Learning Bayesian Networks from Data

Nir Friedman
U.C. Berkeley
www.cs.berkeley.edu/~nir

Moises Goldszmidt
SRI International
www.erg.sri.com/people/moises

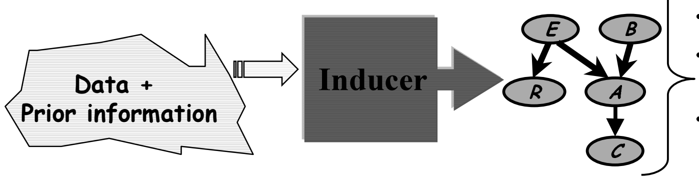
For current slides, additional material, and reading list see http://www.cs.berkeley.edu/~nir/Tutorial

Outline

- »Introduction
- ◆Bayesian networks: a review
- ◆Parameter learning: Complete data
- ◆Parameter learning: Incomplete data
- ◆Structure learning: Complete data
- ◆Application: classification
- ◆Learning causal relationships
- ◆Structure learning: Incomplete data
- ◆Conclusion

Learning (in this context)

- Process
  - **Input:** *dataset* and *prior information*
  - ***Output:*** *Bayesian network*
- Prior information: background knowledge
  - a Bayesian network (or fragments of it)
  - time ordering
  - prior probabilities
  - ...



Represents

- **P(E,B,R,S,C)**
- **Independence Statements**
- **Causality**

Why learning?

- Feasibility of learning
  - Availability of data and computational power
- Need for learning
  - Characteristics of current systems and processes
    - Defy closed form analysis
      - ⇨need data-driven approach for characterization
    - Scale and change fast
      - ⇨need continuous automatic adaptation
- Examples:
  - communication networks, economic markets, illegal activities, the brain...

Why learn a Bayesian network?

- **Combine knowledge engineering and statistical induction**
  - Covers the whole spectrum from *knowledge-intensive* model construction to *data-intensive* model induction
- **More than a learning black-box**
  - Explanation of outputs
  - Interpretability and modifiability
  - Algorithms for decision making, value of information, diagnosis and repair
- **Causal representation, reasoning, and discovery**
  - Does smoking cause cancer?

What will I get out of this tutorial?

- An understanding of the basic concepts behind the process of learning Bayesian networks from data so that you can
  - Read advanced papers on the subject
  - Jump start possible applications
  - Implement the necessary algorithms
  - Advance the state-of-the-art

Outline

- Introduction
- » Bayesian networks: a review
  - Probability 101
  - What are Bayesian networks?
  - What can we do with Bayesian networks?
  - The learning problem...
- Parameter learning: Complete data
- Parameter learning: Incomplete data
- Structure learning: Complete data
- Application: classification
- Learning causal relationships
- Structure learning: Incomplete data
- Conclusion

Probability 101

- Bayes rule

$$P(X \mid Y) = \frac{P(Y \mid X) \cdot P(X)}{P(Y)}$$

- Chain rule

$$P(X_1, \ldots, X_n) = P(X_1)P(X_2 \mid X_1) \cdots P(X_n \mid X_1, \ldots, X_{n-1})$$

- Introduction of a variable (reasoning by cases)

$$P(X \mid Y) = \sum_{Z} P(X \mid Z, Y) \cdot P(Z \mid Y)$$

Representing the Uncertainty in a Domain

- A story with five random variables:
  - Burglary, Earthquake, Alarm, Neighbor Call, Radio Announcement
  - Specify a joint distribution with $2^5-1=31$ parameters

maybe…

- An expert system for monitoring intensive care patients
  - Specify a joint distribution over 37 variables with (at least) 2^{37} parameters

no way!!!

Probabilistic Independence: a Key for Representation and Reasoning

- Recall that if X and Y are **independent** given Z then

$$P(X \mid Z,Y) = P(X \mid Z)$$

- In our story…if
  - *burglary* and *earthquake* are **independent**
  - *burglary* and *radio* are **independent** given *earthquake*
- then we can reduce the number of probabilities needed

Probabilistic Independence: a Key for Representation and Reasoning

- In our story…if
  - *burglary* and *earthquake* are **independent**
  - *burglary* and *radio* are **independent** given *earthquake*
- then instead of 15 parameters we need 8

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

versus

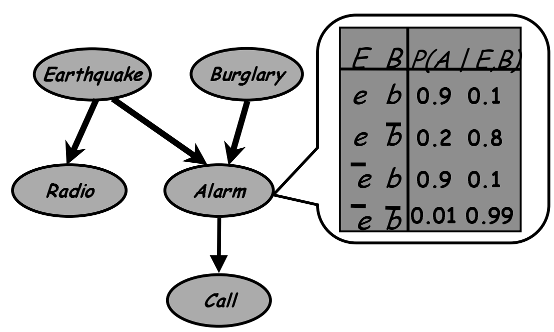
$$P(A,R,E,B) = P(A \mid E,B) \cdot P(R \mid E) \cdot P(E) \cdot P(B)$$

Need a language to represent independence statements

MP1-11

Bayesian Networks

Computer efficient representation of probability distributions via conditional independence

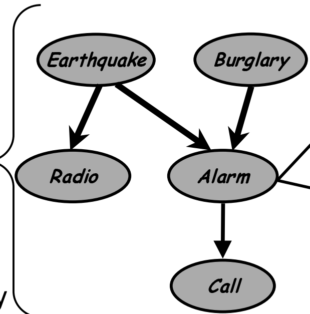


MP1-12

Bayesian Networks

Qualitative part: statistical independence statements (causality!)

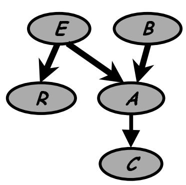
- Directed acyclic graph (DAG)
  - Nodes - random variables of interest (exhaustive and mutually exclusive states)
  - Edges - direct (causal) influence



E	B	P(A | E,B)	
e	b	0.9	0.1
e	$\bar{b}$	0.2	0.8
$\bar{e}$	b	0.9	0.1
$\bar{e}$	$\bar{b}$	0.01	0.99

- **Quantitative part**: Local probability models. Set of conditional probability distributions.

Bayesian Network Semantics



Qualitative part		**Quantitative part**		
conditional independence statements in BN structure	+	local probability models	=	Unique joint distribution over domain

- Compact & efficient representation:
  - nodes have $\leq k$ parents $\Rightarrow O(2^k n)$ vs. $O(2^n)$ params
  - parameters pertain to local interactions

P(C,A,R,E,B) = P(B)*P(E|B)*P(R|E,B)*P(A|R,B,E)*P(C|A,R,B,E)

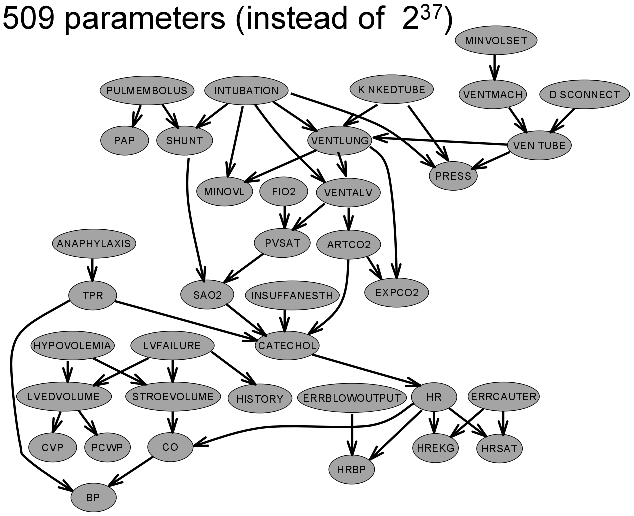
versus

P(C,A,R,E,B) = P(B)*P(E) * P(R|E) * P(A|B,E) * P(C|A)

Monitoring Intensive-Care Patients

The “alarm” network

37 variables, 509 parameters (instead of 2^{37})



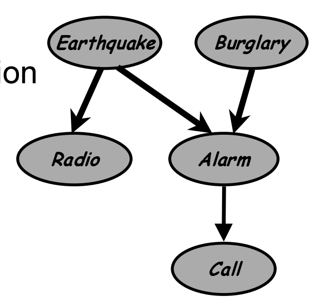
MP1-15

Qualitative part

- Nodes are independent of non-descendants given their parents
  - *P(R|E=y,A) = P(R|E=y) for all values of R,A,E*
    Given that there is and *earthquake*,
    I can predict a *radio announcement*
    regardless of whether the *alarm* sounds

- **d-separation**: a graph theoretic criterion for reading independence statements

  Can be computed in linear time
  (on the number of edges)



MP1-16

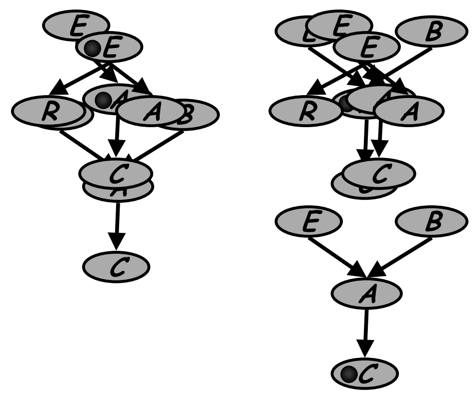
d-separation

- Two variables are independent if all paths between them are **blocked** by evidence

Three cases:

- Common cause
- Intermediate cause
- Common Effect

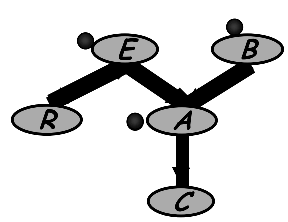
Blocked **Unblocked**



MP1-17

Example

- *I(X,Y|Z)* denotes *X* and *Y* are independent given *Z*
  - *I(R,B)*
  - *~I(R,B|A)*
  - *I(R,B|E,A)*
  - *~I(R,C|B)*

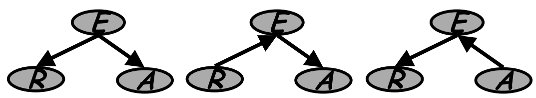


MP1-18

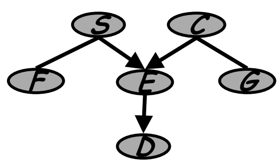
I-Equivalent Bayesian Networks

- Networks are I-equivalent if their structures encode the same independence statements

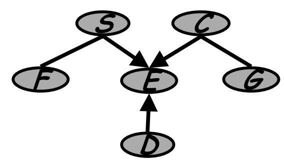
I(R,A|E)



- Theorem: Networks are I-equivalent iff they have the same skeleton and the same "V" structures

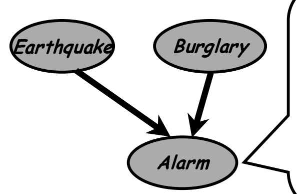


NOT I-equivalent



Quantitative Part

- Associated with each node X_i there is a set of conditional probability distributions $P(X_i|Pa_i:\Theta)$
  - If variables are discrete, Θ is usually multinomial

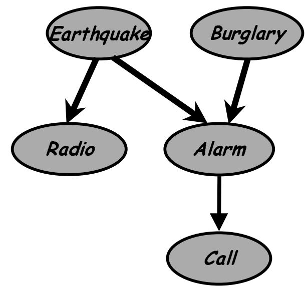


E	B	P(A | E,B)	
e	b	0.9	0.1
e	$\bar{b}$	0.2	0.8
$\bar{e}$	b	0.9	0.1
$\bar{e}$	$\bar{b}$	0.01	0.99

  - Variables can be continuous, Θ can be a linear Gaussian
  - Combinations of discrete and continuous are only constrained by available inference mechanisms

What Can We Do with Bayesian Networks?

