Learning Bayesian Networks from Data

Nir Friedman
Hebrew U.

Daphne Koller
Stanford

Overview

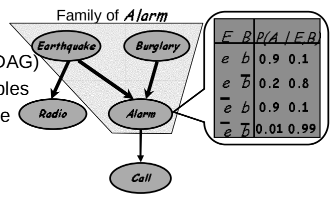
- **Introduction**
- Parameter Estimation
- Model Selection
- Structure Discovery
- Incomplete Data
- Learning from Structured Data

2

Bayesian Networks

Compact representation of probability distributions via conditional independence

Qualitative part:
Directed acyclic graph (DAG)
- Nodes - random variables
- Edges - direct influence



Together:
Define a unique distribution in a factored form

Quantitative part:
Set of conditional probability distributions

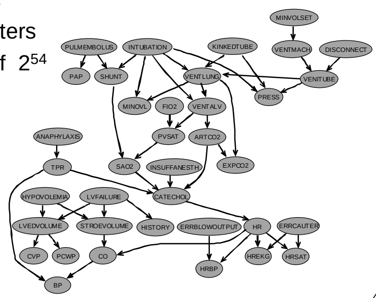
$$P(B,E,A,C,R) = P(B)P(E)P(A \mid B,E)P(R \mid E)P(C \mid A)$$

3

Example: "ICU Alarm" network

Domain: Monitoring Intensive-Care Patients
- 37 variables
- 509 parameters

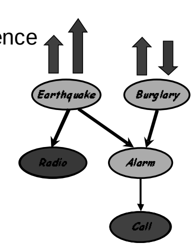
  …instead of 2^{54}



4

Inference

- **Posterior probabilities**
  - Probability of any event given any evidence
- Most likely explanation
  - Scenario that explains evidence
- Rational decision making
  - Maximize expected utility
  - Value of Information
- Effect of intervention



5

Why learning?

Knowledge acquisition bottleneck
- Knowledge acquisition is an expensive process
- Often we don't have an expert

Data is cheap
- Amount of available information growing rapidly
- Learning allows us to construct models from raw data

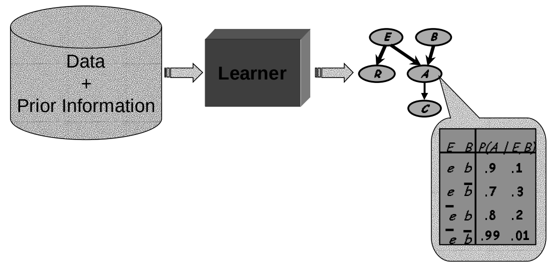
6

Why Learn Bayesian Networks?

- Conditional independencies & graphical language capture structure of many real-world distributions
- Graph structure provides much insight into domain
  - Allows "knowledge discovery"
- Learned model can be used for many tasks
- Supports all the features of probabilistic learning
  - Model selection criteria
  - Dealing with missing data & hidden variables

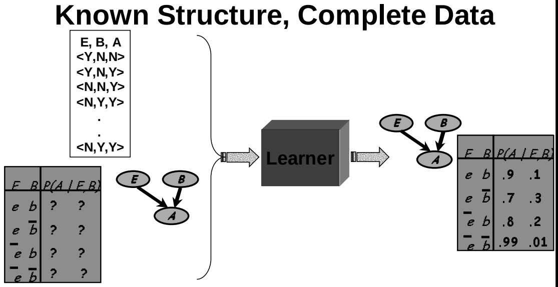
7

Learning Bayesian networks



8

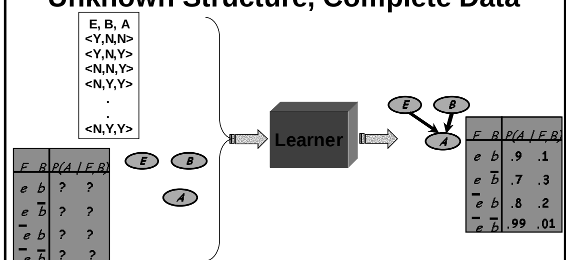
Known Structure, Complete Data



- Network structure is specified
  - Inducer needs to estimate parameters
- Data does not contain missing values

9

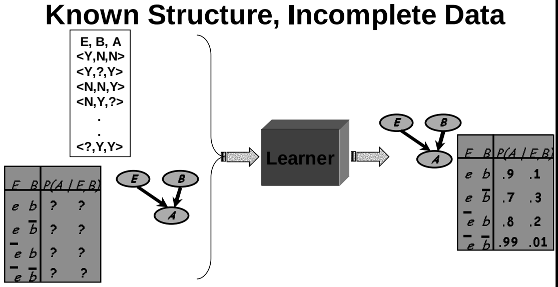
Unknown Structure, Complete Data



- Network structure is not specified
  - Inducer needs to select arcs & estimate parameters
- Data does not contain missing values

10

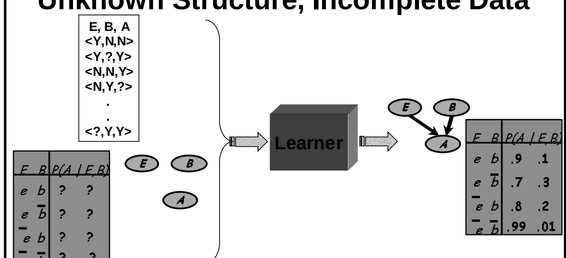
Known Structure, Incomplete Data



- Network structure is specified
- Data contains missing values
  - Need to consider assignments to missing values

11

Unknown Structure, Incomplete Data



- Network structure is not specified
- Data contains missing values
  - Need to consider assignments to missing values

12

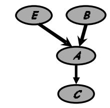
Overview

- Introduction
- **Parameter Estimation**
  - **Likelihood function**
  - **Bayesian estimation**
- Model Selection
- Structure Discovery
- Incomplete Data
- Learning from Structured Data

13

Learning Parameters

- Training data has the form:

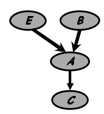


$$D = \begin{bmatrix} E[1] & B[1] & A[1] & C[1] \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ E[M] & B[M] & A[M] & C[M] \end{bmatrix}$$

14

Likelihood Function

- Assume i.i.d. samples
- Likelihood function is

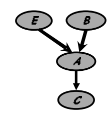


$$L(\Theta : D) = \prod_m P(E[m], B[m], A[m], C[m] : \Theta)$$

15

Likelihood Function

- By definition of network, we get



$$L(\Theta : D) = \prod_m P(E[m], B[m], A[m], C[m] : \Theta)$$

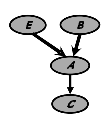
$$= \prod_m \begin{pmatrix} P(E[m] : \Theta) \\ P(B[m] : \Theta) \\ P(A[m] \mid B[m], E[m] : \Theta) \\ P(C[m] \mid A[m] : \Theta) \end{pmatrix}$$

$$\begin{bmatrix} E[1] & B[1] & A[1] & C[1] \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ E[M] & B[M] & A[M] & C[M] \end{bmatrix}$$

