Pr(M^h)}{\Pr(D)}
\end{aligned}
$$

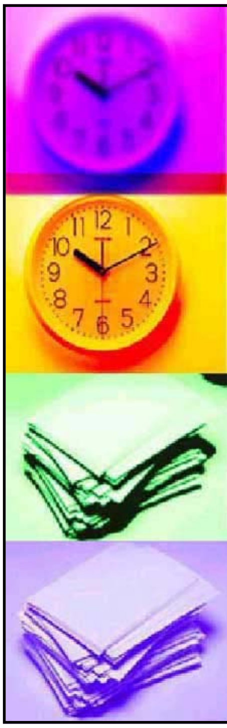
Bayesian Learning

- Where

$$\Pr(D \mid M^h) = \int \Pr(D \mid M^h, \Theta) \Pr(\Theta \mid M^h) d\Theta, \quad (2)$$

$$\Pr(X \mid M^h, D) = \int \Pr(X \mid M^h, \Theta) \Pr(\Theta \mid M^h, D) d\Theta. \quad (3)$$

- We can not afford to sum over all possible models
  - MAP model
  - Sum over models with highest posterior probabilities



Assumptions

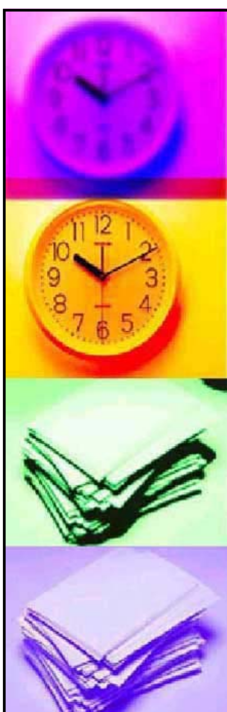
Assumption 1. All the models $\mathcal{M}$ are separable factored models.

Assumption 2. All the models in $\mathcal{M}$ contain only exponential factors.

Assumption 3. For each model $M \in \mathcal{M}$ with k factors the prior distribution over parameters has the form

$$\Pr(\Theta_1^M, \ldots, \Theta_k^M \mid M^h) = \prod_i \Pr(\Theta_i^M \mid M^h).$$

Assumption 4. If $f_i^M = f_j^{M'}$ for some $M, M' \in \mathcal{M}$, then $\Pr(\Theta_i^M \mid M^h) = \Pr(\Theta_j^{M'} \mid M'^h)$.



Exponential Representation

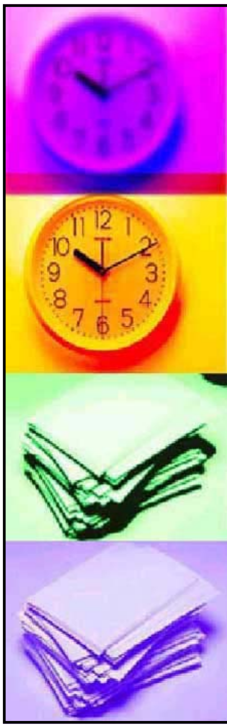
Proposition 2.4: *Given Assumptions 1–4 and a data set $D = \{\mathbf{u}^1, \ldots, \mathbf{u}^N\}$ of complete assignments to $\mathbf{U}$, the score of a model M that consists of k factors $f_1, \ldots, f_k$, is*

$$\Pr(D \mid M^h) = \prod_{i=1}^{k} F_i\left(\sum_{j=1}^{N} s_i(\mathbf{u}^j)\right),$$

where

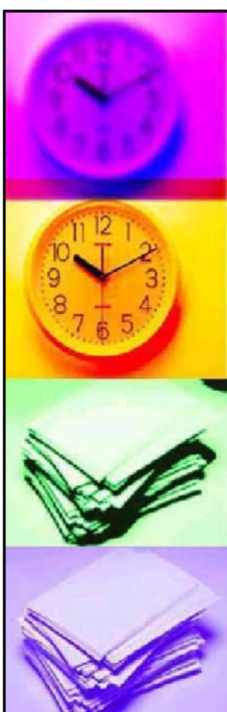
$$F_i(S) = \int e^{t_i(\Theta_i)\cdot S} \Pr(\Theta_i) d\Theta_i,$$

and $t_i(\cdot)$, and $s_i(\cdot)$ are the the exponential representation of f_i.