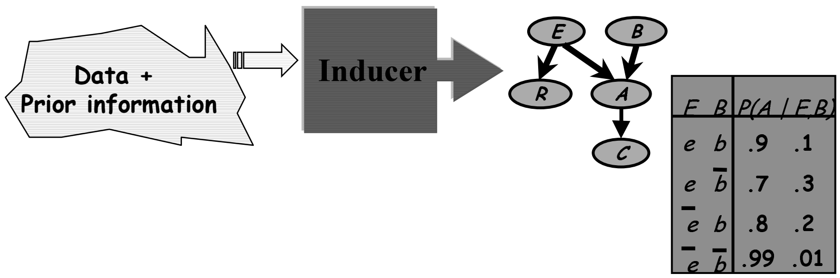
- Probabilistic inference: belief update
  - *P(E =Y| R = Y, C = Y)*
- Probabilistic inference: belief revision
  - $\text{Argmax}_{\{E,B\}}$ *P(e, b | C=Y)*
- Qualitative inference
  - *I(R,C| A)*
- Complex inference
  - rational decision making (influence diagrams)
  - value of information
  - sensitivity analysis
- Causality (analysis under interventions)



Bayesian Networks: Summary

- Bayesian networks:
  an efficient and effective representation of probability distributions
- Efficient:
  - Local models
  - Independence (d-separation)
- Effective: Algorithms take advantage of structure to
  - Compute posterior probabilities
  - Compute most probable instantiation
  - Decision making
- But there is more: statistical induction → LEARNING

Learning Bayesian networks (reminder)

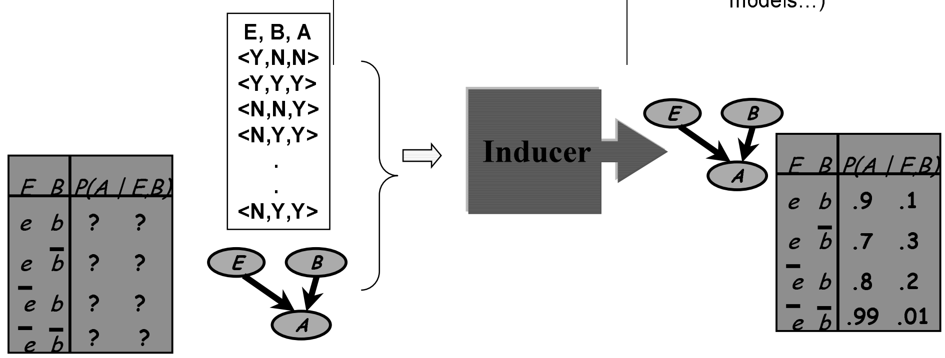


The Learning Problem

	Known Structure	Unknown Structure
Complete Data	Statistical parametric estimation (closed-form eq.)	Discrete optimization over structures (discrete search)
Incomplete Data	Parametric optimization (EM, gradient descent...)	Combined (Structural EM, mixture models…)

Learning Problem

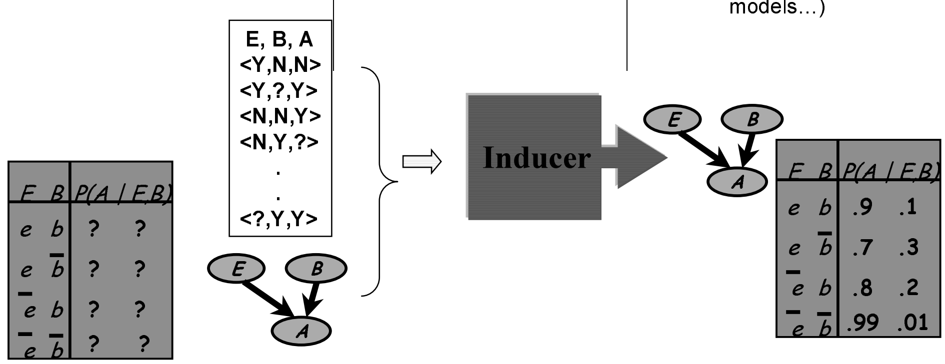
	Known Structure	Unknown Structure
Complete	Statistical parametric estimation (closed-form eq.)	Discrete optimization over structures (discrete search)
Incomplete	Parametric optimization (EM, gradient descent...)	Combined (Structural EM, mixture models…)



MP1-25

Learning Problem

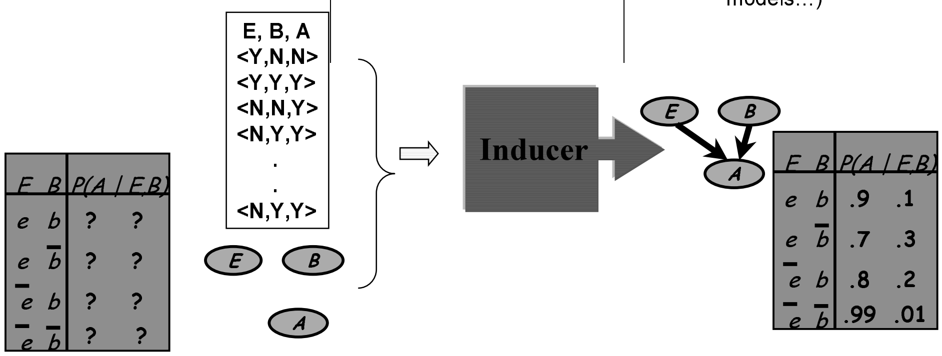
	Known Structure	Unknown Structure
Complete	Statistical parametric estimation (closed-form eq.)	Discrete optimization over structures (discrete search)
Incomplete	Parametric optimization (EM, gradient descent...)	Combined (Structural EM, mixture models…)



MP1-26

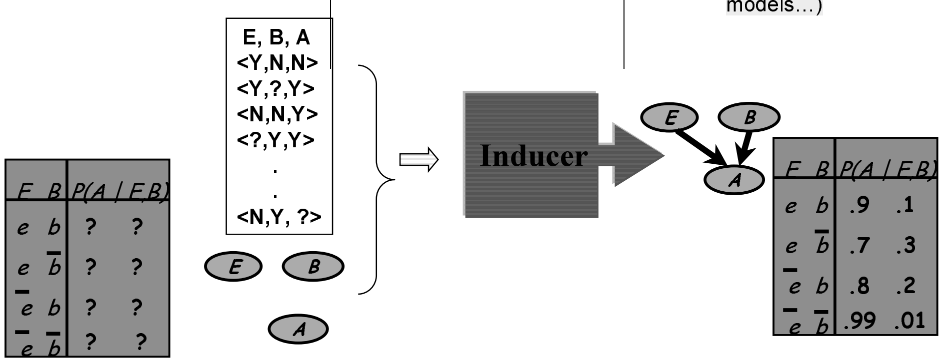
Learning Problem

	Known Structure	Unknown Structure
Complete	Statistical parametric estimation (closed-form eq.)	Discrete optimization over structures (discrete search)
Incomplete	Parametric optimization (EM, gradient descent...)	Combined (Structural EM, mixture models…)



Learning Problem

	Known Structure	Unknown Structure
Complete	Statistical parametric estimation (closed-form eq.)	Discrete optimization over structures (discrete search)
Incomplete	Parametric optimization (EM, gradient descent...)	Combined (Structural EM, mixture models…)



Outline

	Known Structure	Unknown Structure
Complete data	●	
Incomplete data		

- ◆Introduction
- ◆Bayesian networks: a review
- »Parameter learning: Complete data
  - Statistical parametric fitting
  - Maximum likelihood estimation
  - Bayesian inference
- ◆Parameter learning: Incomplete data
- ◆Structure learning: Complete data
- ◆Application: classification
- ◆Learning causal relationships
- ◆Structure learning: Incomplete data
- ◆Conclusion

MP1-29

Example: Binomial Experiment (Statistics 101)

Head Tail

- ◆When tossed, it can land in one of two positions: *Head* or *Tail*
- ◆We denote by θ the (unknown) probability *P(H).*

Estimation task:

- ◆Given a sequence of toss samples *x[1], x[2], …, x[M]* we want to estimate the probabilities *P(H)*= θ and *P(T) = 1* - θ

MP1-30

Statistical parameter fitting

- Consider instances *x[1], x[2], …, x[M]* such that
  - The set of values that x can take is known
  - Each is sampled from the same distribution
  - Each sampled independently of the rest

  } i.i.d. samples

- The task is to find a parameter Θ so that the data can be summarized by a probability $P(x[j] | \Theta)$.
  - The parameters depend on the given family of probability distributions: multinomial, Gaussian, Poisson, etc.
  - We will focus on multinomial distributions
  - The main ideas generalize to other distribution families

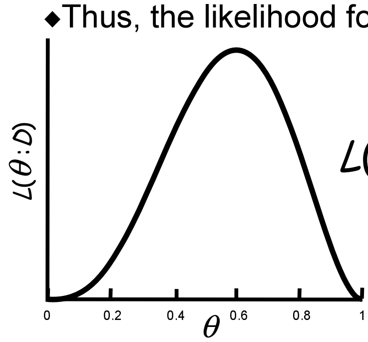
MP1-31

The Likelihood Function

- How good is a particular θ?
  It depends on how likely it is to generate the observed data

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

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



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

MP1-32

Sufficient Statistics

- To compute the likelihood in the thumbtack example we only require N_H and N_T
  (the number of heads and the number of tails)

$$L(\theta : D) = \theta^{N_H} \cdot (1-\theta)^{N_T}$$

  N_H and N_T are **sufficient statistics** for the binomial distribution

- A **sufficient statistic** is a function that summarizes, from the data, the relevant information for the likelihood
  - If $s(D) = s(D')$, then $L(\theta \mid D) = L(\theta \mid D')$

Maximum Likelihood Estimation

MLE Principle:

Learn parameters that maximize the likelihood function

This is one of the most commonly used estimators in statistics

Intuitively appealing

Maximum Likelihood Estimation (Cont.)

- Consistent
  - Estimate converges to best possible value as the number of examples grow
- Asymptotic efficiency
  - Estimate is as close to the true value as possible given a particular training set
- Representation invariant
  - A transformation in the parameter representation does not change the estimated probability distribution

Example: MLE in Binomial Data

- Applying the MLE principle we get

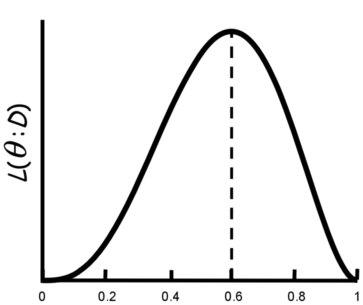
$$\hat{\theta} = \frac{N_H}{N_H + N_T}$$

(Which coincides with what one would expect)

Example:

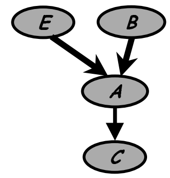
$(N_H, N_T) = (3,2)$

MLE estimate is $3/5 = 0.6$



Learning Parameters for the Burglary Story

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



i.i.d. samples

Network factorization

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

$$= \prod_m P(C[m] \mid A[m] : \Theta_{C|A}) \cdot P(A[m] \mid B[m], E[M] : \Theta_{A|B,E}) \cdot P(B[m] : \Theta_B) \cdot P(E[m] : \Theta_E)$$

$$= \prod_M P(C[m] \mid A[m] : \Theta_{C|A}) \prod_m P(A[m] \mid B[m], E[M] : \Theta_{A|B,E}) \prod_m P(B[m] : \Theta_B) \cdot \prod_m P(E[m] : \Theta_E)$$

We have 4 independent estimation problems

General Bayesian Networks

We can define the likelihood for a Bayesian network**:**

i.i.d. samples

Network factorization

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

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

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

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

The likelihood **decomposes** according to the structure of the network.

General Bayesian Networks (Cont.)

Decomposition ⇒ Independent Estimation Problems

If the parameters for each family are not related, then they can be estimated independently of each other.