16

Likelihood Function

- Rewriting terms, we get



$$L(\Theta : D) = \prod_m P(E[m], B[m], A[m], C[m] : \Theta)$$

$$= \begin{matrix} \prod_m P(E[m] : \Theta) \\ \prod_m P(B[m] : \Theta) \\ \prod_m P(A[m] \mid B[m], E[m] : \Theta) \\ \prod_m P(C[m] \mid A[m] : \Theta) \end{matrix}$$

$$\begin{bmatrix} E[1] & B[1] & A[1] & C[1] \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ E[M] & B[M] & A[M] & C[M] \end{bmatrix}$$

17

General Bayesian Networks

Generalizing for any Bayesian network**:**

$$L(\Theta : D) = \prod_m P(x_1[m], \ldots, x_n[m] : \Theta)$$

$$= \prod_i \prod_m P(x_i[m] \mid Pa_i[m] : \Theta_i)$$

$$= \prod_i L_i(\Theta_i : D)$$

Decomposition

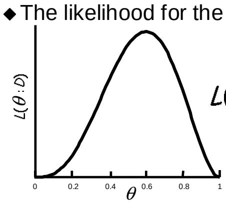
$\Rightarrow$ **Independent estimation problems**

18

Likelihood Function: Multinomials

$$L(\theta : D) = P(D \mid \theta) = \prod_m P(x[m] \mid \theta)$$

- The likelihood for the sequence H,T, T, H, H is



$$L(\theta : D) = \theta \cdot (1-\theta) \cdot (1-\theta) \cdot \theta \cdot \theta$$

General case: $L(\Theta : D) = \prod_{k=1}^{K} \theta_k^{N_k}$

Count of k^th outcome in D

Probability of k^th outcome

Bayesian Inference

- Represent uncertainty about parameters using a probability distribution over parameters, data
- Learning using Bayes rule

$$P(\theta \mid x[1], \ldots x[M]) = \frac{P(x[1], \ldots x[M] \mid \theta) P(\theta)}{P(x[1], \ldots x[M])}$$

Likelihood

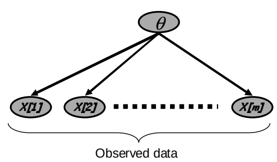
Prior

Posterior

Probability of data

Bayesian Inference

- Represent Bayesian distribution as Bayes net



- The values of X are independent given θ
- $P(x[m] \mid \theta) = \theta$
- Bayesian prediction is inference in this network

Example: Binomial Data

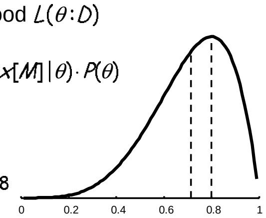
- Prior: uniform for θ in $[0,1]$

  $\Rightarrow$ $P(\theta \mid D) \propto$ the likelihood $L(\theta : D)$

$$P(\theta \mid x[1], \ldots x[M]) \propto P(x[1], \ldots x[M] \mid \theta) \cdot P(\theta)$$

$(N_H, N_T) = (4,1)$

- MLE for $P(X = H)$ is $4/5 = 0.8$
- Bayesian prediction is



$$P(x[M+1] = H \mid D) = \int \theta \cdot P(\theta \mid D) d\theta = \frac{5}{7} = 0.7142\ldots$$

Dirichlet Priors

- Recall that the likelihood function is

$$L(\Theta : D) = \prod_{k=1}^{K} \theta_k^{N_k}$$

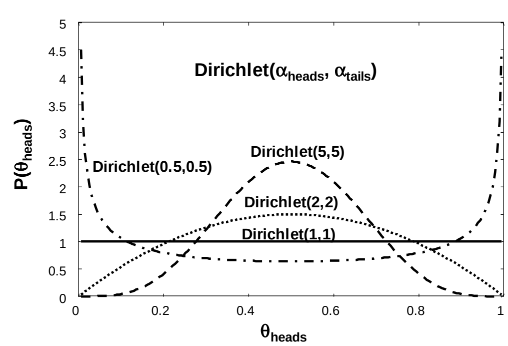
- **Dirichlet** prior with hyperparameters $\alpha_1, \ldots, \alpha_K$

$$P(\Theta) \propto \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

$\Rightarrow$ the posterior has the same form, with hyperparameters $\alpha_1 + N_1, \ldots, \alpha_K + N_K$

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

Dirichlet Priors - Example



Dirichlet Priors (cont.)

- If $P(\Theta)$ is Dirichlet with hyperparameters $\alpha_1,\ldots,\alpha_K$

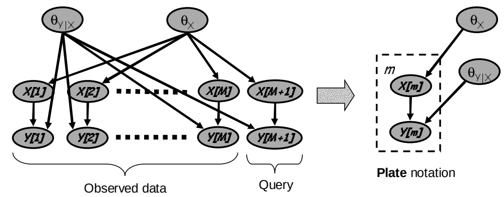
$$P(X[1]=k)=\int \theta_k \cdot P(\Theta)d\Theta = \frac{\alpha_k}{\sum_\ell \alpha_\ell}$$

- Since the posterior is also Dirichlet, we get

$$P(X[M+1]=k \mid D)=\int \theta_k \cdot P(\Theta \mid D)d\Theta = \frac{\alpha_k + N_k}{\sum_\ell (\alpha_\ell + N_\ell)}$$

25

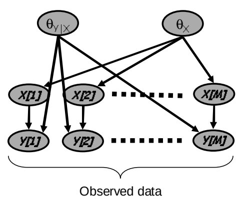
Bayesian Nets & Bayesian Prediction



- Priors for each parameter group are independent
- Data instances are independent given the unknown parameters

26

Bayesian Nets & Bayesian Prediction



- We can also "read" from the network:

  Complete data $\Rightarrow$

  posteriors on parameters are independent
- Can compute posterior over parameters separately!

27

Learning Parameters: Summary

- Estimation relies on **sufficient statistics**
  - For multinomials: counts $N(x_i, pa_i)$
  - Parameter estimation

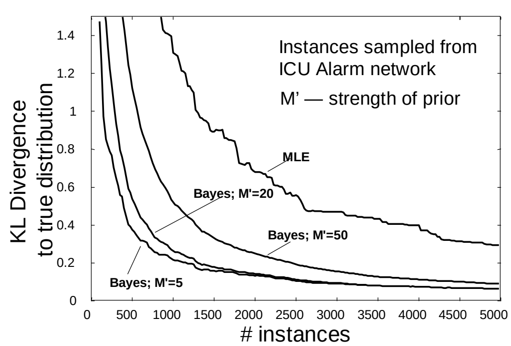
$$\hat{\theta}_{x_i|pa_i} = \frac{N(x_i, pa_i)}{N(pa_i)} \qquad \tilde{\theta}_{x_i|pa_i} = \frac{\alpha(x_i, pa_i) + N(x_i, pa_i)}{\alpha(pa_i) + N(pa_i)}$$

MLE — Bayesian (Dirichlet)

- Both are asymptotically equivalent and consistent
- Both can be implemented in an on-line manner by accumulating sufficient statistics

28

Learning Parameters: Case Study



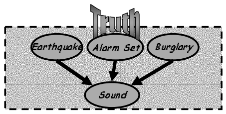
29

Overview

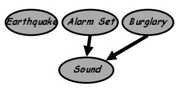
- Introduction
- Parameter Learning
- **Model Selection**
  - Scoring function
  - Structure search
- Structure Discovery
- Incomplete Data
- Learning from Structured Data

30

Why Struggle for Accurate Structure?