Prior

- In practice, it is useful to require that the prior for each factor is a conjugate prior.
- For many types of exponential distributions, the conjugate priors lead to a close-form solution for the posterior beliefs and for the probability of the data.



Dirichlet Prior

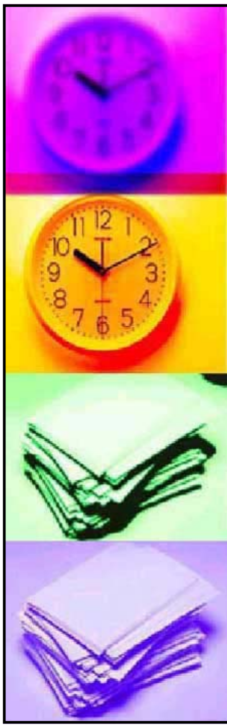
Example 2.5: We now complete the description of the learning problem of multinomial belief networks. Following [9, 17] we use *Dirichlet priors*. A Dirichlet prior for a multinomial distribution of a variable X is specified by a set of *hyperparameters* $\{N'_{v_1}, \ldots, N'_{v_l}\}$ where $v_1, \ldots, v_l$ are the values of X. We say that

$$\Pr(\Theta) \sim \text{Dirichlet}(\{N'_{v_1}, \ldots, N'_{v_l}\}) \text{ if } \Pr(\Theta) \propto \prod_{v_i} \theta_{v_i}^{N'_{v_i}-1}.$$

For a Dirichlet prior with parameters $N'_{v_1}, \ldots, N'_{v_k}$ the probability of the values of X with sufficient statistics $S = \langle N_{v_1}, \ldots, N_{v_k} \rangle$ is given by

$$F(S) = \frac{\Gamma(\sum_i N'_{v_i})}{\Gamma(\sum_i (N'_{v_i} + N_{v_i}))} \prod_i \frac{\Gamma(N'_{v_i} + N_{v_i})}{\Gamma(N'_{v_i})}, \qquad (4)$$

where $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$ is the *Gamma* function. For more details on Dirichlet priors, see [10].



Learning From complete data

- Learning factored models from data is done by searching over the space of models for a model that maximize the score
- By changing the factored model locally, the score of the new model differs from the score of the old model by only a few terms
- By caching accumulated sufficient statistics for various factors, various combination of different factors can be evaluated efficiently

Modifying the model

- Operations:
  - Arc Additions
  - Arc Removals
  - Arc Reversals
- Complexity
  - $O(n^2)$ neighbors at each step
  - O(n) re-evaluations

Learning from incomplete data

- Harder than that for complete data
  - The posterior is no longer product of independent terms
  - The probability of data is no longer product of terms
  - The model can not be represented with closed form
  - Can not make exact prediction give a model using the integral of (3)

Learning from Incomplete data

- Harder than learning from complete data
  - Since the probability of the data given a model no longer decomposes, direct estimate the integral of (2) is needed.
  - Approximating the integral
    - If the posterior over parameters is sharply peaked, the integral in (3) is dominated by the prediction in a small region around the posterior' peak, so that

$$\Pr(X \mid M^h, D) \approx \Pr(X \mid M^h, \hat{\Theta})$$

Learning from Incomplete data

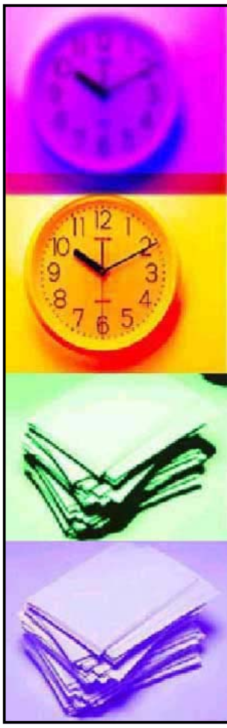
- Estimate the integral

$$\Pr(D \mid M^h) = \int \Pr(D \mid M^h, \Theta) \Pr(\Theta \mid M^h) d\Theta$$

- Stochastic Simulation
- Large-sample approximation

The structural EM Algorithm

- Directly optimize the Bayesian score rather than asymptotic approximation



The Structural EM

- A class of models M that each model is parameterized by a vector Θ^M such that each choice of values Θ^M defines a probability distribution Pr(·:, M, Θ^M)
- Assuming prior over models and parameter assignment in each model
- Maximize

