Readings: K&F 17.4, 18.1, 18.2, 18.3

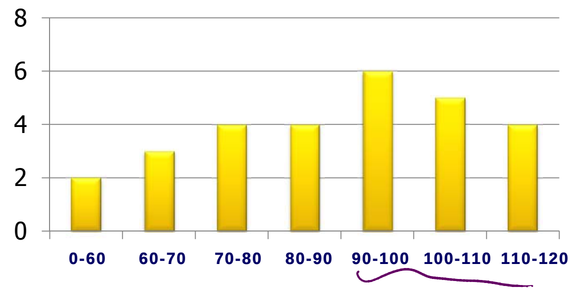
Parameter Estimation & Structure Learning

Lecture 10 – Apr 27, 2011
CSE 515, Statistical Methods, Spring 2011

Instructor: Su-In Lee
University of Washington, Seattle

Announcements

- Problem Set #1 has been graded.
  - Assuming Gaussian, sufficient statistics:
    Mean: 89.17; Std: 19.86
- Graded HW will be handed back after class.



Bayesian Approach: General Formulation

- Joint distribution over D,θ $P(D,\theta) = P(D \mid \theta)P(\theta)$

  $\alpha_h = \alpha_t$

  - As we saw, likelihood can be described compactly using sufficient statistics

- Posterior distribution over parameters

$$P(\theta \mid D) = \frac{P(D \mid \theta)P(\theta)}{P(D)}$$

- P(D) is the marginal likelihood of the data

$$P(D) = \int_{\theta} P(D \mid \theta)P(\theta)d\theta \quad \} \quad p(\theta) \qquad p(\theta \mid D).$$

- We want conditions in which posterior is also compact

Conjugate Families

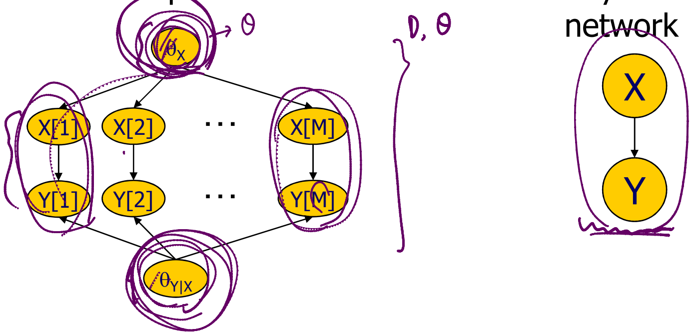
- A family of priors P(θ:α) is conjugate to a model P(ξ|θ) if for any possible dataset D of i.i.d samples from P(ξ|θ) and choice of hyperparameters α for the prior over θ, there are hyperparameters α' that describe the posterior, i.e.,
  $P(\theta{:}\alpha') \propto P(D \mid \theta)P(\theta{:}\alpha)$
  - Posterior has the same parametric form as the prior
  - Dirichlet prior is a conjugate family for the multinomial likelihood

- Conjugate families are useful since:
  - Many distributions can be represented with hyperparameters
  - They allow for sequential update within the same representation
  - In many cases we have closed-form solutions for prediction

Bayesian Estimation in BayesNets

Bayesian network for parameter estimation

Bayesian network

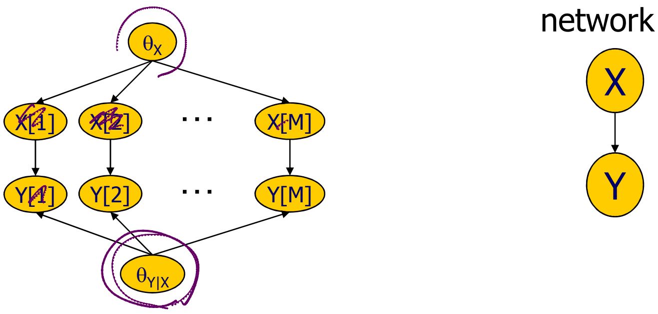


- Instances are independent given the parameters
  - (x[m'],y[m']) are d-separated from (x[m],y[m]) given θ
- Priors for individual variables are a priori independent
  - Global independence of parameters $P(\theta)=\prod_i P(\theta_{X_i|Pa(X_i)})$