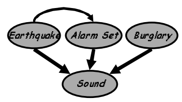


Missing an arc



- Cannot be compensated for by fitting parameters
- Wrong assumptions about domain structure

Adding an arc

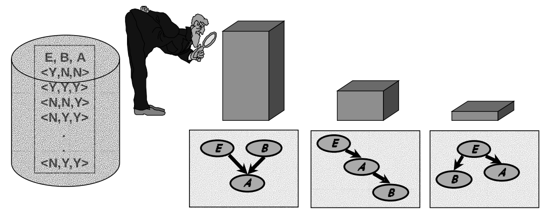


- Increases the number of parameters to be estimated
- Wrong assumptions about domain structure

31

Score-based Learning

Define scoring function that evaluates how well a structure matches the data



Search for a structure that maximizes the score

32

Likelihood Score for Structure

$$\ell(G:D) = \log L(G:D) = M\sum_i \left(I(X_i\,;Pa_i^G) - H(X_i)\right)$$

Mutual information between X_i and its parents

- Larger dependence of X_i on $Pa_i \Rightarrow$ higher score
- Adding arcs always helps
  - $I(X;Y) \leq I(X;\{Y,Z\})$
  - Max score attained by fully connected network
  - Overfitting: A bad idea…

33

Bayesian Score

Likelihood score: $L(G:D) = P(D \mid G, \hat{\theta}_G)$

Max likelihood params

Bayesian approach:

- Deal with uncertainty by assigning probability to all possibilities

$$P(D \mid G) = \int P(D \mid G,\theta) P(\theta \mid G) d\theta$$

Marginal Likelihood · Likelihood · Prior over parameters

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

34

Marginal Likelihood: Multinomials

Fortunately, in many cases integral has closed form

- $P(\Theta)$ is Dirichlet with hyperparameters $\alpha_1,\ldots,\alpha_K$
- D is a dataset with sufficient statistics $N_1,\ldots,N_K$

Then

$$P(D) = \frac{\Gamma\left(\sum_\ell \alpha_\ell\right)}{\Gamma\left(\sum_\ell (\alpha_\ell + N_\ell)\right)} \prod_\ell \frac{\Gamma(\alpha_\ell + N_\ell)}{\Gamma(\alpha_\ell)}$$

35

Marginal Likelihood: Bayesian Networks

- Network structure determines form of marginal likelihood

	1	2	3	4	5	6	7
X	H	T	T	H	T	H	H
Y	H	T	H	H	T	T	H

Network 1:
Two Dirichlet marginal likelihoods

$P($ $)$ **Integral over θ_X**

$P($ $)$ **Integral over θ_Y**

36

Marginal Likelihood: Bayesian Networks

- Network structure determines form of marginal likelihood

	1	2	3	4	5	6	7
X	H	T	T	H	T	H	H
Y	H	T	H	H	T	T	H

Network 2:
Three Dirichlet marginal likelihoods

$X \rightarrow Y$

$P($ $)$ **Integral over θ_X**

$P($ $)$ **Integral over $\theta_{Y|X=H}$**

$P($ $)$ **Integral over $\theta_{Y|X=T}$**

37

Marginal Likelihood for Networks

The marginal likelihood has the form:

$$P(D \mid G) = \prod_i \prod_{pa_i^G}$$

Dirichlet marginal likelihood for multinomial $P(X_i \mid pa_i)$

$$\frac{\Gamma(\alpha(pa_i^G))}{\Gamma(\alpha(pa_i^G)+N(pa_i^G))}\prod_{x_i}\frac{\Gamma(\alpha(x_i,pa_i^G)+N(x_i,pa_i^G))}{\Gamma(\alpha(x_i,pa_i^G))}$$

$N(..)$ are counts from the data
$\alpha(..)$ are hyperparameters for each family **given G**

38

Bayesian Score: Asymptotic Behavior

$$\log P(D \mid G) = \ell(G : D) - \frac{\log M}{2} dim(G) + O(1)$$
$$= \underbrace{M \sum_i \left(I(X_i ; Pa_i^G) - H(X_i)\right)}_{\text{Fit dependencies in empirical distribution}} - \underbrace{\frac{\log M}{2} dim(G)}_{\text{Complexity penalty}} + O(1)$$

- As M (amount of data) grows,
  - Increasing pressure to fit dependencies in distribution
  - Complexity term avoids fitting noise
- Asymptotic equivalence to MDL score
- Bayesian score is **consistent**
  - Observed data eventually overrides prior

39

Structure Search as Optimization

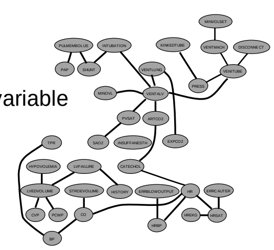
Input:
- Training data
- Scoring function
- Set of possible structures

Output:
- A network that maximizes the score

Key Computational Property: Decomposability:
score(G) = $\sum$ score (family of X in G)

40

Tree-Structured Networks



Trees:
- At most one parent per variable

Why trees?
- Elegant math
  - ⇒ we can solve the optimization problem
- Sparse parameterization
  - ⇒ avoid overfitting

41

Learning Trees

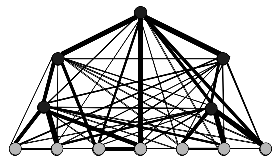
- Let $p(i)$ denote parent of X_i
- We can write the Bayesian score as

$$Score(G : D) = \sum_i Score(X_i : Pa_i)$$
$$= \underbrace{\sum_i \left(Score(X_i : X_{p(i)}) - Score(X_i)\right)}_{\text{Improvement over "empty" network}} + \underbrace{\sum_i Score(X_i)}_{\text{Score of "empty" network}}$$

Score = sum of edge scores + constant

42

Learning Trees



- Set $w(j \to i) = Score(X_j \to X_i) - Score(X_i)$
- Find tree (or forest) with maximal weight
  - Standard max spanning tree algorithm — $O(n^2 \log n)$

Theorem: This procedure finds tree with max score

Beyond Trees

When we consider more complex network, the problem is not as easy

- Suppose we allow at most two parents per node
- A greedy algorithm is no longer guaranteed to find the optimal network

- In fact, no efficient algorithm exists

Theorem: Finding maximal scoring structure with at most k parents per node is NP-hard for $k > 1$

Heuristic Search

- Define a search space:
  - search states are possible structures
  - operators make small changes to structure
- Traverse space looking for high-scoring structures
- Search techniques:
  - Greedy hill-climbing
  - Best first search
  - Simulated Annealing
  - ...