From Binomial to Multinomial

- For example, suppose X can have the values $1,2,\ldots,K$
- We want to learn the parameters $\theta_1, \theta_2, \ldots, \theta_K$

Sufficient statistics:

- $N_1, N_2, \ldots, N_K$ - the number of times each outcome is observed

Likelihood function:

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

MLE:

$$\hat{\theta}_k = \frac{N_k}{\sum_{\ell} N_{\ell}}$$

Likelihood for Multinomial Networks

- When we assume that $P(X_i \mid Pa_i)$ is multinomial, we get further decomposition:

$$L_i(\Theta_i : D) = \prod_m P(x_i[m] \mid Pa_i[m] : \Theta_i)$$
$$= \prod_{pa_i} \prod_{m, Pa_i[m]=pa_i} P(x_i[m] \mid pa_i : \Theta_i)$$
$$= \prod_{pa_i} \prod_{x_i} P(x_i \mid pa_i : \Theta_i)^{N(x_i, pa_i)} = \prod_{pa_i} \prod_{x_i} \theta_{x_i \mid pa_i}^{N(x_i, pa_i)}$$

- For each value pa_i of the parents of X_i we get an independent multinomial problem
- The MLE is $\hat{\theta}_{x_i \mid pa_i} = \frac{N(x_i, pa_i)}{N(pa_i)}$

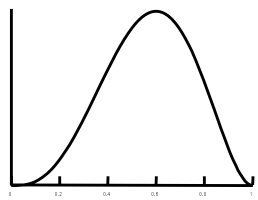
Is MLE all we need?

- Suppose that after 10 observations,
  ML estimates P(H) = 0.7 for the thumbtack
  - Would you bet on heads for the next toss?

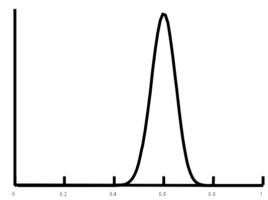
- Suppose now that after 10 observations,
  ML estimates P(H) = 0.7 for a coin
  - Would you place the same bet?

Bayesian Inference

◆MLE commits to a specific value of the unknown parameter(s)



vs.



Coin **Thumbtack**

◆MLE is the same in both cases

◆Confidence in prediction is clearly different

Bayesian Inference (cont.)

Frequentist Approach:

◆Assumes there is an unknown but fixed parameter θ

◆Estimates θ with some confidence

◆Prediction by using the estimated parameter value

Bayesian Approach:

◆Represents uncertainty about the unknown parameter

◆Uses probability to quantify this uncertainty:

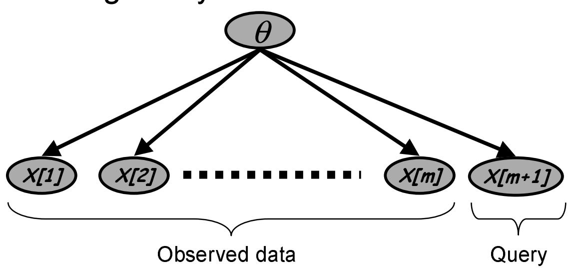
- Unknown parameters as random variables

◆Prediction follows from the rules of probability:

- Expectation over the unknown parameters

Bayesian Inference (cont.)

- We can represent our uncertainty about the sampling process using a Bayesian network

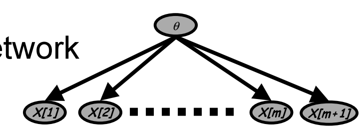


- The observed values of X are independent given θ
- The conditional probabilities, $P(x[m] \mid \theta)$, are the parameters in the model
- Prediction is now inference in this network

MP1-45

Bayesian Inference (cont.)

- Prediction as **inference** in this network



$$P(x[M+1] \mid x[1],\ldots,x[M])$$
$$= \int P(x[M+1] \mid \theta, x[1],\ldots,x[M]) P(\theta \mid x[1],\ldots,x[M]) d\theta$$
$$= \int P(x[M+1] \mid \theta) P(\theta \mid x[1],\ldots,x[M]) d\theta$$

where

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

Likelihood

Prior

Posterior

Probability of data

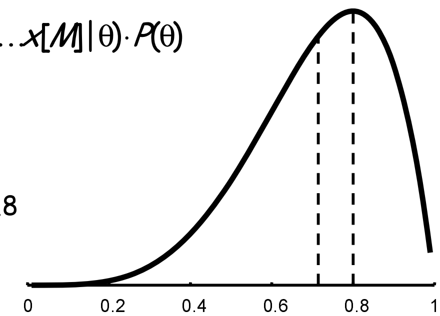
MP1-46

Example: Binomial Data Revisited

- Suppose that we choose a uniform prior $P(\theta) = 1$ for θ in [0,1]
- Then $P(\theta \mid D)$ is proportional to 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



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

Bayesian Inference and MLE

- In our example, MLE and Bayesian prediction differ

But…

If prior is well-behaved

- Does not assign 0 density to any “feasible” parameter value

  Then: both MLE and Bayesian prediction converge to the same value
- Both converge to the “true” underlying distribution (almost surely)

Dirichlet Priors

◆Recall that the likelihood function is

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

◆A **Dirichlet** prior with hyperparameters $\alpha_1,\ldots,\alpha_K$ is defined as

$$P(\Theta) \propto \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$ for legal $\theta_1,\ldots,\theta_K$

Then 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 (cont.)

◆We can compute the prediction on a new event in closed form:

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

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

Since the posterior is also Dirichlet, we get

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

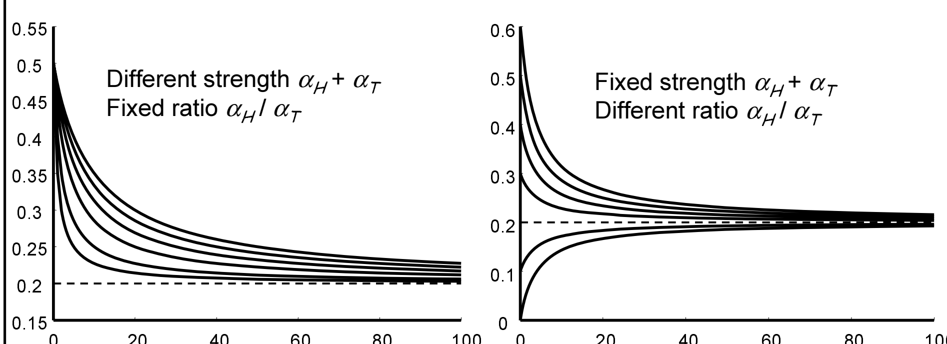
Priors Intuition

- The hyperparameters $\alpha_1,\ldots,\alpha_K$ can be thought of as "imaginary" counts from our prior experience
- Equivalent sample size = $\alpha_1+\ldots+\alpha_K$
- The larger the **equivalent sample size** the more confident we are in our prior

MP1-51

Effect of Priors

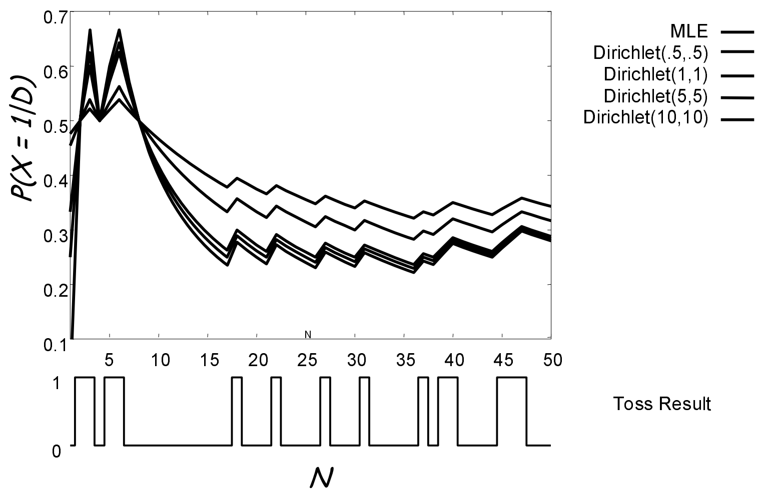
Prediction of $P(X=H)$ after seeing data with $N_H = 0.25 \cdot N_T$ for different sample sizes



MP1-52

Effect of Priors (cont.)

- In real data, Bayesian estimates are less sensitive to noise in the data



MP1-53

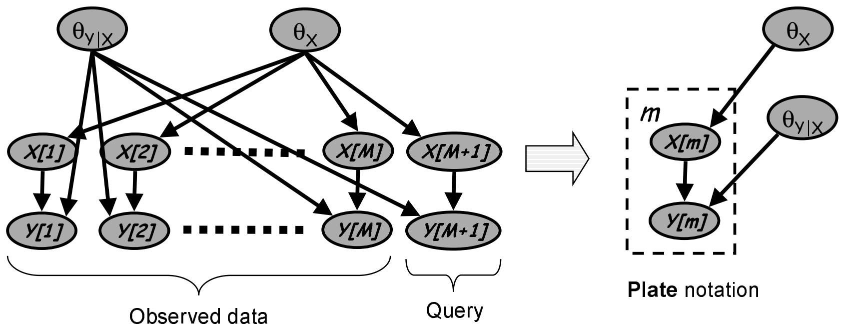
Conjugate Families

- The property that the posterior distribution follows the same parametric form as the prior distribution is called *conjugacy*
  - Dirichlet prior is a *conjugate family* for the multinomial likelihood

- Conjugate families are useful since:
  - For many distributions we can represent them with hyperparameters
  - They allow for sequential update within the same representation
  - In many cases we have closed-form solution for prediction

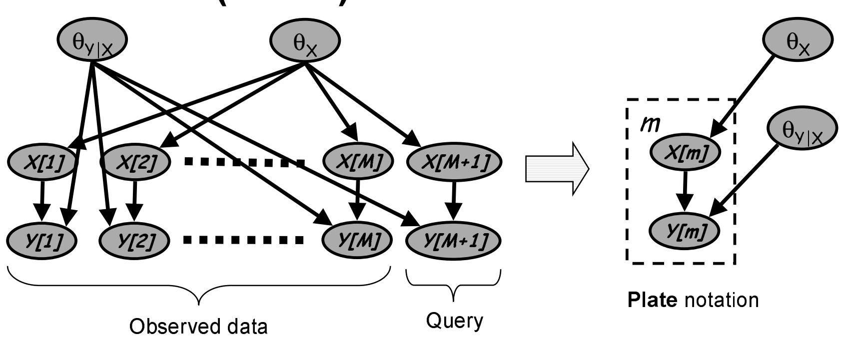
MP1-54

Bayesian Networks and Bayesian Prediction



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

Bayesian Networks and Bayesian Prediction (Cont.)



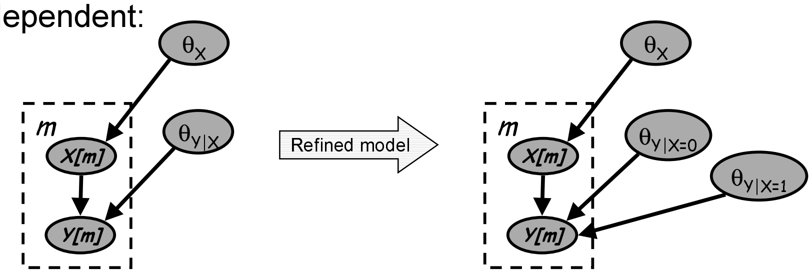
- We can also "read" from the network:

Complete data $\Rightarrow$

posteriors on parameters are independent

Bayesian Prediction(cont.)

- Since posteriors on parameters for each family are independent, we can compute them separately
- Posteriors for parameters within families are also independent:



- Complete data ⇒ the posteriors on $\theta_{Y|X=0}$ and $\theta_{Y|X=1}$ are independent

Bayesian Prediction(cont.)

- Given these observations, we can compute the posterior for each multinomial $\theta_{X_i \mid pa_i}$ independently
  - The posterior is Dirichlet with parameters
    $\alpha(X_i=1|pa_i)+N(X_i=1|pa_i),\ldots, \alpha(X_i=k|pa_i)+N(X_i=k|pa_i)$
- The predictive distribution is then represented by the parameters

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

which is what we expected!

The Bayesian analysis just made the assumptions explicit

Assessing Priors for Bayesian Networks

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

◆We can use initial parameters Θ_0 as prior information

- Need also an *equivalent sample size* parameter M_0
- Then, we let $\alpha(x_i,pa_i) = M_0 \bullet P(x_i,pa_i|\Theta_0)$

◆This allows to *update* a network using new data

MP1-59

Learning Parameters: Case Study (cont.)