Bayesian Estimation in BayesNets

Bayesian network for parameter estimation

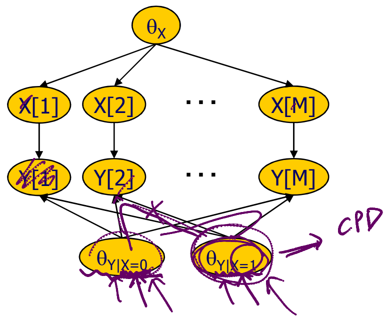
Bayesian network



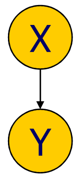
- Posteriors of θ are independent given complete data
  - Complete data d-separates parameters for different CPDs
  - $P(\theta_X,\theta_{Y|X}\,|\,D)=P(\theta_X\,|\,D)P(\theta_{Y|X}\,|\,D)$
  - As in MLE, we can solve each estimation problem separately

Bayesian Estimation in BayesNets

Bayesian network for parameter estimation



Bayesian network

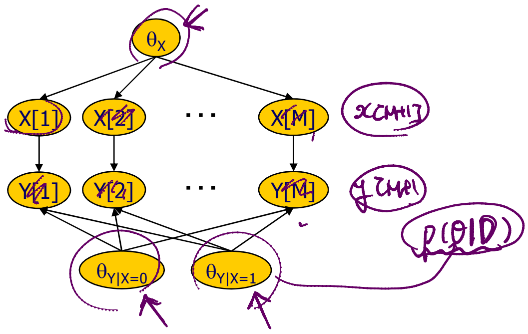


- Posteriors of θ are independent given complete data
  - Also holds for parameters within families
  - Note context specific independence between $\theta_{Y|X=0}$ and $\theta_{Y|X=1}$ when given both X and Y

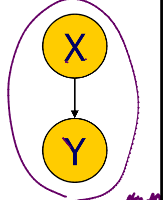
Bayesian Estimation in BayesNets

Bayesian network for parameter estimation

X	*Y*	
	y^0	y^1
x^0	$\theta_{y0|x0}$	$\theta_{y1|x0}$
x^1	$\theta_{y0|x1}$	$\theta_{y1|x0}$



Bayesian network



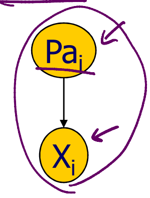
D: M instances
counts

- Posteriors of θ can be computed independently
  - For multinomial $\theta_{X_i|pa_i}$, posterior is Dirichlet with parameters $\alpha_{X_i=1|pa_i}+M[X_i=1|pa_i],\ldots, \alpha_{X_i=k|pa_i}+M[X_i=k|pa_i]$
  - $P(X_i[M+1]=x_i \mid Pa_i[M+1]=pa_i, D) = \dfrac{\alpha_{x_i|pa_i} + M[x_i, pa_i]}{\sum_i \alpha_{x_i|pa_i} + M[x_i, pa_i]}$

Parameter Estimation Summary

D : ⟨Pa_i, X_i⟩

- Estimation relies on sufficient statistics
  - For multinomials these are of the form $M[x_i, pa_i]$
    X_i Pa_i
  - Parameter estimation



$$\hat{\theta}_{x_i|pa_i} = \frac{M[x_i, pa_i]}{M[pa_i]} \qquad P(x_i \mid pa_i, D) = \frac{\alpha_{x_i,pa_i} + M[x_i, pa_i]}{\alpha_{pa_i} + M[pa_i]}$$

MLE Bayesian (Dirichlet)

D M

Pa	X: x^0	X: x^1
p^0	$\theta_{x0\|pa0}$	$\theta_{x1\|pa0}$
p^1	$\theta_{x0\|pa1}$	$\theta_{x1\|pa1}$

(μ, σ^2)

- Bayesian methods also require choice of priors
- MLE and Bayesian are asymptotically equivalent
- Both can be implemented in an online manner by accumulating sufficient statistics

$P(\theta)$

$P(\theta|D)$

Assessing Priors for BayesNets

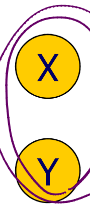
- We need the $\alpha(x_i, pa_i)$ for each node x_i

X

- We can use initial parameters Θ_0 as prior information
  - Need also an equivalent sample size parameter M'
  - Then, we let $\alpha(x_i,pa_i) = M' \cdot P(x_i,pa_i|\Theta_0)$
- This allows to update a network using new data