Local Search

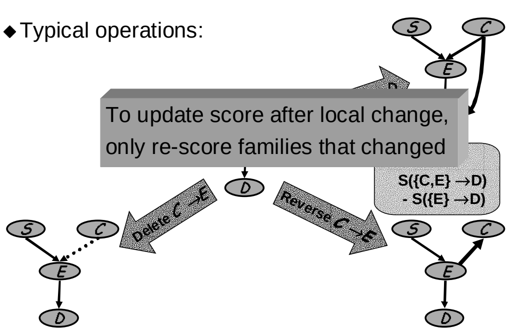
- Start with a given network
  - empty network
  - best tree
  - a random network

- At each iteration
  - Evaluate all possible changes
  - Apply change based on score

- Stop when no modification improves score

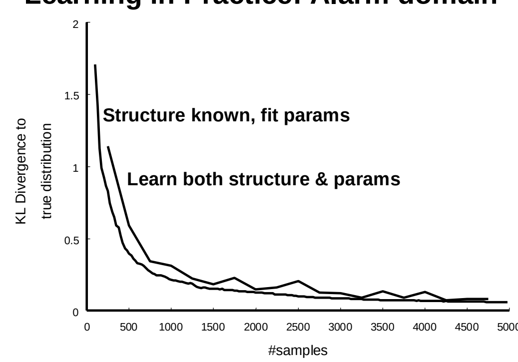
Heuristic Search

- Typical operations:



To update score after local change, only re-score families that changed

Learning in Practice: Alarm domain



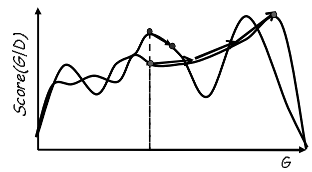
Local Search: Possible Pitfalls

- Local search can get stuck in:
  - **Local Maxima:**
    - All one-edge changes reduce the score
  - **Plateaux:**
    - Some one-edge changes leave the score unchanged
- Standard heuristics can escape both
  - Random restarts
  - TABU search
  - Simulated annealing

49

Improved Search: Weight Annealing

- Standard annealing process:
  - Take bad steps with probability $\propto \exp(\Delta\text{score}/t)$
  - Probability increases with temperature
- Weight annealing:
  - Take uphill steps relative to **perturbed** score
  - Perturbation increases with temperature



50

Perturbing the Score

- Perturb the score by reweighting instances
- Each weight sampled from distribution:
  - Mean = 1
  - Variance $\propto$ temperature
- Instances sampled from "original" distribution
- … but perturbation changes emphasis

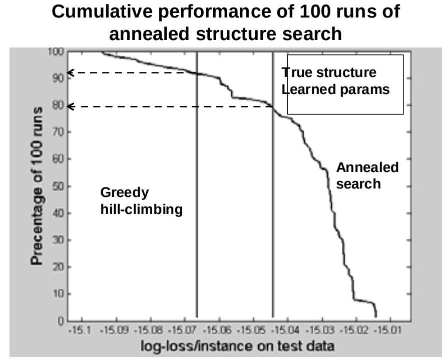
Benefit:

- **Allows global moves in the search space**

51

Weight Annealing: ICU Alarm network

Cumulative performance of 100 runs of annealed structure search



52

Structure Search: Summary

- Discrete optimization problem
- In some cases, optimization problem is easy
  - Example: learning trees
- In general, NP-Hard
  - Need to resort to heuristic search
  - In practice, search is relatively fast (~100 vars in ~2-5 min):
    - Decomposability
    - Sufficient statistics
  - Adding randomness to search is critical

53

Overview

- Introduction
- Parameter Estimation
- Model Selection
- **Structure Discovery**
- Incomplete Data
- Learning from Structured Data

54

Structure Discovery

Task: Discover structural properties

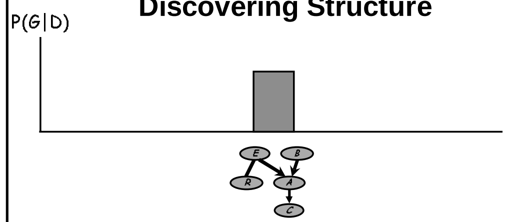
- Is there a direct connection between X & Y
- Does X separate between two "subsystems"
- Does X causally effect Y

Example: scientific data mining

- Disease properties and symptoms
- Interactions between the expression of genes

55

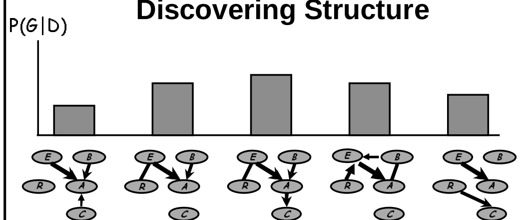
Discovering Structure



- Current practice: model selection
  - Pick a single high-scoring model
  - Use that model to infer domain structure

56

Discovering Structure



Problem

- Small sample size $\Rightarrow$ many high scoring models
- Answer based on one model often useless
- Want features common to many models

57

Bayesian Approach

- Posterior distribution over structures
- Estimate probability of **features**
  - Edge $X \rightarrow Y$
  - Path $X \rightarrow \ldots \rightarrow Y$
  - ...

$$P(f \mid D) = \sum_{G} f(G) P(G \mid D)$$

Bayesian score for G

Feature of G, e.g., $X \rightarrow Y$

Indicator function for feature f

58

MCMC over Networks

- Cannot enumerate structures, so sample structures

$$P(f(G) \mid D) \approx \frac{1}{n} \sum_{i=1}^{n} f(G_i)$$

- MCMC Sampling
  - Define Markov chain over BNs
  - Run chain to get samples from posterior ***P(G | D)***

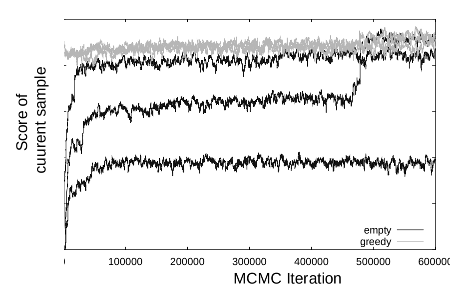
Possible pitfalls:

- Huge (superexponential) number of networks
- Time for chain to converge to posterior is unknown
- Islands of high posterior, connected by low bridges

59

ICU Alarm BN: No Mixing

- 500 instances:

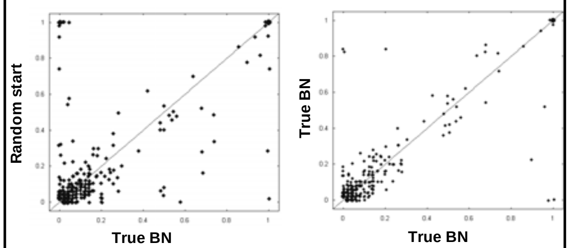


- The runs clearly do not mix

60

Effects of Non-Mixing

- Two MCMC runs over same 500 instances
- Probability estimates for edges for two runs