◆Experiment:

- Sample a stream of instances from the alarm network
- Learn parameters using
  - MLE estimator
  - Bayesian estimator with uniform prior with different strengths

MP1-60

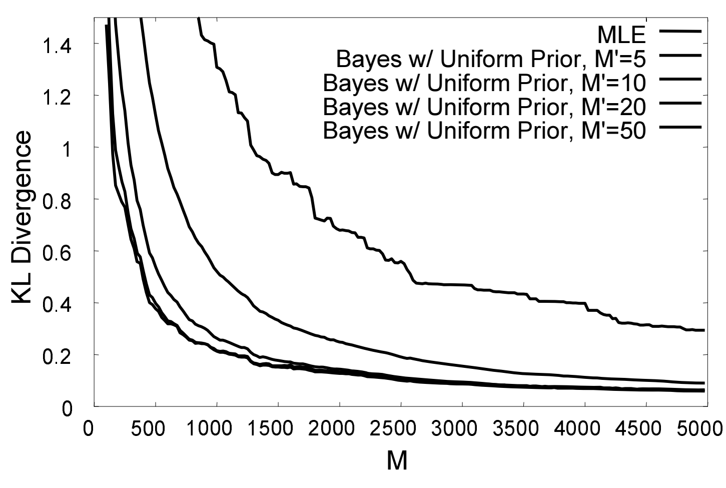
Learning Parameters: Case Study (cont.)

Comparing two distribution *P(x)* (true model) vs. *Q(x)* (learned distribution) -- Measure their **KL Divergence**

$$KL(P||Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)}$$

- 1 KL divergence (when logs are in base 2) =
  - The probability *P* assigns to an instance will be, on average, twice as small as the probability *Q* assigns to it
- $KL(P||Q) \geq 0$
- $KL(P||Q) = 0$ iff are P and Q equal

Learning Parameters: Case Study (cont.)



Learning Parameters: Summary

- Estimation relies on **sufficient statistics**
  - For multinomial these are of the form $N(x_i, pa_i)$
  - 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)

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

MP1-63

Outline

	Known Structure	Unknown Structure
Complete data		
Incomplete data	●	

- Introduction
- Bayesian networks: a review
- Parameter learning: Complete data

» Parameter learning: Incomplete data

- Structure learning: Complete data
- Application: classification
- Learning causal relationships
- Structure learning: Incomplete data
- Conclusion

MP1-64

Incomplete Data

Data is often **incomplete**

◆Some variables of interest are not assigned value

This phenomena happen when we have

◆Missing values

◆Hidden variables

Missing Values

◆**Examples:**

◆Survey data

◆Medical records

- Not all patients undergo all possible tests

Missing Values (cont.)

Complicating issue:

- The fact that a value is missing might be indicative of its value
  - The patient did not undergo X-Ray since she complained about fever and not about broken bones….

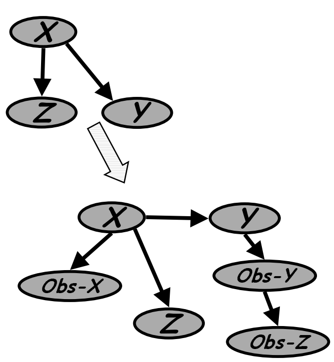
To learn from incomplete data we need the following assumption:

Missing at Random (MAR):

- The probability that the value of X_i is missing is independent of its actual value **given other observed values**

Missing Values (cont.)

- If MAR assumption does not hold, we can create new variables that ensure that it does
- We now can predict new examples (w/ pattern of ommisions)
- We might not be able to learn about the underlying process



Data

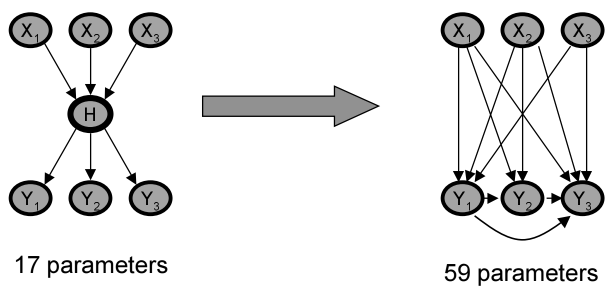
X	Y	Z
H	?	T
T	?	?
H	H	?
H	T	T
T	T	H

Augmented Data

X	Y	Z	Obs-X	Obs-Y	Obs-Z
H	?	T	Y	N	Y
T	?	?	Y	N	N
H	H	?	Y	Y	N
H	T	T	Y	Y	Y
T	T	H	Y	Y	Y

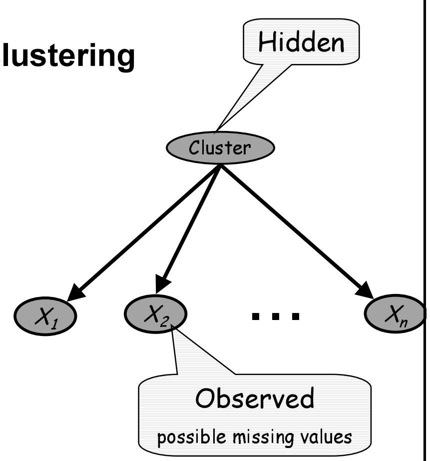
Hidden (Latent) Variables

- Attempt to learn a model with variables we never observe
  - In this case, MAR always holds
- Why should we care about unobserved variables?

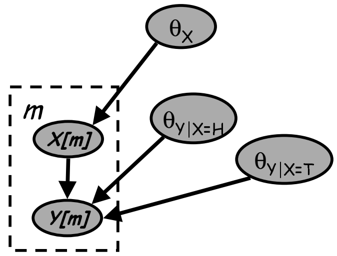


Hidden Variables (cont.)

- Hidden variables also appear in **clustering**
- **Autoclass** model:
  - Hidden variables assigns class labels
  - Observed attributes are independent given the class



Learning Parameters from Incomplete Data



Complete data:

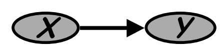
- Independent posteriors for θ_X, $\theta_{Y|X=H}$ and $\theta_{Y|X=T}$

Incomplete data:

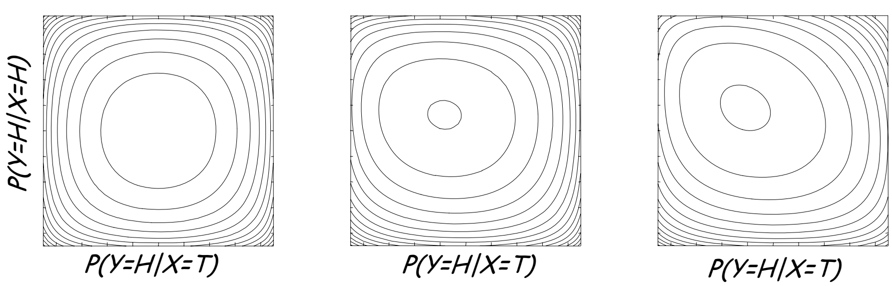
- Posteriors can be interdependent
- Consequence:
  - ML parameters can **not** be computed separately for each multinomial
  - Posterior is **not** a product of independent posteriors

MP1-71

Example



- Simple network:
- *P(X)* assumed to be known
- Likelihood is a function of 2 parameters: *P(Y=H|X=H), P(Y=H|X=T)*
- Contour plots of log likelihood for different number of missing values of *X* *(M = 8)*:

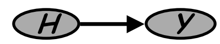


no missing values **2 missing value** **3 missing values**

MP1-72

Learning Parameters from Incomplete Data (cont.).

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



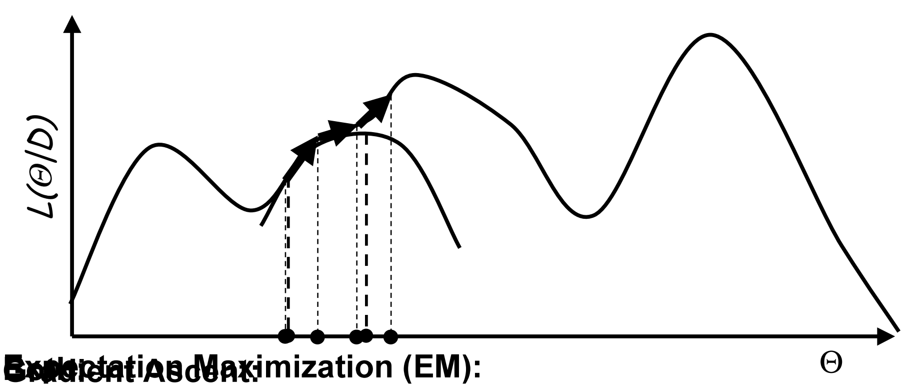
- **Example:**
  - We can rename the values of hidden variable H
  - If H has two values, likelihood has two global maxima

- Similarly, local maxima are also replicated
- Many hidden variables ⇒ a serious problem

MP1-73

MLE from Incomplete Data

- Finding MLE parameters: **nonlinear optimization** problem



Expectation Maximization (EM):

- Use "current point" to construct alternative function (which is "nice")
- Guaranty: maximum of new function is better scoring than the current point

Gradient Ascent:

- Follow gradient of likelihood w.r.t. to parameters
- Add line search and conjugate gradient methods to get fast convergence
- Require computations in each iteration

MP1-74

Gradient Ascent

◆Main result
$$\frac{\partial \log P(D \mid \Theta)}{\partial \theta_{x_i, pa_i}} = \frac{1}{\theta_{x_i, pa_i}} \sum_m P(x_i, pa_i \mid o[m], \Theta)$$

◆Requires computation: $P(x_i, Pa_i | o[m], \Theta)$ for all i, m

◆**Pros:**
- Flexible
- Closely related to methods in neural network training

◆**Cons:**
- Need to project gradient onto space of legal parameters
- To get reasonable convergence we need to combine with “smart” optimization techniques

Expectation Maximization (EM)

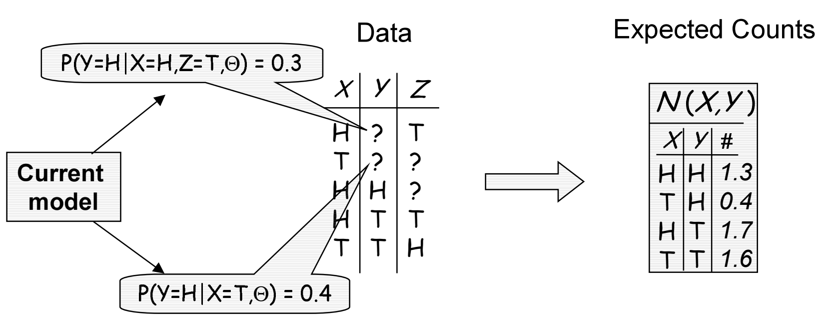
◆A general purpose method for learning from incomplete data

Intuition:

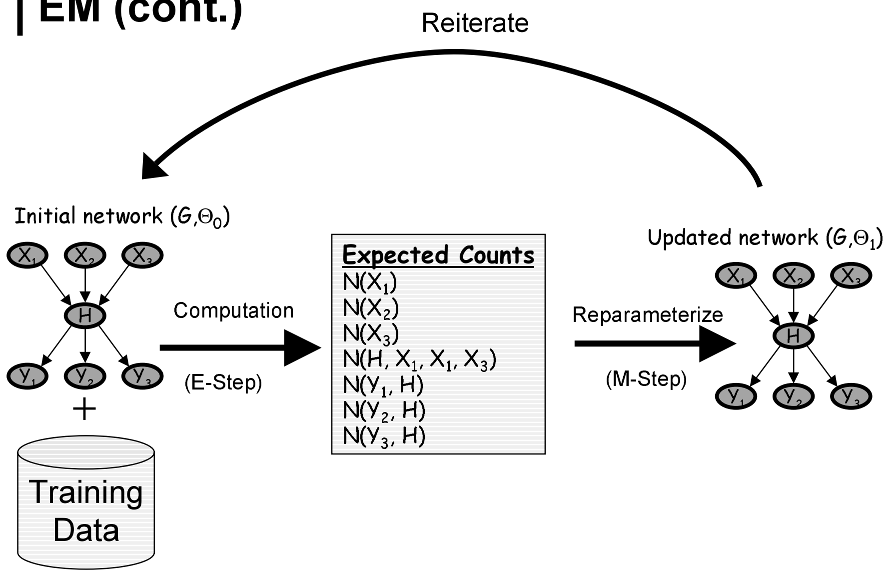
◆If we had access to counts, then we can estimate parameters

◆However, missing values do not allow to perform counts

◆“Complete” counts using current parameter assignment



EM (cont.)