- Example network for priors
  - $P(X=0)=P(X=1)=0.5$
  - $P(Y=0)=P(Y=1)=0.5$
  - M'=1
  - Note: $\alpha(x_0)=0.5$ $\alpha(x_0,y_0)=0.25$

0.5×0.5

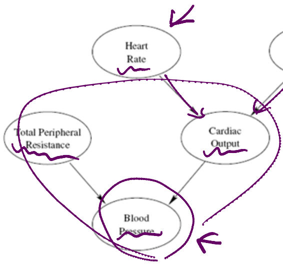


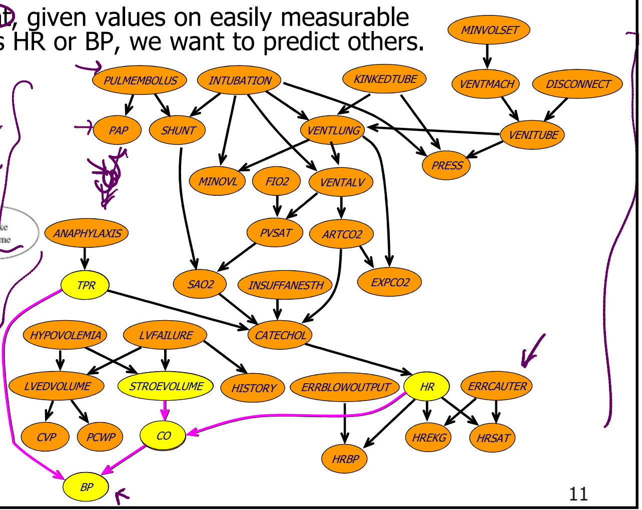
θ_0.

Case Study: ICU Alarm Network

- The "Alarm" network
  - Hand-constructed by experts: 37 variables; 504 parameters
- Predicting patient status in ICU
  - For a new patient, given values on easily measurable variables such as HR or BP, we want to predict others.

D M M-1 θ





Roberts et al. Bayesian networks for cardiovascular monitoring
http://lcp.mit.edu/pdf/Roberts06.pdf

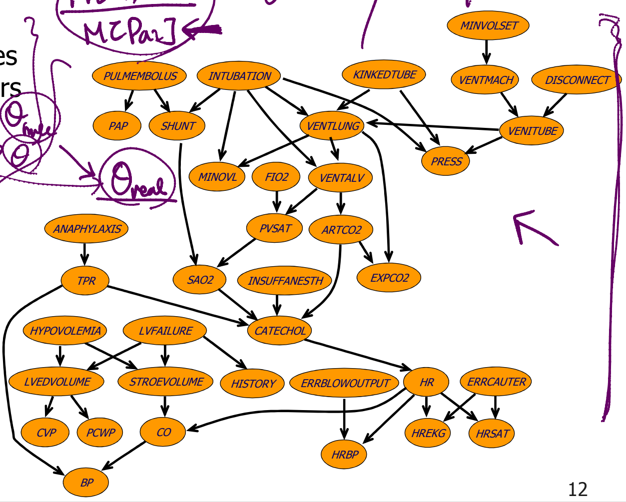
Case Study: ICU Alarm Network

- The "Alarm" network
  - Hand-constructed by experts
  - 37 variables; 504 parameters
- Experiment
  - Sample instances
  - Learn parameters
    - MLE
    - Bayesian