$$\Pr(M^h \mid D) = \frac{\Pr(D \mid M^h)\Pr(M^h)}{\Pr(D)}$$

Pr(D) is the probability over all models, which is the same for all the models, so maximize the nominator is enough

The structural EM

- With missing data in D, evaluating $\Pr(D|M^h)$ is not easy
- Assuming the evaluation of $\Pr(H,O|M^h)$ is possible
  - True for models satisfying assumption 1 – 4

The structural EM Algorithm

Procedure Bayesian-SEM($M_0, \mathbf{o}$):
  Loop for $n = 0, 1, \ldots$ until convergence
    Compute the posterior $\Pr(\Theta^{M_n} \mid M_n^h, \mathbf{o})$.
    E-step: For each M, compute

$$Q(M : M_n) = E[\log \Pr(\mathbf{H}, \mathbf{o}, M^h) \mid M_n^h, \mathbf{o}] = \sum_{\mathbf{h}} \Pr(\mathbf{h} \mid \mathbf{o}, M_n^h) \log \Pr(\mathbf{h}, \mathbf{o}, M^h)$$

    M-step Choose M_{n+1} that maximizes $Q(M : M_n)$
    if $Q(M_n : M_n) = Q(M_{n+1} : M_n)$ then
      return M_n

The Structural EM

- At each iteration, the algorithm attempts to maximize the expected score of models instead of their actual score
  - Why is this easier?
    - Depends on the class of model
  - What does this buy us?
    - The evaluation is efficient

Theorem 3.1

Theorem 3.1: *Let $M_0, M_1, \ldots$ be the sequence of models examined by the Bayesian SEM procedure. Then,*

$$\begin{aligned}&\log \Pr(\mathbf{o}, M^h_{n+1}) - \log \Pr(\mathbf{o}, M^h_n)\\ &\quad \geq \; Q(M_{n+1} : M_n) - Q(M_n : M_n)\end{aligned}$$

Proof:

$$\begin{aligned}
&\log \frac{\Pr(\mathbf{o}, M^h_{n+1})}{\Pr(\mathbf{o}, M^h_n)} \\
&= \log \sum\nolimits_{\mathbf{h}} \frac{\Pr(\mathbf{h}, \mathbf{o}, M^h_{n+1})}{\Pr(\mathbf{o}, M^h_n)} \cdot \frac{\Pr(\mathbf{h} | \mathbf{o}, M^h_n)}{\Pr(\mathbf{h} | \mathbf{o}, M^h_n)} \\
&= \log \sum\nolimits_{\mathbf{h}} \Pr(\mathbf{h} \mid \mathbf{o}, M^h_n) \frac{\Pr(\mathbf{h}, \mathbf{o}, M^h_{n+1})}{\Pr(\mathbf{h}, \mathbf{o}, M^h_n)} \qquad (6) \\
&\geq \sum\nolimits_{\mathbf{h}} \Pr(\mathbf{h} \mid \mathbf{o}, M^h_n) \log \frac{\Pr(\mathbf{h}, \mathbf{o}, M^h_{n+1})}{\Pr(\mathbf{h}, \mathbf{o}, M^h_n)} \qquad (7) \\
&= E[\log \frac{\Pr(\mathbf{H}, \mathbf{o}, M^h_{n+1})}{\Pr(\mathbf{H}, \mathbf{o}, M^h_n)} \mid M^h_n, \mathbf{o}] \\
&= Q(M_{n+1} : M_n) - Q(M_n : M_n)
\end{aligned}$$

where all the transformations are by algebraic manipulations, and the inequality between (6) and (7) is a consequence of Jensen's inequality.[3] ∎