Probability estimates highly variable, nonrobust

Fixed Ordering

Suppose that

- We know the **ordering** of variables
  - say, $X_1 \succ X_2 \succ X_3 \succ X_4 \succ \dots \succ X_n$
    parents for X_i must be in $X_1, \dots, X_{i-1}$
- Limit number of parents per nodes to k

$2^{k \cdot n \cdot \log n}$ networks

Intuition: Order **decouples** choice of parents

- Choice of $Pa(X_7)$ does not restrict choice of $Pa(X_{12})$

Upshot: Can compute efficiently in closed form

- Likelihood $P(D \mid \prec)$
- Feature probability $P(f \mid D, \prec)$

Our Approach: Sample Orderings

We can write

$$P(f \mid D) = \sum_{\prec} P(f \mid \prec, D) P(\prec \mid D)$$

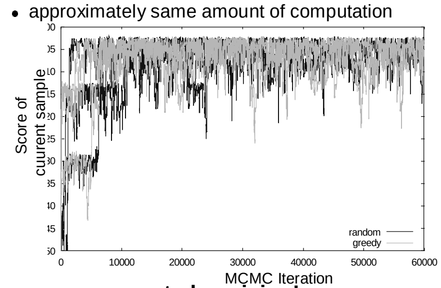
Sample orderings and approximate

$$P(f \mid D) \approx \sum_{i=1}^{n} P(f \mid \prec_i, D)$$

- MCMC Sampling
  - Define Markov chain over orderings
  - Run chain to get samples from posterior $P(\prec \mid D)$

Mixing with MCMC-Orderings

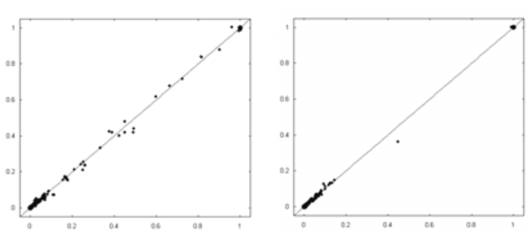
- 4 runs on ICU-Alarm with 500 instances
  - fewer iterations than MCMC-Nets
  - approximately same amount of computation



Process appears to be mixing!

Mixing of MCMC runs

- Two MCMC runs over same instances
- Probability estimates for edges



500 instances   1000 instances

Probability estimates very robust

Application: Gene expression

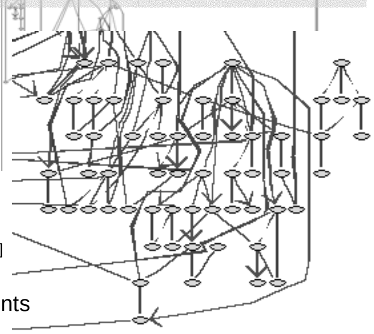
Input:

- Measurement of gene expression under different conditions
  - Thousands of genes
  - Hundreds of experiments

Output:

- Models of gene interaction
  - Uncover pathways

Map of Feature Confidence

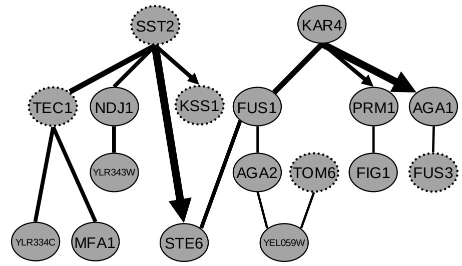


Yeast data
[Hughes et al 2000]

- 600 genes
- 300 experiments

67

"Mating response" Substructure



- **Automatically constructed sub-network of high-confidence edges**
- **Almost exact reconstruction of yeast mating pathway**

68

Overview

- Introduction
- Parameter Estimation
- Model Selection
- Structure Discovery
- **Incomplete Data**
  - **Parameter estimation**
  - **Structure search**
- Learning from Structured Data

69

Incomplete Data

Data is often **incomplete**

- Some variables of interest are not assigned values

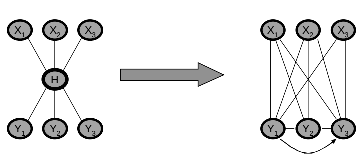
This phenomenon happens when we have

- **Missing values:**
  - Some variables unobserved in some instances
- **Hidden variables:**
  - Some variables are never observed
  - We might not even know they exist

70

Hidden (Latent) Variables

Why should we care about unobserved variables?

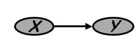


17 parameters

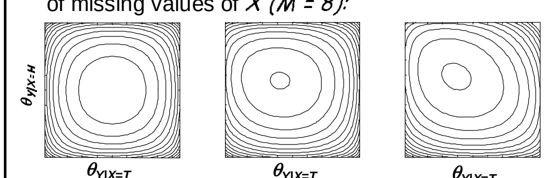
59 parameters

71

Example



- *P(X)* assumed to be known
- Likelihood function of: $\theta_{Y|X=T}$, $\theta_{Y|X=H}$
- Contour plots of log likelihood for different number of missing values of *X (M = 8)*:



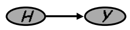
no missing values | **2 missing value** | **3 missing values**

In general: likelihood function has multiple modes

72

Incomplete Data

- In the presence of incomplete data, the likelihood can have multiple maxima

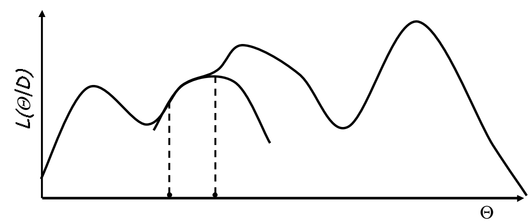


Example:

- We can rename the values of hidden variable H
- If H has two values, likelihood has two maxima

- **In practice, many local maxima**

EM: MLE from Incomplete Data



- Use current point to construct "nice" alternative function
- Max of new function scores ≥ than current point

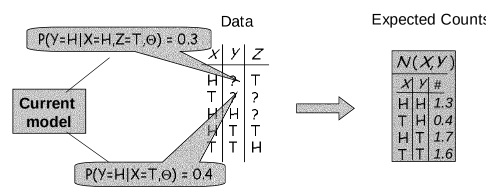
Expectation Maximization (EM)

- A general purpose method for learning from incomplete data

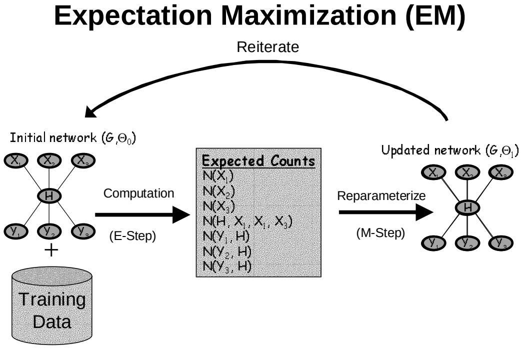
Intuition:

- If we had true counts, we could estimate parameters
- But with missing values, counts are unknown
- We "complete" counts using probabilistic inference based on current parameter assignment
- We use completed counts as if real to re-estimate parameters

Expectation Maximization (EM)



Expectation Maximization (EM)



Expectation Maximization (EM)

Formal Guarantees:

- $L(\Theta_1:D) \geq L(\Theta_0:D)$
  - Each iteration improves the likelihood
- If $\Theta_1 = \Theta_0$, then Θ_0 is a **stationary point** of $L(\Theta:D)$
  - Usually, this means a local maximum

Expectation Maximization (EM)

Computational bottleneck:

- Computation of expected counts in E-Step
  - Need to compute posterior for each unobserved variable in each instance of training set
  - All posteriors for an instance can be derived from one pass of standard BN inference

79

Summary: Parameter Learning with Incomplete Data

- Incomplete data makes parameter estimation hard
- Likelihood function
  - Does not have closed form
  - Is multimodal

- Finding max likelihood parameters:
  - EM
  - Gradient ascent
- Both exploit inference procedures for Bayesian networks to compute expected sufficient statistics

80

Incomplete Data: Structure Scores

Recall, Bayesian score:

$$P(G \mid D) \propto P(G)P(D \mid G)$$
$$= P(G)\boxed{\int P(D \mid G,\Theta)P(\Theta \mid G)d\theta}$$

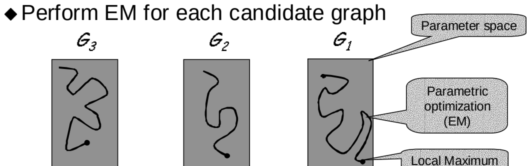
With incomplete data:

- Cannot evaluate marginal likelihood in closed form
- We have to resort to **approximations**:
  - Evaluate score around MAP parameters
  - Need to find MAP parameters (e.g., EM)

81

Naive Approach

- Perform EM for each candidate graph



- Computationally expensive:
  - Parameter optimization via EM — non-trivial
  - Need to perform EM for all candidate structures
  - Spend time even on poor candidates
- ⇒In practice, considers only a few candidates

82

Structural EM

Recall, in complete data we had

- Decomposition ⇒ efficient search

Idea:

- Instead of optimizing the real score…
- Find **decomposable** alternative score
- Such that maximizing new score

  ⇒ improvement in real score

83

Structural EM

Idea:

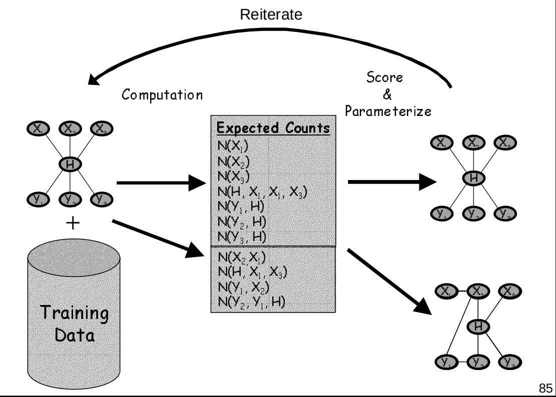
- Use current model to help evaluate new structures

Outline:

- Perform search in (Structure, Parameters) space
- At each iteration, use current model for finding either:
  - Better scoring parameters: “parametric” EM step

  or
  - Better scoring structure: “structural” EM step

84



Example: Phylogenetic Reconstruction

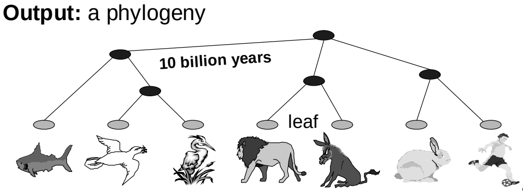
Input: Biological sequences

Human	CGTTGC…
Chimp	CCTAGG…
Orang	CGAACG…
….	

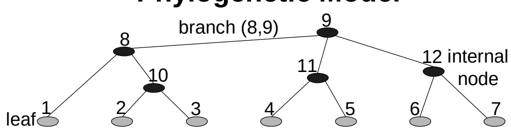
An "instance" of evolutionary process

Assumption: positions are independent

Output: a phylogeny



Phylogenetic Model



- Topology: bifurcating
  - Observed species – $1 \ldots N$
  - Ancestral species – $N+1 \ldots 2N-2$
- Lengths $t = \{t_{i,j}\}$ for each branch (i,j)
- Evolutionary model:
  - *P(A changes to T| 10 billion yrs)*

Phylogenetic Tree as a Bayes Net

- Variables: Letter at each position for each species
  - Current day species – observed
  - Ancestral species - hidden
- BN Structure: Tree topology
- BN Parameters: Branch lengths (time spans)

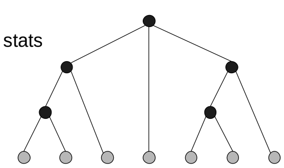
Main problem: Learn topology

If ancestral were observed
⇒ easy learning problem (learning trees)

Algorithm Outline

→ Compute expected pairwise stats

→ Weights: Branch scores



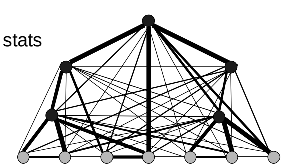
Original Tree (T^0, t^0)

Algorithm Outline

→ Compute expected pairwise stats

→ Weights: Branch scores

→ Find: $T' = \arg\max_T \sum_{(i,j) \in T} w_{i,j}$

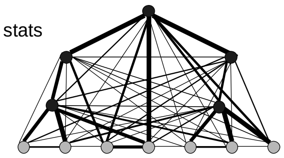


Pairwise weights

$O(N^2)$ pairwise statistics suffice to evaluate all trees

Algorithm Outline

- → Compute expected pairwise stats
- → Weights: Branch scores
- → Find: $T' = \arg\max_T \sum_{(i,j)\in T} w_{i,j}$
- → Construct bifurcation T_1

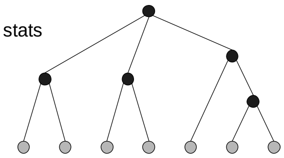


Max. Spanning Tree

91

Algorithm Outline

- → Compute expected pairwise stats
- → Weights: Branch scores
- → Find: $T' = \arg\max_T \sum_{(i,j)\in T} w_{i,j}$
- → Construct bifurcation T_1
- → Theorem: $L(T_1,t_1) \geq L(T_0,t_0)$



New Tree

Repeat until convergence…

92

Real Life Data

		Lysozyme *c*	Mitochondrial genomes
	# sequences	43	34
	# pos	122	3,578
Log-likelihood	Traditional approach	-2,916.2	-74,227.9
	Structural EM Approach	-2,892.1	-70,533.5
	Difference per position	0.19	1.03