EM (cont.)

Formal Guarantees:

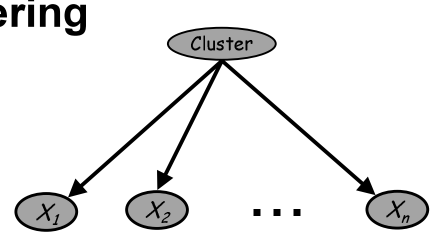
- $L(\Theta_1:D) \geq L(\Theta_0:D)$
  - Each iteration improves the likelihood

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

Main cost:

- Computations of expected counts in E-Step
- Requires a computation pass for each instance in training set
  - These are exactly the same as for gradient ascent!

Example: EM in clustering



- Consider clustering example

E-Step:

- Compute $P(C[m]|X_1[m],\ldots,X_n[m],\Theta)$
- This corresponds to "soft" assignment to clusters
- Compute expected statistics:

$$E[N(x_i,c)] = \sum_{m, X_i[m]=x_i} P(c \mid x_1[m],\ldots,x_n[m],\Theta)$$

M-Step

- Re-estimate $P(X_i|C)$, $P(C)$

MP1-79

EM in Practice

Initial parameters:

- Random parameters setting
- "Best" guess from other source

Stopping criteria:

- Small change in likelihood of data
- Small change in parameter values

Avoiding bad local maxima:

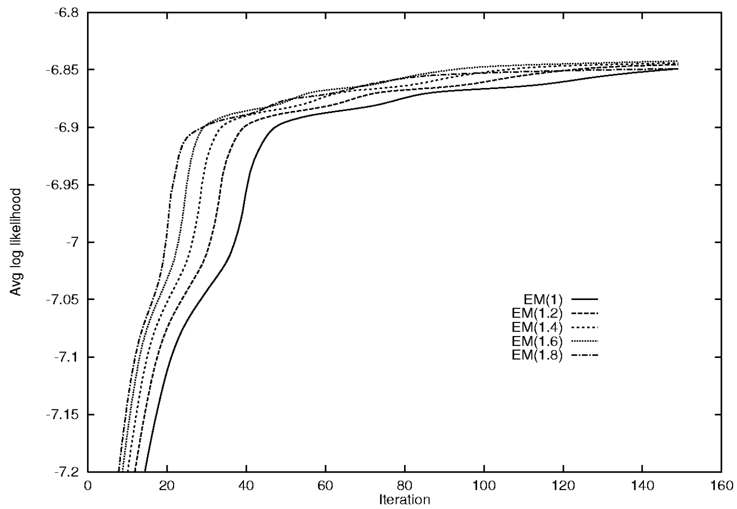
- Multiple restarts
- Early "pruning" of unpromising ones

Speed up:

- **various methods to speed convergence**

MP1-80

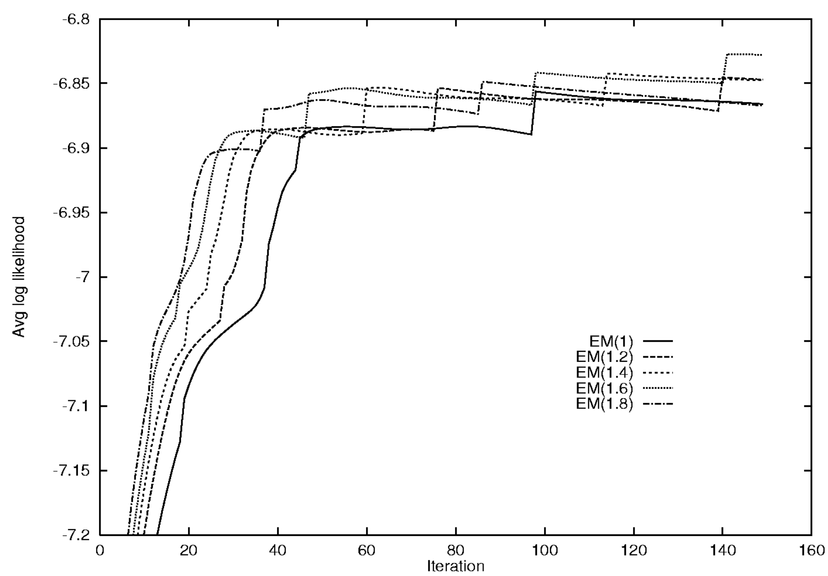
Error on training set (Alarm)



Experiment by Baur, Koller and Singer [UAI97]

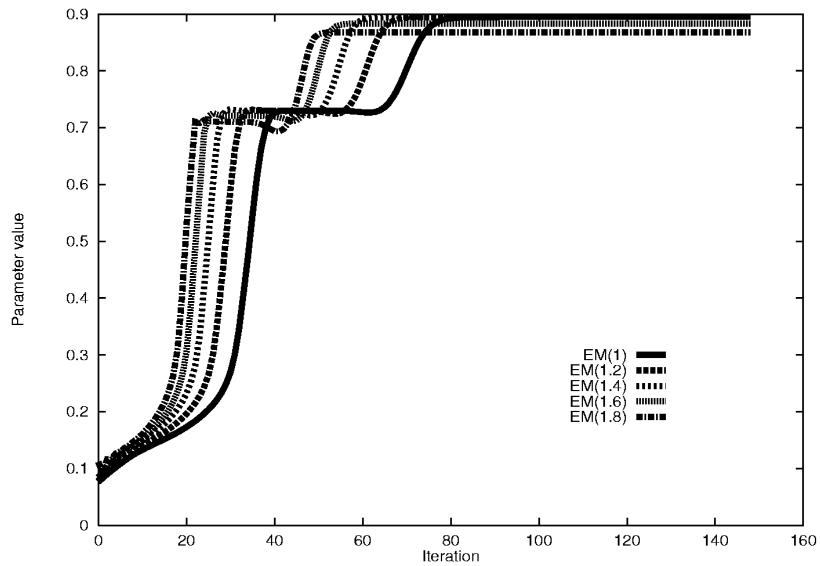
MP1-81

Test set error (alarm)



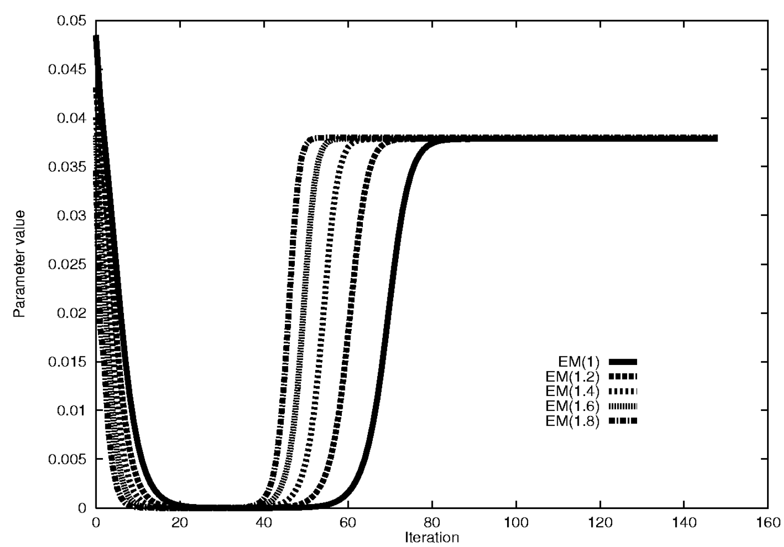
MP1-82

Parameter value (Alarm)



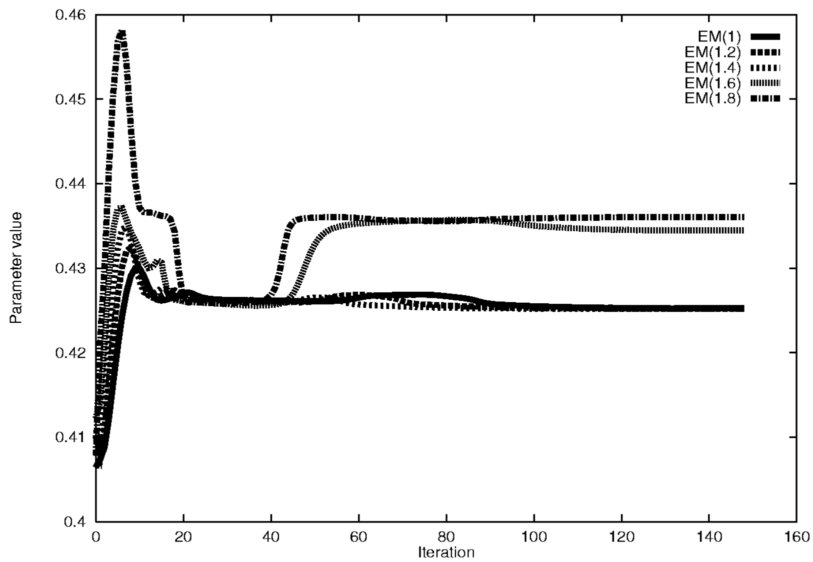
MP1-83

Parameter value (Alarm)



MP1-84

Parameter value (Alarm)



MP1-85

Bayesian Inference with Incomplete Data

Recall, Bayesian estimation:

$$P(x[M+1] \mid D) = \int P(x[M+1] \mid \theta) P(\theta \mid D) d\theta$$

Complete data: closed form solution for integral

Incomplete data:

◆No sufficient statistics (except the data)

◆Posterior does not decompose

◆No closed form solution

⇨Need to use approximations

MP1-86

MAP Approximation

- Simplest approximation: MAP parameters
  - MAP --- **Maximum A-posteriori Probability**

$$P(x[M+1] \mid D) \approx P(x[M+1] \mid \tilde{\theta})$$

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

Assumption:

- Posterior mass is dominated by a MAP parameters

Finding MAP parameters:

- Same techniques as finding ML parameters
- Maximize $P(\theta|D)$ instead of $L(\theta{:}D)$

Stochastic Approximations

Stochastic approximation:

- Sample $\theta_1, \ldots, \theta_k$ from $P(\theta|D)$
- Approximate

$$P(x[M+1] \mid D) \approx \frac{1}{k} \sum_i P(x[M+1] \mid \theta_i)$$

Stochastic Approximations (cont.)

How do we sample from $P(\theta|D)$?

Markov Chain Monte Carlo (MCMC) methods:

- Find a Markov Chain whose stationary probability Is $P(\theta|D)$
- Simulate the chain until convergence to stationary behavior
- Collect samples for the "stationary" regions

Pros:

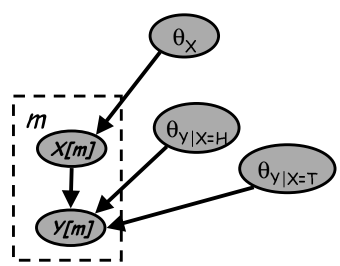
- Very flexible method: when other methods fails, this one usually works
- The more samples collected, the better the approximation

Cons:

- Can be computationally expensive
- How do we know when we are converging on stationary distribution?