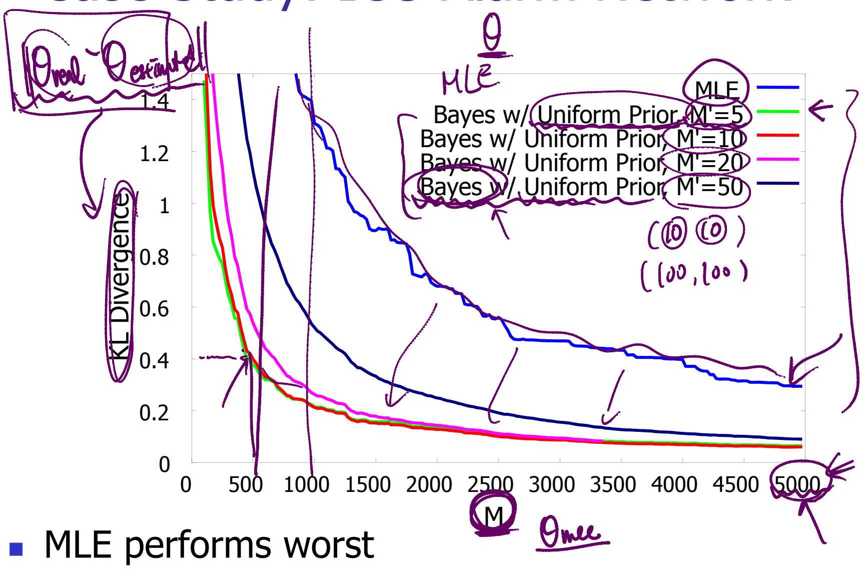
D M → θ_{mle}, θ_{Bayes}, θ_{true}

$M[X_i, Pa_i]$ / $M[Pa_i]$

θ_{mle} θ D θ_{real}



Case Study: ICU Alarm Network



- MLE performs worst
- Prior M'=5 provides best smoothing

STRUCTURE LEARNING

Structure Learning Motivation

- Network structure is often unknown
- Purposes of structure learning
  - Discover the dependency structure of the domain
    - Goes beyond statistical correlations between individual variables and detects direct vs. indirect correlations
    - Set expectations: at best, we can recover the structure up to the I-equivalence class
  - Density estimation
    - Estimate a statistical model of the underlying distribution and use it to reason with and predict new instances

Application in Artificial Intelligence

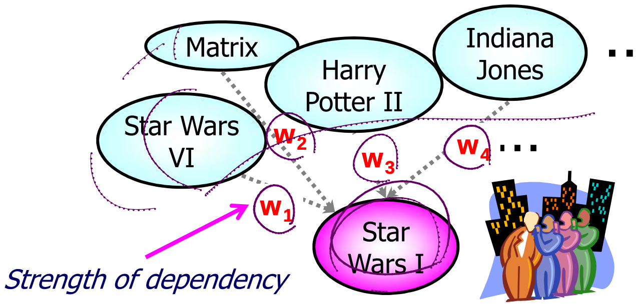
- Collaborative filtering: Predicting a user's preference on a certain product based on his or her preference on other products
  - For example: Netflix competition (movie rating prediction), amazon recommendation system …

Predict User rating of Star Wars I (task movie)

Given Ratings of other movies by the user (feature movies)

Training instances Many users

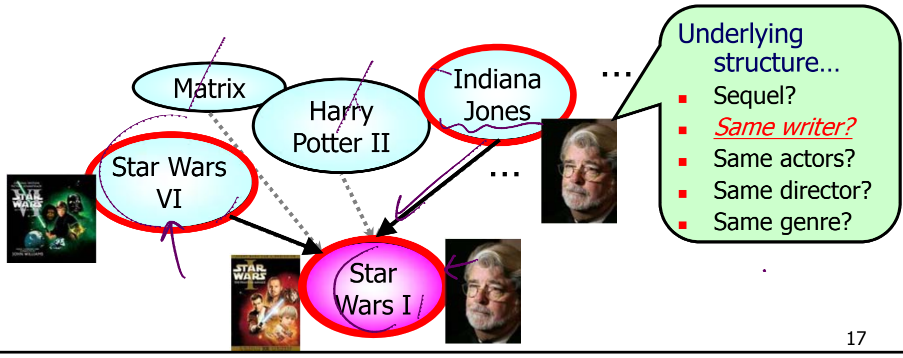
>110,000 movies in IMDB*
→ **Too many** parameters in the CPD



*Internet Movie Database

BayesNet Learning in Netflix Challenge

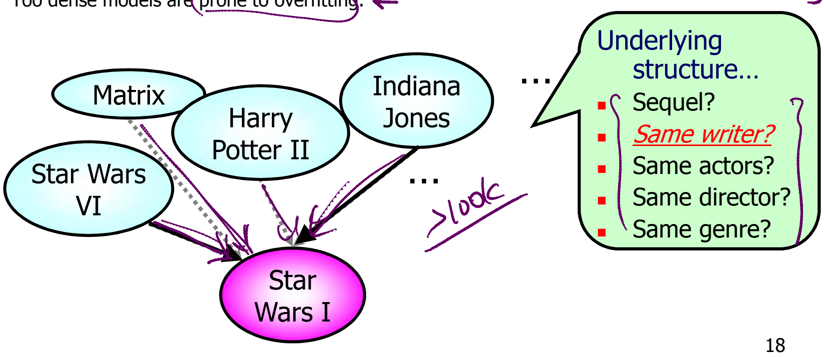
- Underlying structure should have a varying dependency between each feature movie and the task movie (Star Wars I)…