A weaker algorithm

- M*-step
  - Choose M_{n+1} such that
    $Q(M_{n+1} : M_n) > Q(M_n : M_n)$

Theorem 3.2

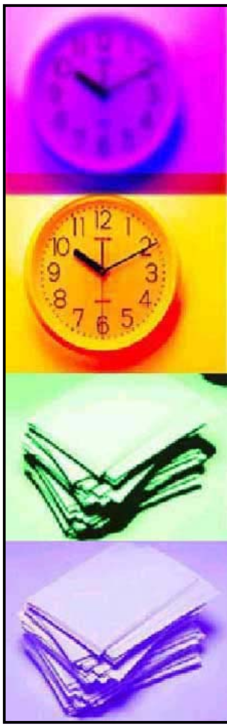
Theorem 3.2: *Let $M_0, M_1, \ldots$ be the sequence of models examined by the Bayesian SEM procedure. If the number of models in $\mathcal{M}$ is finite, or if there is a constant c such that $\Pr(D \mid M^h, \Theta^M) < c$ for all models M and parameters Θ^M, then the limit $\lim_{n \to \infty} \Pr(\mathbf{o}, M_n^h)$ exists.*

Bayesian Structural EM for factored models

Proposition 4.1: *Let $D = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ be a training set that consist of incomplete assignments to $\mathbf{U}$. Given Assumptions 1–4, if M consists of k factors, $f_1, \ldots, f_k$, then*

$$E[\log \Pr(\mathbf{H}, \mathbf{o} \mid M^h)] = \sum_{i=1}^{k} E[\log F_i(S_i)],$$

where $S_i = \sum_{j=1}^{N} s_i(\mathbf{U}^j)$ is a random variable that represents the accumulated sufficient statistics for the factor f_i in possible completions of the data.



Bayesian Structural EM for factored models

- Evaluating

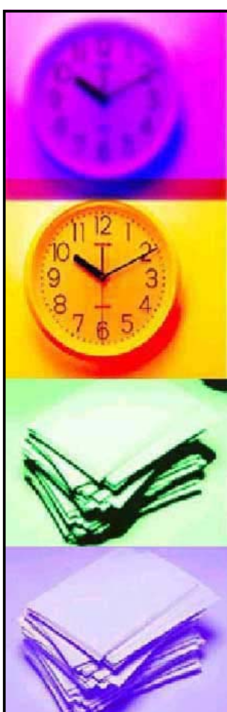
$$E[\log F_i(S_i)]$$

- Simple approximation

$$E[\log F_i(S_i)] \approx \log F_i(E[S_i])$$

- Computing probability over assignments **H**
  - **Use MAP approximation**

$$\Pr(X \mid M^h, D) \approx \Pr(X \mid M^h, \hat{\Theta})$$



Bayesian Structural EM for factored models

Procedure Factored-Bayesian-SEM(M_0, $\mathbf{o}$):
  Loop for $n = 0, 1, \ldots$ until convergence
    Compute the MAP parameters $\hat{\Theta}^{M_n}$ for M_n given $\mathbf{o}$.
    Perform search over models, evaluating each model by
      $Score(M : M_n) = \sum_i E[\log F_i^M(S_i^M) \mid \mathbf{o}, M_n^h, \hat{\Theta}_n^M]$
    Let M_{n+1} be the model with the highest score among
      these encountered during the search.
    if $Score(M_n : M_n) = Score(M_{n+1} : M_n)$ then
      return M_n

Computing E[logF(S)]

- Linear approximation

$$\log F(S) = \log F(E[S]) + (S - E[S])\nabla(\log F)(E[S]) + \frac{1}{2}(S - E[S])^T \nabla^2(\log F)(S^*)(S - E[S])$$