MP1-89

Stochastic Approximations: Gibbs Sampling



Gibbs Sampler:

- A simple method to construct MCMC sampling process

Start:

- Choose (random) values for all unknown variables

Iteration:

- Choose an unknown variable
  - A missing data variable or unknown parameter
  - Either a random choice or round-robin visits
- Sample a value for the variable given the current values of all other variables

MP1-90

Parameter Learning from Incomplete Data: Summary

- ◆Non-linear optimization problem
- ◆Methods for learning: EM and Gradient Ascent
  - Exploit inference for learning

Difficulties:

- ◆Exploration of a complex likelihood/posterior
  - More missing data ⇒ many more local maxima
  - Cannot represent posterior ⇒ must resort to approximations
- ◆Inference
  - Main computational bottleneck for learning
  - Learning large networks
    ⇒ exact inference is infeasible
    ⇒ resort to stochastic simulation or approximate inference
    (e.g., see Jordan's tutorial)

Outline

	Known Structure	Unknown Structure
Complete data		●
Incomplete data		

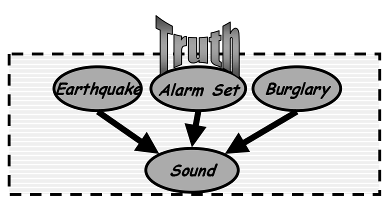
Benefits of Learning Structure

- Efficient learning -- more accurate models with less data
  - Compare: $P(A)$ and $P(B)$ vs joint $P(A,B)$
    former requires less data!
  - Discover structural properties of the domain
  - Identifying independencies in the domain helps to
    - Order events that occur sequentially
    - Sensitivity analysis and inference
- Predict effect of actions
  - Involves learning causal relationship among variables

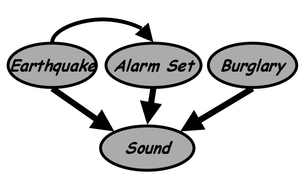
  ⇒defer to later part of the tutorial

MP1-93

Why Struggle for Accurate Structure

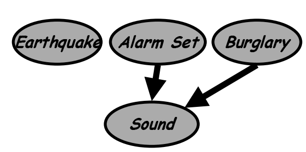


Adding an arc



- Increases the number of parameters to be fitted
- Wrong assumptions about causality and domain structure

Missing an arc



- Cannot be compensated by accurate fitting of parameters
- Also misses causality and domain structure

MP1-94

Approaches to Learning Structure

- **Constraint based**
  - Perform tests of conditional independence
  - Search for a network that is consistent with the observed dependencies and independencies
- **Score based**
  - Define a score that evaluates how well the (in)dependencies in a structure match the observations
  - Search for a structure that maximizes the score

Constraints versus Scores

- **Constraint based**
  - Intuitive, follows closely the definition of BNs
  - Separates structure construction from the form of the independence tests
  - Sensitive to errors in individual tests
- **Score based**
  - Statistically motivated
  - Can make compromises
- **Both**
  - Consistent---with sufficient amounts of data and computation, they learn the correct structure

Likelihood Score for Structures

First cut approach:

- Use likelihood function

◆Recall, the likelihood score for a network structure and parameters is

$$L(G,\Theta_G : D) = \prod_m P(x_1[m],\ldots,x_n[m] : G,\Theta_G)$$
$$= \prod_m \prod_i P(x_i[m] \mid Pa_i^G[m] : G,\Theta_{G,i})$$

◆Since we know how to maximize parameters from now we assume $L(G : D) = \max_{\Theta_G} L(G,\Theta_G : D)$

Likelihood Score for Structure (cont.)

Rearranging terms:

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

where

◆$H(X)$ is the **entropy** of X

◆$I(X;Y)$ is the **mutual information** between X and Y

- $I(X;Y)$ measures how much “information” each variables provides about the other
- $I(X;Y) \geq 0$
- $I(X;Y) = 0$ iff X and Y are independent
- $I(X;Y) = H(X)$ iff X is totally predictable given Y

Likelihood Score for Structure (cont.)

$$l(G : D) = M \sum_i \left(I(X_i ; Pa_i^G) - H(X_i) \right)$$

Good news:

- ◆Intuitive explanation of likelihood score:
  - The larger the dependency of each variable on its parents, the higher the score
  - Likelihood as a compromise among dependencies, based on their strength

Bad news:

- ◆Adding arcs always helps
  - $I(X;Y) \leq I(X;Y,Z)$
  - Maximal score attained by "complete" networks
  - Such networks can overfit the data --- the parameters they learn capture the noise in the data

MP1-99

Avoiding Overfitting

"Classic" issue in learning.

Standard approaches:

- ◆Restricted hypotheses
  - Limits the overfitting capability of the learner
  - Example: restrict # of parents or # of parameters
- ◆Minimum description length
  - Description length measures complexity
  - Choose model that compactly describes the training data
- ◆Bayesian methods
  - Average over all possible parameter values
  - Use prior knowledge

MP1-100

Avoiding Overfitting (cont..)

Other approaches include:

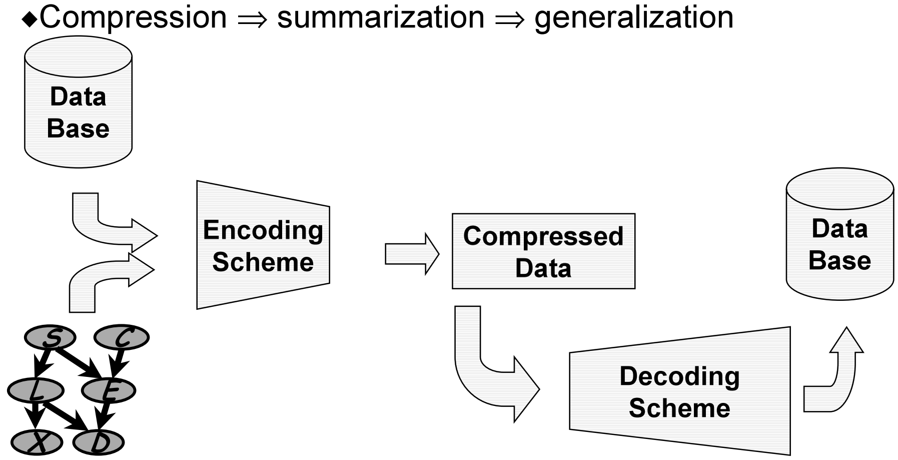
- Holdout/Cross-validation/Leave-one-out
  - Validate generalization on data withheld during training
- Structural Risk Minimization
  - Penalize hypotheses subclasses based on their VC dimension

MP1-101

Minimum Description Length

Rationale:

- prefer networks that facilitate compression of the data
- Compression ⇒ summarization ⇒ generalization



MP1-102

Minimum Description Length (cont.)

- Computing the description length of the data, we get

$$DL(D:G) = DL(G) + \frac{\log M}{2}\dim(G) - l(G:D)$$

- $DL(G)$: # bits to encode G
- $\frac{\log M}{2}\dim(G)$: # bits to encode Θ_G
- $l(G:D)$: # bits to encode D using (G,Θ_G)

- Minimizing this term is equivalent to maximizing

$$MDL(G:D) = l(G:D) - \frac{\log M}{2}\dim(G) - DL(G)$$

MP1-103

Minimum Description: Complexity Penalization

$$MDL(G:D) = l(G:D) - \frac{\log M}{2}\dim(G) - DL(G)$$

- Likelihood is (roughly) **linear** in M

$$l(G:D) = \sum_m \log P(x[m] \mid G, \hat{\Theta})$$
$$\approx M \cdot E[\log P(x \mid G, \hat{\Theta})]$$

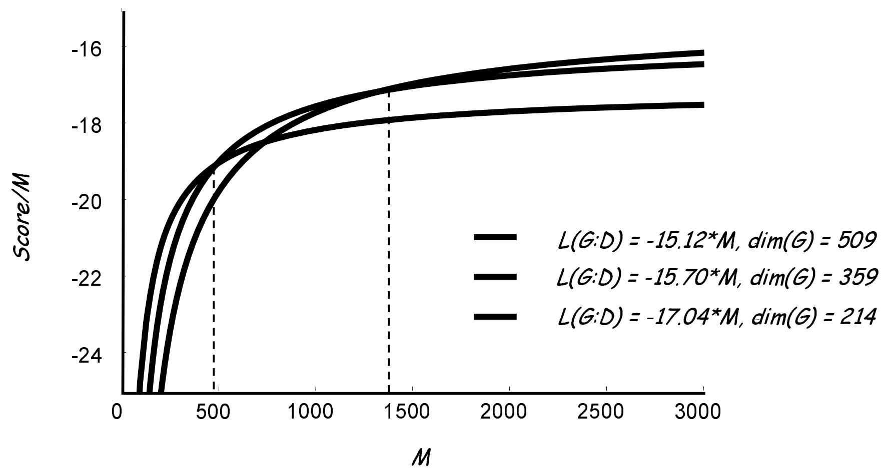
- Penalty is **logarithmic** in M

As we get more data, the penalty for complex structure is less harsh

MP1-104

Minimum Description: Example

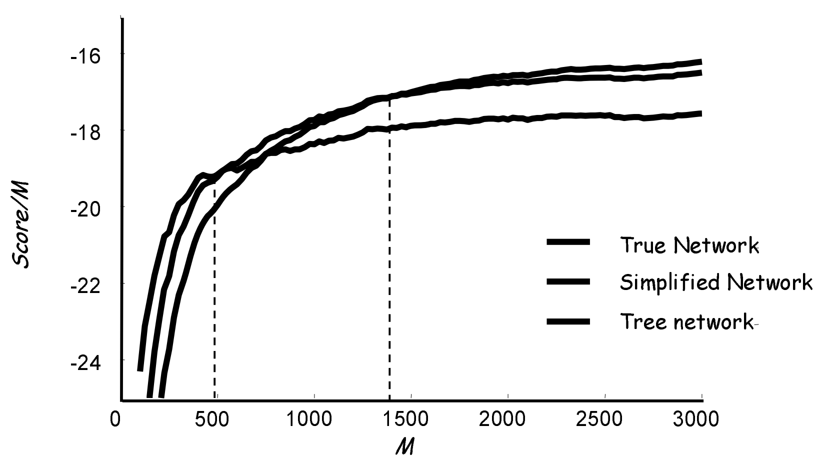
- Idealized behavior:



Minimum Description: Example (cont.)

Real data illustration with three network:

- "True" alarm (509 param), simplified (359 param), tree (214 param)



Consistency of the MDL Score

MDL Score is **consistent**

- As $M \to \infty$ the “true” structure G^* maximizes the score (almost surely)
- For sufficiently large M, the maximal scoring structures are **equivalent** to G^*

Proof (outline):

- Suppose G implies an independence statement not in G*, then
  as $M \to \infty$, $l(G:D) \to l(G^*:D) - eM$ (e depends on G)
  so $MDL(G^*:D) - MDL(G:D) \to eM - (dim(G^*)-dim(G))/2 \log M$
- Now suppose G* implies an independence statement not in G, then
  as $M \to \infty$, $l(G:D) \to l(G^*:D)$
  so $MDL(G:D) - MDL(G^*:D) \to (dim(G)-dim(G^*))/2 \log M$

MP1-107

Bayesian Inference

- Bayesian Reasoning---compute expectation over unknown G

$$P(x[M+1] \mid D) = \sum_G P(x[M+1] \mid D, G) P(G \mid D)$$

where

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

Marginal likelihood: $P(D \mid G)$
Prior over structures: $P(G)$
Posterior score: $P(G \mid D)$
Likelihood: $P(D \mid G, \theta)$
Prior over parameters: $P(\theta \mid G)$

Assumption: Gs are mutually exclusive and exhaustive

MP1-108

Marginal Likelihood: Binomial case

- Assume we observe a sequence of coin tosses….
- By the chain rule we have:

$$P(x[1],\ldots,x[M]) = P(x[1])P(x[2] \mid x[1])\cdots P(x[M] \mid x[1],\ldots,x[M-1])$$

recall that

$$P(x[m+1] = H \mid x[1],\ldots,x[m]) = \frac{N_H^m + \alpha_H}{m + \alpha_H + \alpha_T}$$

where N^m_H is the number of heads in first m examples.