17

BayesNet Learning in Netflix Challenge

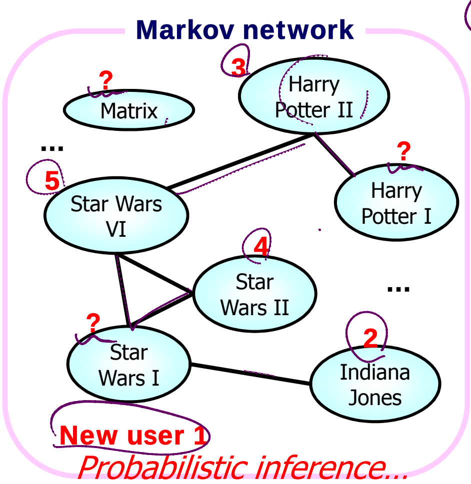
- Underlying structure should have a varying dependency between each feature movie and the task movie (Star Wars I)…
- Bayesian network
  - Variables: ratings on movies (can be partially observed)
  - Structure: prediction model (directed) or affinity (undirected)
  - Training data D <star_wars_I[m], matrix[m], harry_potter[m],…>: ratings on movies from M users (complete or partially observed)
- Structure learning:
  - We don't want to fix the structure based on our prior knowledge, but learn from the training data.
  - Too dense models are prone to overfitting.

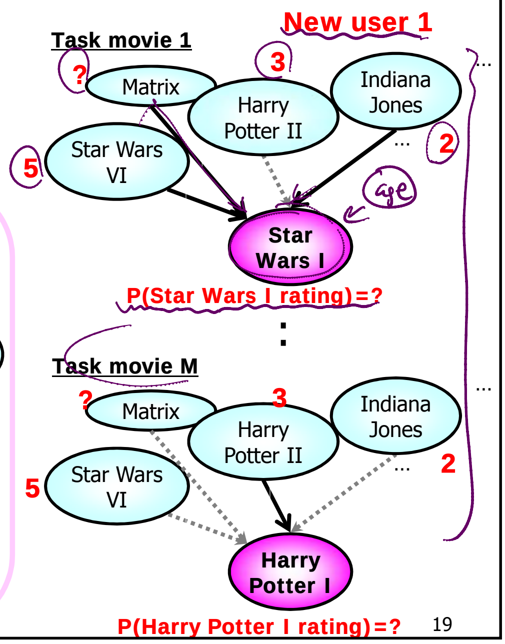


18

Predicting Ratings of New Users

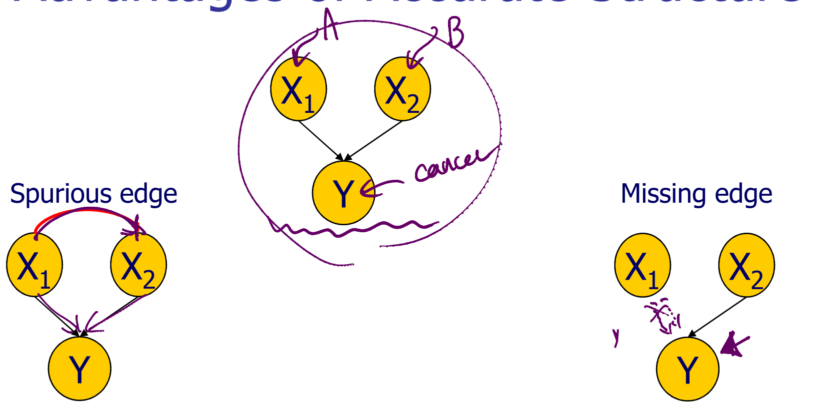
- Given a new user's ratings, predict ratings on task movies
  - MLE
  - Bayesian approach





19

Advantages of Accurate Structure



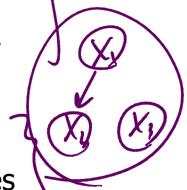
- Increases number of fitted parameters → overfitting
- Wrong causality and domain structure assumptions