Each position twice as likely

93

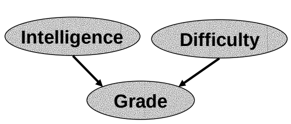
Overview

- Introduction
- Parameter Estimation
- Model Selection
- Structure Discovery
- Incomplete Data
- **Learning from Structured Data**

94

Bayesian Networks: Problem

- Bayesian nets use propositional representation
- Real world has objects, related to each other

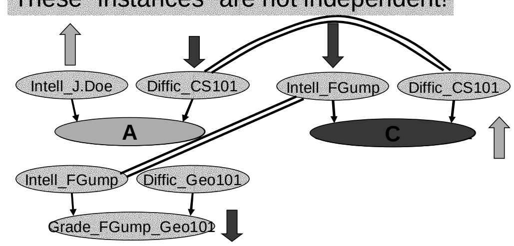


95

Bayesian Networks: Problem

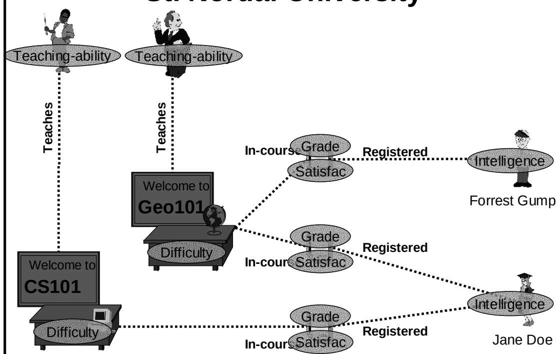
- Bayesian nets use propositional representation
- Real world has objects, related to each other

These “instances” are not independent!



96

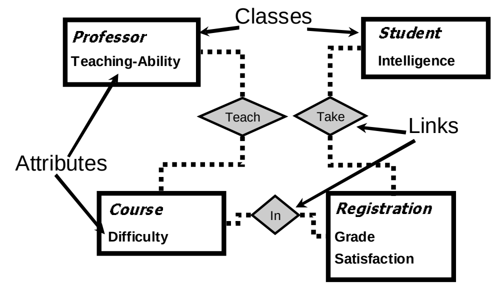
St. Nordaf University



97

Relational Schema

- Specifies types of objects in domain, attributes of each type of object, & types of links between objects



98

Representing the Distribution

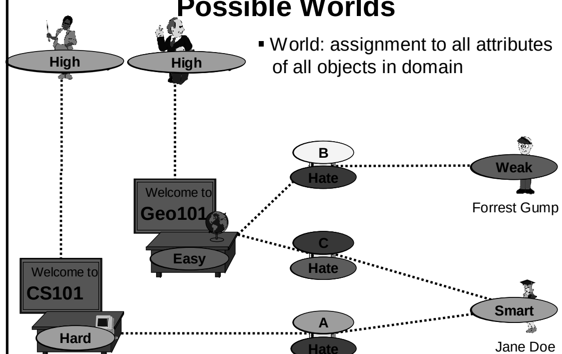
- Many possible worlds for a given university
  - All possible assignments of all attributes of all objects
- Infinitely many potential universities
  - Each associated with a very different set of worlds

Need to represent infinite set of complex distributions

99

Possible Worlds

- World: assignment to all attributes of all objects in domain

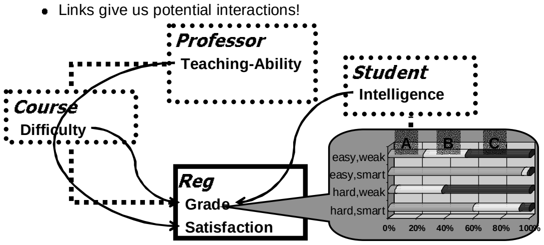


100

Probabilistic Relational Models

Key ideas:

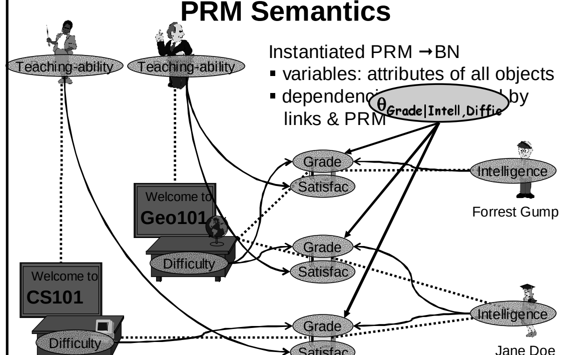
- Universals: Probabilistic patterns hold for all objects in class
- Locality: Represent direct probabilistic dependencies
  - Links give us potential interactions!



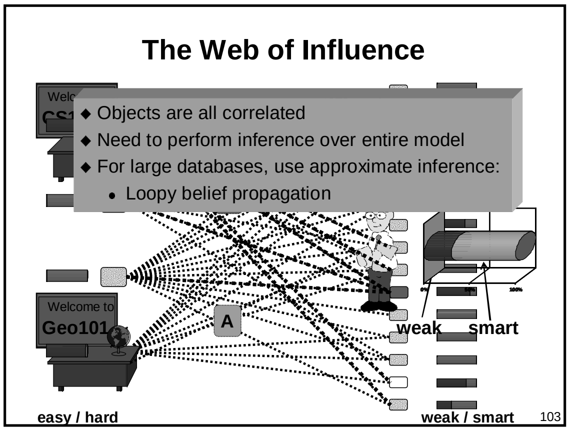
101

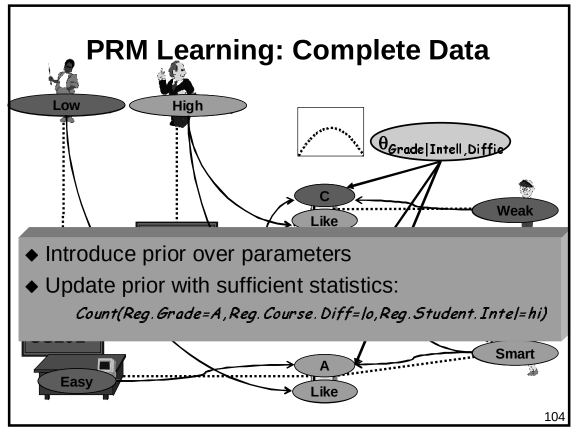
PRM Semantics

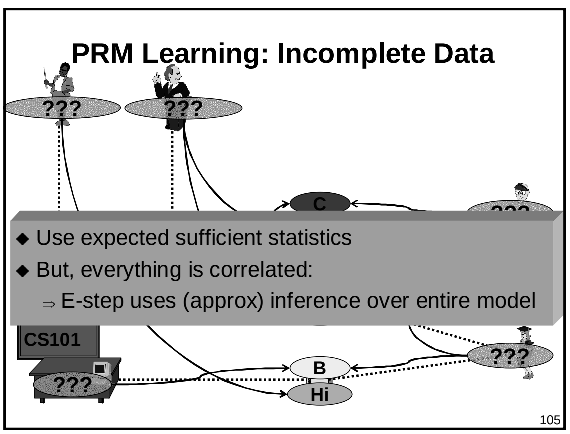
Instantiated PRM →BN
- variables: attributes of all objects
- dependencies determined by links & PRM