MP1-109

Marginal Likelihood: Binomials (cont.)

$$P(x[1],\ldots,x[M]) = \frac{\alpha_H}{\alpha_H+\alpha_T}\cdots\frac{N_H-1+\alpha_H}{N_H-1+\alpha_H+\alpha_T}\cdot \frac{\alpha_T}{N_H+\alpha_H+\alpha_T}\cdots\frac{N_T-1+\alpha_T}{N_H+N_T-1+\alpha_H+\alpha_T}$$

We simplify this by using $(\alpha)(1+\alpha)\cdots(N-1+\alpha) = \frac{\Gamma(N+\alpha)}{\Gamma(\alpha)}$

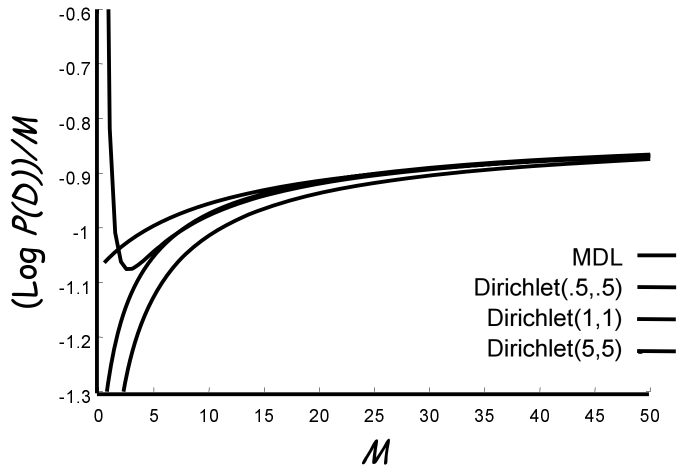
Thus

$$P(x[1],\ldots,x[M]) = \frac{\Gamma(\alpha_H+\alpha_T)}{\Gamma(\alpha_H+\alpha_T+N_H+N_T)}\frac{\Gamma(\alpha_H+N_H)}{\Gamma(\alpha_H)}\frac{\Gamma(\alpha_T+N_T)}{\Gamma(\alpha_T)}$$

MP1-110

Binomial Likelihood: Example

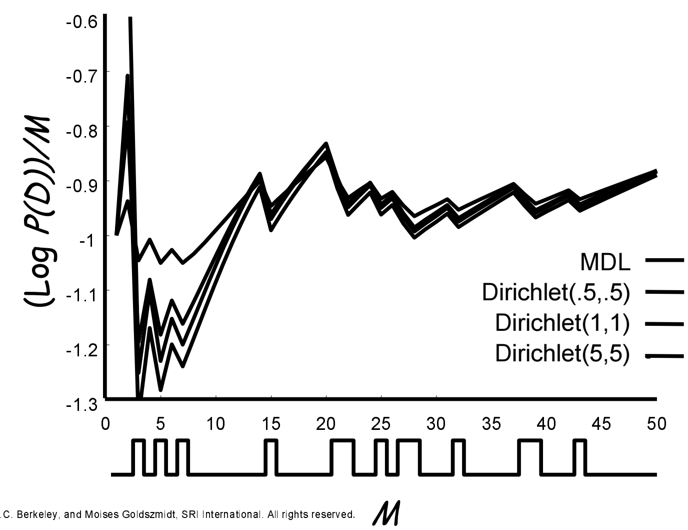
- Idealized experiment with *P(H) = 0.25*



MP1-111

Marginal Likelihood: Example (cont.)

- Actual experiment with *P(H) = 0.25*



MP1-112

Marginal Likelihood: Multinomials

The same argument generalizes to multinomials with Dirichlet prior

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

Then

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

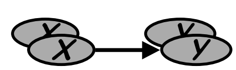
Marginal Likelihood: Bayesian Networks

- 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: / Network 2:

- Two Dirichlet marginal likelihoods / Three Dirichlet marginal likelihoods
- $P(X[1],\dots,X[7])$
- $P(Y[1],\dots,Y[7])$ / $P(Y[1],Y[4],Y[6],Y[7])$
- $P(Y[2],Y[3],Y[5])$



Marginal Likelihood (cont.)

In general networks, the marginal likelihood has the form:

$$P(D \mid G) = \prod_i \prod_{pa_i^G} \underbrace{\frac{\Gamma(\alpha(pa_i^G))}{\Gamma(\alpha(pa_i^G) + N(pa_i^G))} \prod_{x_i} \frac{\Gamma(\alpha(x_i, pa_i^G) + N(x_i, pa_i^G))}{\Gamma(\alpha(x_i, pa_i^G))}}$$

Dirichlet Marginal Likelihood
For the sequence of values of X_i when
X_i's parents have a particular value

- where
- *N(..)* are the counts from the data
- α*(..)* are the hyperparameters for each family **given *G***

MP1-115

Priors and BDe score

- We need: prior counts $\alpha(..)$ for each network structure G
- This can be a formidable task
  - There are exponentially many structures…

Possible solution: The BDe prior

- Use prior of the form M_0, $B_0=(G_0, \Theta_0)$
  - Corresponds to M_0 prior examples distributed according to B_0
- Set $\alpha(x_i,pa_i^G) = M_0 P(x_i,pa_i^G \mid G_0, \Theta_0)$
  - Note that pa_i^G are, in general, not the same as the parents of X_i in G_0. We can compute this using standard BN tools
- This choice also has desirable theoretical properties
  - Equivalent networks are assigned the same score

MP1-116

Bayesian Score: Asymptotic Behavior

- The Bayesian score seems quite different from the MDL score
- However, the two scores are asymptotically equivalent

Theorem: If the prior $P(\Theta \mid G)$ is "well-behaved", then

$$\log P(D \mid G) = l(G : D) - \frac{\log M}{2} \dim(G) + O(1)$$

Proof:

- **(Simple)** Use Stirling's approximation to $\Gamma(\,)$
  - Applies to Bayesian networks with Dirichlet priors
- **(General)** Use properties of exponential models and Laplace's method for approximating integrals
  - Applies to Bayesian networks with other parametric families

Bayesian Score: Asymptotic Behavior

Consequences:

- Bayesian score is asymptotically equivalent to MDL score
  - The terms $\log P(G)$ and description length of G are constant and thus they are negligible when M is large.
- Bayesian score is **consistent**
  - Follows immediately from consistency of MDL score
- Observed data eventually overrides prior information
  - Assuming that the prior does not assign probability 0 to some parameter settings

Scores -- Summary

- Likelihood, MDL and (log) BDe have the form

$$Score(G : D) = \sum_i Score(X_i \mid Pa_i^G : N(X_i Pa_i))$$

- BDe requires assessing prior network. It can naturally incorporate prior knowledge and previous experience
- Both MDL and BDe are consistent and asymptotically equivalent (up to a constant)
- All three are **score-equivalent**---they assign the same score to equivalent networks

Outline

	Known Structure	Unknown Structure
Complete data		●
Incomplete data		

- Introduction
- Bayesian networks: a review
- Parameter learning: Complete data
- Parameter learning: Incomplete data
- »Structure learning: Complete data
  - Scoring metrics
  - » **Maximizing the score**
  - Learning local structure
- Application: classification
- Learning causal relationships
- Structure learning: Incomplete data
- Conclusion

Optimization Problem

Input:
- Training data
- Scoring function (including priors, if needed)
- Set of possible structures
  - Including prior knowledge about structure

Output:
- A network (or networks) that maximize the score

Key Property:
- **Decomposability**: the score of a network is a sum of terms.

Learning Trees

- **Trees:**
  - At most one parent per variable

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

Learning Trees (cont.)

- Let $p(i)$ denote the parent of X_i, or 0 if X_i has no parents
- We can write the score as

$$
\begin{aligned}
Score(G:D) &= \sum_i Score(X_i : Pa_i) \\
&= \sum_{i, p(i)>0} Score(X_i : X_{p(i)}) + \sum_{i, p(i)=0} Score(X_i) \\
&= \underbrace{\sum_{i, p(i)>0} \left(Score(X_i : X_{p(i)}) - Score(X_i) \right)}_{\text{Improvement over “empty” network}} + \underbrace{\sum_i Score(X_i)}_{\text{Score of “empty” network}}
\end{aligned}
$$

Improvement over “empty” network

Score of “empty” network

- Score = sum of edge scores + constant

Learning Trees (cont)

Algorithm:

- Construct graph with vertices: 1, 2, …
- Set $w(i \rightarrow j)$ be $Score(X_j \mid X_i) - Score(X_j)$
- Find tree (or forest) with maximal weight
  - This can be done using standard algorithms in low-order polynomial time by building a tree in a greedy fashion (Kruskal’s maximum spanning tree algorithm)

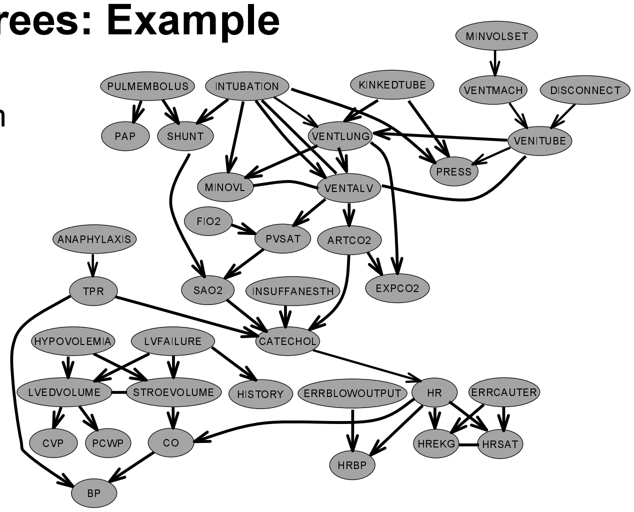
Theorem: This procedure finds the tree with maximal score

When score is likelihood, then $w(i \rightarrow j)$ is proportional to $I(X_i; X_j)$ this is known as the Chow & Liu method

Learning Trees: Example

Tree learned from alarm data

- ◆Green -- correct arcs
- ◆Red -- spurious arcs



- ◆Not every edge in tree is in the the original network
- ◆Tree direction is arbitrary --- we can't learn about arc direction

Beyond Trees

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

- ◆Suppose we allow two parents
- ◆A greedy algorithm is no longer guaranteed to find the optimal network

- ◆In fact, no efficient algorithm exists

Theorem: Finding maximal scoring network structure with at most k parents for each variables is NP-hard for $k > 1$

Heuristic Search

- We address the problem by using heuristic search

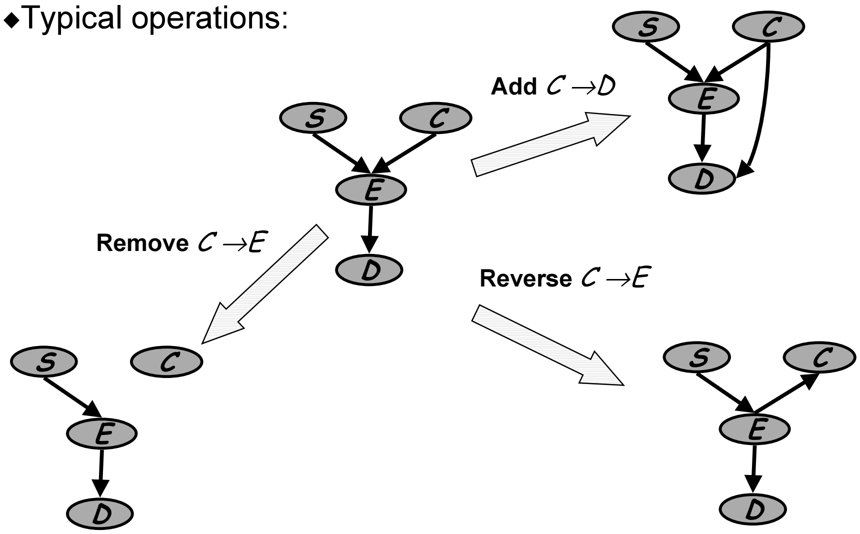
- Define a search space:
  - nodes are possible structures
  - edges denote adjacency of structures
- Traverse this space looking for high-scoring structures

Search techniques:
- Greedy hill-climbing
- Best first search
- Simulated Annealing
- ...

MP1-127

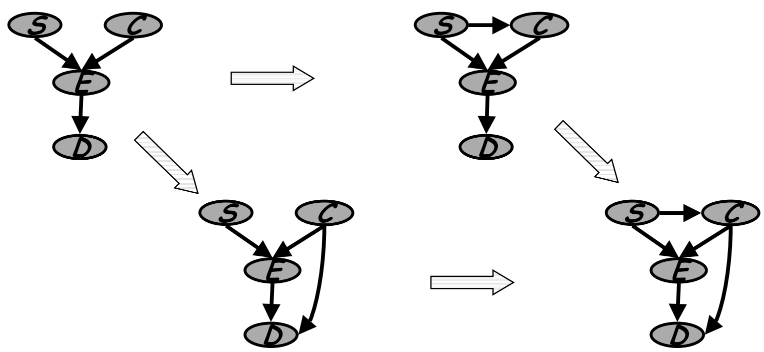
Heuristic Search (cont.)