- Cannot be compensated by parameter estimation
- Wrong causality and domain structure assumptions

20

Structure Learning Approaches

- Constraint based methods
  - View the Bayesian network as representing dependencies
  - Find a network that best explains dependencies ← P
  - Limitation: sensitive to errors in single dependencies $X_i \perp X_j$
- Score based approaches ⇦ *Today...*
  - View learning as a model selection problem
    - Define a scoring function specifying how well the model fits the data
    - Search for a high-scoring network structure
  - Limitation: super-exponential search space ←



- Bayesian model averaging methods
  - Average predictions across all possible structures
  - Can be done exactly (some cases) or approximately

21

Score Based Approaches

- Strategy
  - Define a scoring function for each candidate structure
  - Search for a high scoring structure
- Key: choice of scoring function
  - Likelihood based scores
  - Bayesian based scores

22

Likelihood Scores

- Goal: find (G,θ) that maximize the likelihood
  - $Score_L(G:D) = \log P(D \mid G, \theta'_G)$ where θ'_G is MLE for G
  - Find G that maximizes $Score_L(G:D)$

$\hat{\theta} = \arg\max_{\theta} P(D \mid \theta)$

$\arg\max_{G,\theta} P(D \mid G, \theta_G)$ -

$$\max_{G,\theta} P(D \mid G, \theta) = \max_{G} \left[\max_{\theta_G} P(D \mid G, \theta_G) \right]$$

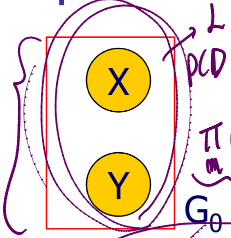
$$Score_L(G:D) = \log P(D \mid G, \hat{\theta}_G)$$

$P(D \mid G, \hat{\theta}_G)$

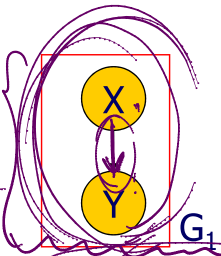
Example

L?

$P(D : G, \theta)$



$\prod_m \theta_{x[m]} \cdot \prod_m \theta_{y[m]}$ D $\{x[m], y[m]\}$



$$Score_L(G_0 : D) = \sum_m \log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]}$$

$$Score_L(G_1 : D) = \sum_m \log \hat{\theta}_{x[m]} + \log \hat{\theta}_{y[m]|x[m]}$$

$$\begin{aligned}
Score_L(G_1 : D) - Score_L(G_0 : D) &= \sum_m \log \hat{\theta}_{y[m]|x[m]} - \log \hat{\theta}_{y[m]} \\
&= \sum_{x,y} M[x,y] \log \hat{\theta}_{y|x} - \sum_y M[y] \log \hat{\theta}_y \\
&= M \sum_{x,y} \hat{P}(x,y) \log \hat{P}(y \mid x) - M \sum_y \hat{P}(y) \log \hat{P}(y) \\
&= M \sum_{x,y} \hat{P}(x,y) \log \hat{P}(y \mid x) - M \sum_{x,y} \hat{P}(x,y) \log \hat{P}(y) \\
&= M\, \mathbf{I}_{\hat{P}}(X,Y) \geq 0.
\end{aligned}$$

$M[x,y] = M \hat{P}(x,y)$

Information-theoretic interpretation:
High mutual information implies stronger dependency.
Stronger dependency implies stronger preference for the model where X and Y depend on each other.

General Decomposition

$Pa_{X_i} \to X_i$

- The Likelihood score decomposes as:

$$Score_L(G:D) = M\sum_{i=1}^{n}\mathbf{I}_{\hat{P}}(X_i, Pa_{X_i}^G) - M\sum_{i=1}^{n}\mathbf{H}_{\hat{P}}(X_i)$$