- Gaussian approximation

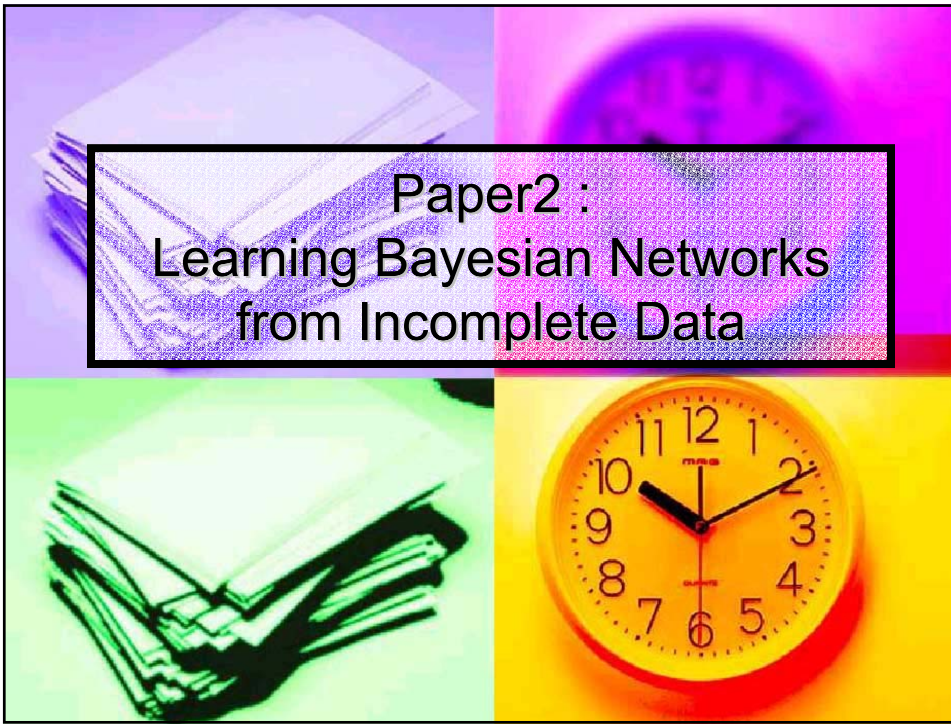
$$E[\log F(S)] \approx \int \log F(S)\varphi(S : E[S], \Sigma[S])dS$$

E[logF(S)] on Dirichlet Prior

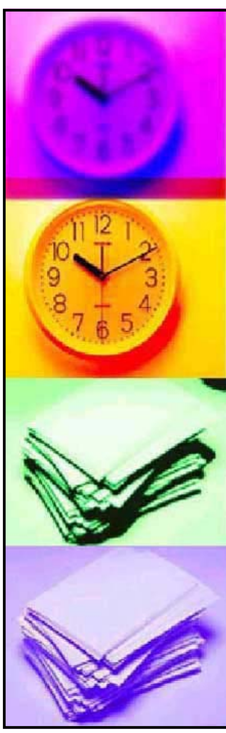
$$\begin{aligned}\log F(\langle N_{v_1}, \ldots, N_{v_l}\rangle) &= \log\Gamma(\textstyle\sum_i N'_{v_i}) - \log\Gamma(\sum_i (N'_{v_i} + N(v_i))) \\ &\quad + \textstyle\sum_i (\log\Gamma(N'_{v_i} + N(v_i)) - \log\Gamma(N'_{v_i}))\end{aligned}$$

$$\begin{aligned}E[\log F(\langle N_{v_1}, \ldots, N_{v_l}\rangle)] &= \textstyle\sum_i E[\log\Gamma(N'_{v_i} + N(v_i))] - \\ &\quad E[\log\Gamma(\textstyle\sum_i (N'_{v_i} + N(v_i)))] + c\end{aligned}$$

See the paper for details

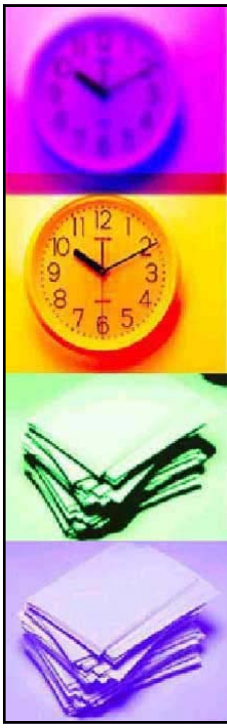


Paper2 :
Learning Bayesian Networks from Incomplete Data



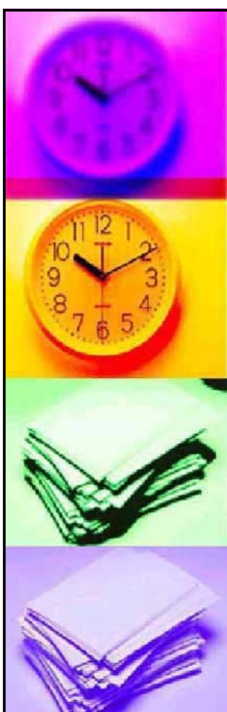
Introduction

- Learning both the structure and the parameters
- Using combination of EM and Imputation techniques