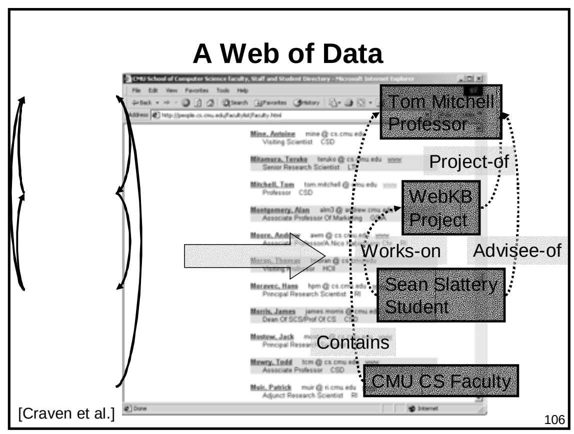


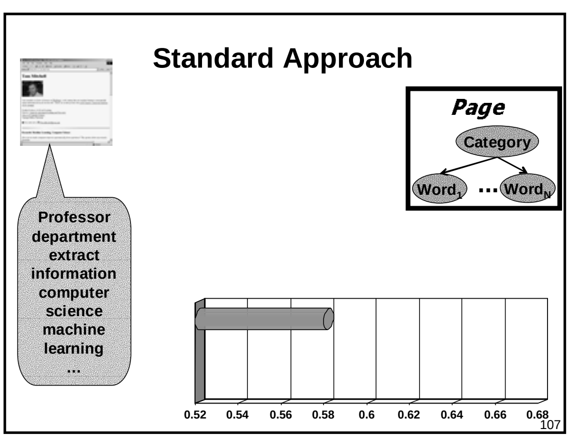
102

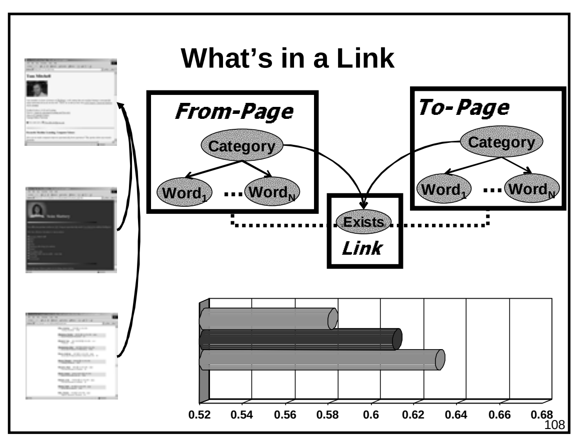

The Web of Influence
Objects are all correlated
Need to perform inference over entire model
For large databases, use approximate inference:
Loopy belief propagation
Welcome to
Geo101
A
weak
smart
easy / hard
weak / smart
103


PRM Learning: Complete Data
Low
High
$\theta_{Grade|Intell,Diffic}$
C
Like
Weak
Introduce prior over parameters
Update prior with sufficient statistics:
Count(Reg.Grade=A,Reg.Course.Diff=lo,Reg.Student.Intel=hi)
Smart
A
Easy
Like
104

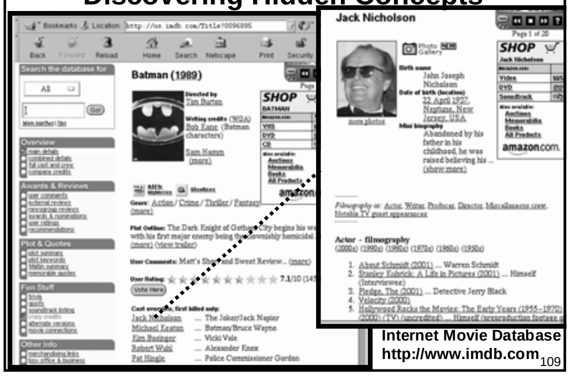

PRM Learning: Incomplete Data
???
???
C
Use expected sufficient statistics
But, everything is correlated:
⇒ E-step uses (approx) inference over entire model
CS101
B
???
???
Hi
105


A Web of Data
Tom Mitchell
Professor
Project-of
WebKB
Project
Works-on
Advisee-of
Sean Slattery
Student
Contains
CMU CS Faculty
[Craven et al.]
106


Standard Approach
Page
Category
Word1
...
WordN
Professor
department
extract
information
computer
science
machine
learning
...
0.52
0.54
0.56
0.58
0.6
0.62
0.64
0.66
0.68
107


What's in a Link
From-Page
Category
Word1
...
WordN
To-Page
Category
Word1
...
WordN
Exists
Link
0.52
0.54
0.56
0.58
0.6
0.62
0.64
0.66
0.68
108

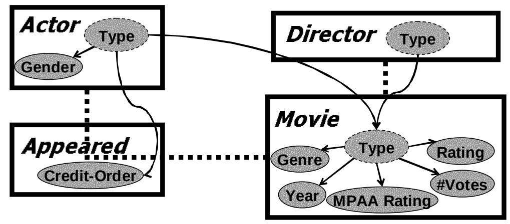
Discovering Hidden Concepts



Internet Movie Database
http://www.imdb.com

109

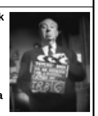
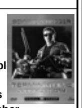
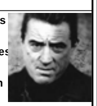
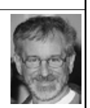
Discovering Hidden Concepts



Internet Movie Database
http://www.imdb.com

110

Web of Influence, Yet Again

Movies	Actors	Directors
Wizard of Oz Cinderella Sound of Music The Love Bug Pollyanna The Parent Trap Mary Poppins Swiss Family Robinson 	Sylvester Stallone Bruce Willis Harrison Ford Steven Seagal Kurt Russell Kevin Costner Jean-Claude Van Damme Arnold Schwarzenegger 	Alfred Hitchcock Stanley Kubrick David Lean Milos Forman Terry Gilliam Francis Coppola 
Terminator 2 Batman Batman Forever Mission: Impossibl GoldenEye Starship Troopers Hunt for Red October 	Anthony Hopkins Robert De Niro Tommy Lee Jones Harvey Keitel Morgan Freeman Gary Oldman 	Steven Spielberg Tim Burton Tony Scott James Cameron John McTiernan Joel Schumacher 
…	…	…

111

Conclusion

- Many distributions have combinatorial dependency structure
- Utilizing this structure is good
- Discovering this structure has implications:
  - To density estimation
  - To knowledge discovery
- Many applications
  - Medicine
  - Biology
  - Web

112

The END

Thanks to

- Gal Elidan
- Lise Getoor
- Moises Goldszmidt
- Matan Ninio
- Dana Pe'er
- Eran Segal
- Ben Taskar

Slides will be available from:

http://www.cs.huji.ac.il/~nir/

http://robotics.stanford.edu/~koller/

113