- Proof:

$$Score_L(G:D) = \sum_{i=1}^{n}\left[\sum_{u_i \in Val(Pa_{X_i}^G)}\sum_{x_i} M[x_i,u_i]\log\hat{\theta}_{x_i|u_i}\right]$$

$$\begin{aligned}
\frac{1}{M}\sum_{u_i}\sum_{x_i} M[x_i,u_i]\log\hat{\theta}_{x_i|u_i} &= \sum_{u_i}\sum_{x_i}\hat{P}(x_i,u_i)\log\hat{P}(x_i \mid u_i) \\
&= \sum_{u_i}\sum_{x_i}\hat{P}(x_i,u_i)\log\left(\frac{\hat{P}(x_i,u_i)\hat{P}(x_i)}{\hat{P}(u_i)\hat{P}(x_i)}\right) \\
&= \sum_{u_i}\sum_{x_i}\hat{P}(x_i,u_i)\log\left(\frac{\hat{P}(x_i,u_i)}{\hat{P}(u_i)\hat{P}(x_i)}\right) + \sum_{x_i}\left(\sum_{u_i}\hat{P}(x_i,u_i)\right)\log\hat{P}(x_i) \\
&= \mathbf{I}_{\hat{P}}(X_i,U_i) + \sum_{x_i}\hat{P}(x_i)\log\hat{P}(x_i) \\
&\mathbf{I}_{\hat{P}}(X_i,U_i) - \mathbf{H}_{\hat{P}}(X_i)
\end{aligned}$$

Information-theoretic interpretation:
High mutual information implies stronger dependency.
Stronger dependency implies stronger preference for the model where X and Y depend on each other.

General Decomposition

- The Likelihood score decomposes as:

X_i Pa_{X_i}

$$Score_L(G:D) = M\sum_{i=1}^{n}\mathbf{I}_{\hat{P}}(X_i, Pa_{X_i}^G) - M\sum_{i=1}^{n}\mathbf{H}_{\hat{P}}(X_i)$$

  - Second term does not depend on network structure and thus is irrelevant for selecting between two structures
  - Score increases as mutual information, or strength of dependence between connected variable increases

D

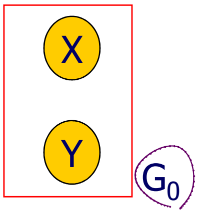
$\Downarrow$

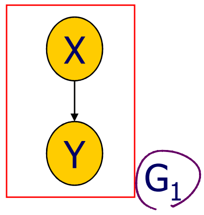
- After some manipulation can show:

$$Score_L(G:D) = \mathbf{H}_{\hat{P}}(X_1,\ldots,X_n) - \sum_{i=1}^{n}\mathbf{I}_{\hat{P}}(X_i, \{X_1,\ldots X_{i-1}\} - Pa_{X_i}^G \mid Pa_{X_i}^G)$$

  - These two interpretations are complementary, one is measuring the strength of dependence between and X and its parents, and the other is measuring the extent of the independence of X, from its predecessors given its parents.

Limitations of Likelihood Score





$$Score_L(G_1 : D) - Score_L(G_0 : D) = M \cdot \mathbf{I}_{\hat{P}}(X,Y)$$

- Since $I_P(X,Y) \geq 0 \rightarrow Score_L(G_1:D) \geq Score_L(G_0:D)$
- Adding arcs always helps
- Maximal scores attained for fully connected network
- Such networks overfit the data (i.e., fit the noise in the data)

27

Avoiding Overfitting

- Classical problem in machine learning
- Solutions
  - Restricting the hypotheses space
    - Limits the overfitting capability of the learner
    - Example: restrict # of parents or # of parameters
  - Minimum description length
    - Description length measures complexity
    - Prefer models that compactly describes the training data
  - Bayesian methods
    - Average over all possible parameter values
    - Use prior knowledge

28

Bayesian Score

- Main principle of the Bayesian approach
  - Whenever we have uncertainty over anything, we should place a distribution over it. What uncertainty? (G, Θ_G)

Marginal likelihood

Prior over structures

$$P(G \mid D) = \frac{P(D \mid G)P(G)}{P(D)}$$

Marginal probability of Data

P(D) does not depend on the network

Bayesian Score: $Score_B(G:D) = \log P(D \mid G) + \log P(G)$

29

Marginal Likelihood of Data Given G

Bayesian Score: $Score_B(G:D) = \log P(D \mid G) + \log P(G)$

Likelihood

Prior over parameters

Marginal likelihood

$$\boxed{P(D \mid G)} = \int_{\theta_G} P(D \mid G, \theta_G) P(\theta_G \mid G) d\theta_G$$

Note similarity to maximum likelihood score, but with the key difference that ML finds maximum of likelihood and here we compute average of the terms over parameter space

30