Missing Data

- MCAR
- MAR
- NMAR



Methods for handling missing data

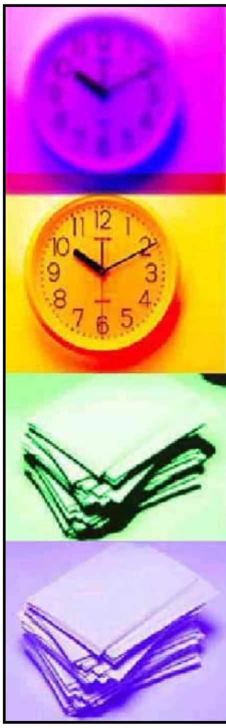
- Using only fully-observed cases
- Assign to each missing value a new value
- Replacing each missing value by a single value
- Replacing each missing value by the mean of observed values
- Multiple imputation method
- Sum over all possible values for each missing data point while calculating the required parameters
- EM and Gibbs sampling

The Algorithm

- Combination of EM and Imputation to interactively refine the structure
  - Use current estimate of the structure and the incomplete data to refine the conditional probabilities
  - Impute new values for missing data points by sampling from the new estimate of the conditional probabilities
  - Refines the structure from new estimate of the data using standard algorithms for learning Bayesian network from complete data

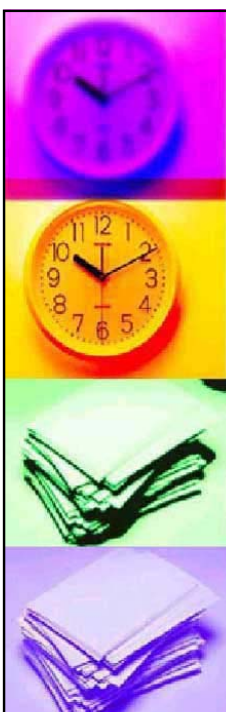
Imputation

- Missing data can be imputed to values drawn from the estimated conditional probability distributions



The Algorithm

1. Create M complete - datasets, $\hat{D}_s^{(0)}, 1 \le s \le M$, by sampling M values for each missing value from the prior distributi on of each sttribute
2. For s := 1 to M do
   2a. From the compete - dataset $\hat{D}_s^{(t)}$, induce the Bayesian network structure, $\hat{B}_s^{(t)}$, that has the maximum posterior probabilit y given the data, i.e. maximizes $P(B_s|\hat{D}_s^{(t)})$
   2b. Use the EM algorithm to learn the conditiona l probabilie s $\hat{\theta}_s^{(t)}$, using the original incomplete data D and the network structure $\hat{B}_s^{(t)}$ the graph union of all the resultant structures .



The Algorithm

3. Fuse the networks to create a single Bayesian network $< \hat{B}_s^{(t)}, \theta^{(t)} >$ as follows. Construct the network structure $B^{(t)}$ by taking the arc - union of the individual, network structures. i.e. $B^{(t)} = \bigcup_{s=1 \cdots M} B^{(t)}$. If the orderings imposed on the attributes by the various network structures are not consistent, then it is possible to construct $B^{(t)}$ by choosing one of the orderings(e.g. a total ordering consistent with the network structure with the maximum posterior probability), making all the other network structures consistent with this ordering by performing necessary arc - reversals, and then taking the graph union of all the resultant structures.

The Algorithm

4. If the convergence criteria is achieved, stop. Else go to step 5
5. Create M new complete datasets $\hat{D}_s^{(t+1)}$ by sampling M values for each missing values by sampling from the distribution obtained from last step

Question?

Any question?

Thank you!