- Typical operations:



MP1-128

Exploiting Decomposability in Local Search



◆**Caching:** To update the score of after a local change, we only need to re-score the families that were changed in the last move

Greedy Hill-Climbing

Simplest heuristic local search

- Start with a given network
  - empty network
  - best tree
  - a random network
- At each iteration
  - Evaluate all possible changes
  - Apply change that leads to best improvement in score
  - Reiterate
- Stop when no modification improves score

◆Each step requires evaluating approximately *n* new changes

Greedy Hill-Climbing (cont.)

- Greedy Hill-Climbing can get struck in:
  - **Local Maxima:**
    - All one-edge changes reduce the score
  - **Plateaus:**
    - Some one-edge changes leave the score unchanged
- Both are occur in the search space

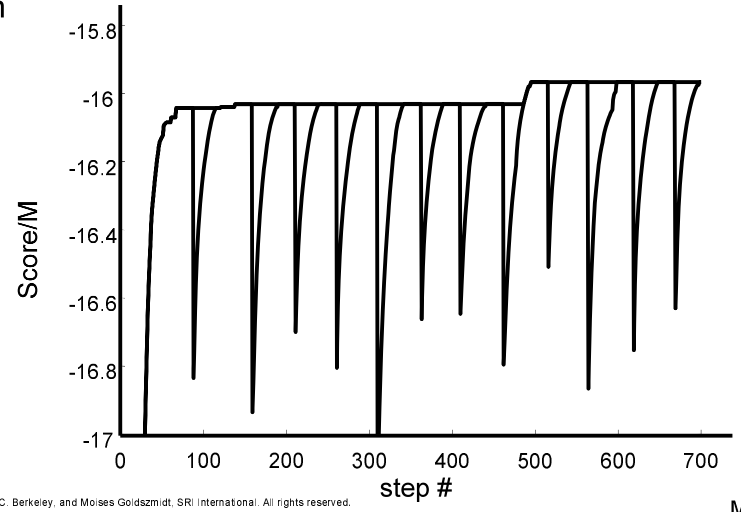
Greedy Hill-Climbing (cont.)

To avoid these problems, we can use:

- **TABU-search**
  - Keep list of K most recently visited structures
  - Apply best move that does not lead to a structure in the list
  - This escapes plateaus and local maxima and with “basin” smaller than K structures
- **Random Restarts**
  - Once stuck, apply some fixed number of random edge changes and restart search
  - This can escape from the basin of one maxima to another

Greedy Hill-Climbing

◆Greedy Hill Climbing with TABU-list and random restarts on alarm



Other Local Search Heuristics

◆**Stochastic First-Ascent Hill-Climbing**

- Evaluate possible changes at random
- Apply the first one that leads "uphill"
- Stop when a fix amount of "unsuccessful" attempts to change the current candidate

◆**Simulated Annealing**

- Similar idea, but also apply "downhill" changes with a probability that is proportional to the change in score
- Use a temperature to control amount of random downhill steps
- Slowly "cool" temperature to reach a regime where performing strict uphill moves

I-Equivalence Class Search

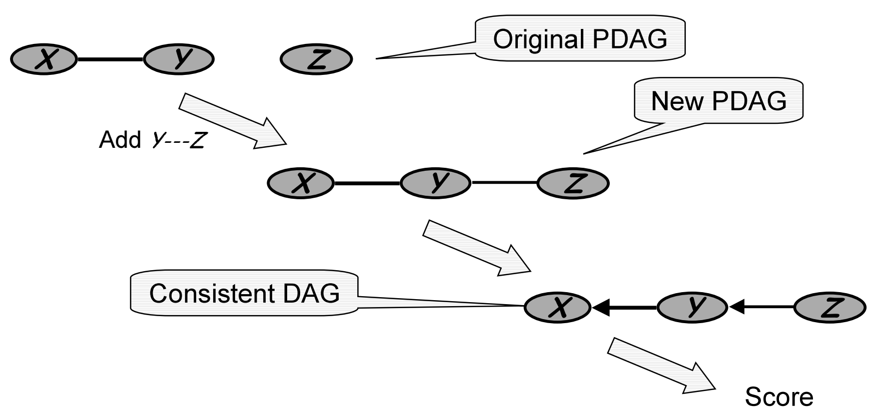
So far, we seen generic search methods…
- Can exploit the structure of our domain?

Idea:
- Search the space of I-equivalence classes
- Each I-equivalence class is represented by a PDAG (partially ordered graph) -- skeleton + v-structures

Benefits:
- The space of PDAGs has fewer local maxima and plateaus
- There are fewer PDAGs than DAGs

I-Equivalence Class Search (cont.)

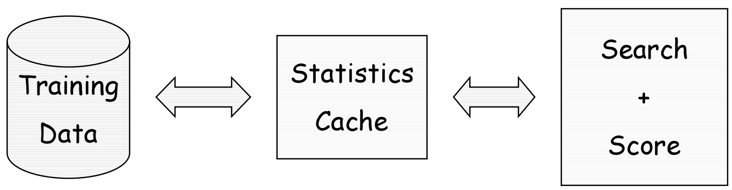
Evaluating changes is more expensive



- These algorithms are more complex to implement

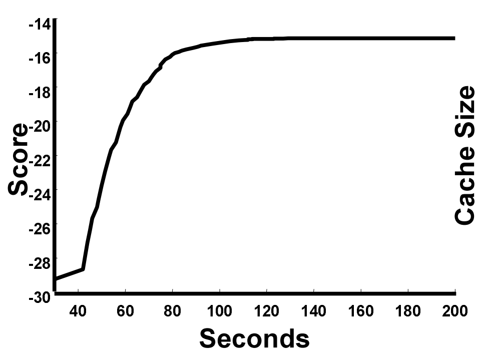
Search and Statistics

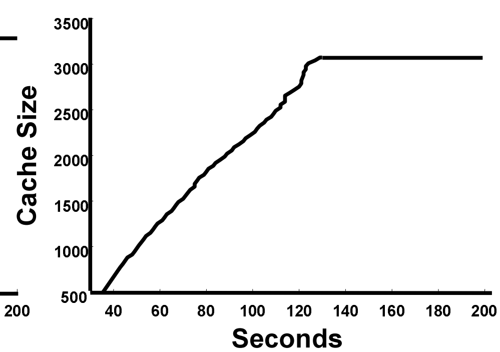
- Evaluating the score of a structure requires the corresponding counts (sufficient statistics)
- Significant computation is spent in collecting these counts
  - Requires a pass over the training data
- Reduce overhead by caching previously computed counts
  - Avoid duplicated efforts
  - Marginalize counts: $N(X,Y) \rightarrow N(X)$



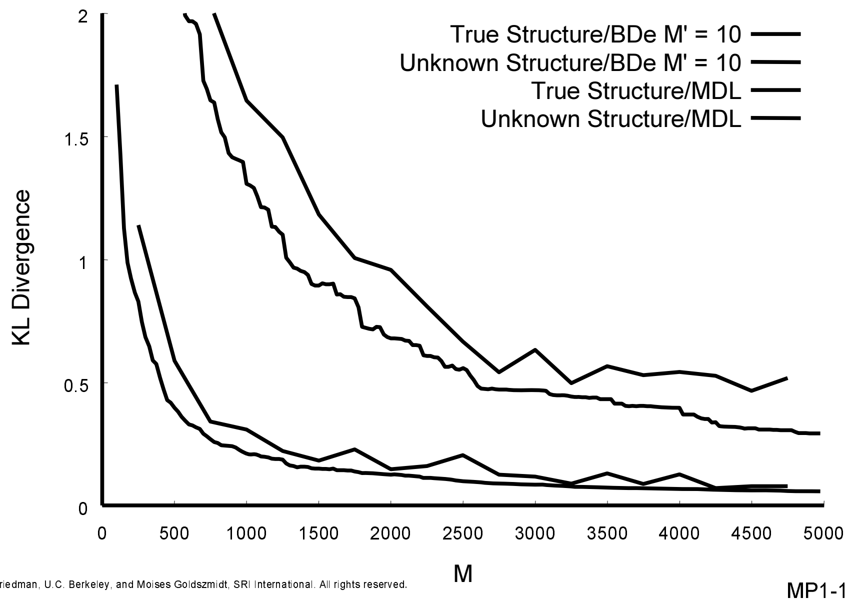
Learning in Practice: Time & Statistics

- Using greedy Hill-Climbing on 10000 instances from alarm





Learning in Practice: Alarm domain



MP1-139

Model Averaging

◆Recall, Bayesian analysis started with

$$P(x[M+1] \mid D) = \sum_G P(x[M+1] \mid D, G) P(G \mid D)$$

- This requires us to average over all possible models

MP1-140

Model Averaging (cont.)

- So far, we focused on single model
  - Find best scoring model
  - Use it to predict next example
- Implicit assumption:
  - Best scoring model dominates the weighted sum

- **Pros:**
  - We get a single structure
  - Allows for efficient use in our tasks
- **Cons:**
  - We are committing to the independencies of a particular structure
  - Other structures might be as probable given the data

Model Averaging (cont.)

Can we do better?

- **Full Averaging**
  - Sum over all structures
  - Usually intractable---there are exponentially many structures
- **Approximate Averaging**
  - Find K largest scoring structures
  - Approximate the sum by averaging over their prediction
  - Weight of each structure determined by the **Bayes Factor**

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

The actual score we compute

Search: Summary

- Discrete optimization problem
- In general, NP-Hard
  - Need to resort to heuristic search
  - In practice, search is relatively fast (~100 vars in ~10 min):
    - Decomposability
    - Sufficient statistics
- In some cases, we can reduce the search problem to an easy optimization problem
  - Example: learning trees

MP1-143

Outline

	Known Structure	Unknown Structure
Complete data		●
Incomplete data		

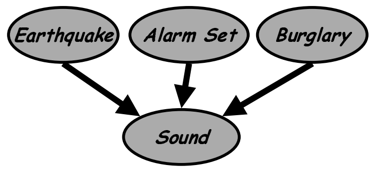
- Introduction
- Bayesian networks: a review
- Parameter learning: Complete data
- Parameter learning: Incomplete data

» Structure learning: Complete data
  - Scoring metrics
  - Maximizing the score
  - » **Learning local structure**
- Application: classification
- Learning causal relationships
- Structure learning: Incomplete data
- Conclusion

MP1-144

Local and Global Structure

Global structure

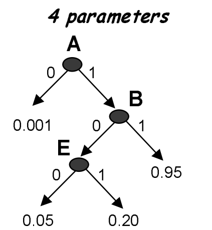


Explicitly represents
P(E|A) = P(E)

8 parameters

A	B	E	P(S=1|A,B,E)
1	1	1	.95
1	1	0	.95
1	0	1	.20
1	0	0	.05
0	1	1	.001
0	1	0	.001
0	0	1	.001
0	0	0	.001

Local structure



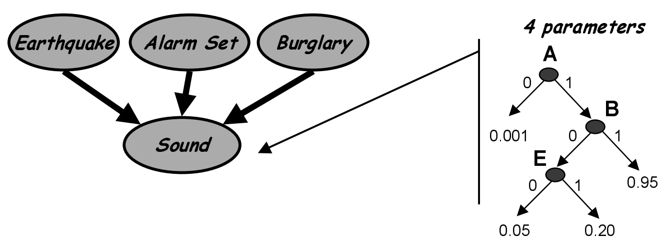
Explicitly represents
P(S|A = 0) = P(S|B,E,A=0)

MP1-145

Local structure: Decision trees

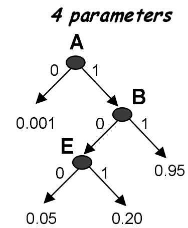
- Capture properties of **context specific independence**
  - *B* and *S* are independent given *A = false*

- Internal nodes: A tests on *X*'s parents values
- Leafs: Distribution on *X*



MP1-146

Learning decision trees



- Parameter learning:
  - As with tabular representations
  - Multinomial distribution at each leaf
  - Counts are at the level of leaves
- Structure learning
  - Define the MDL or marginal likelihood
  - General structure similar to scores of Bayesian networks

Effects on learning

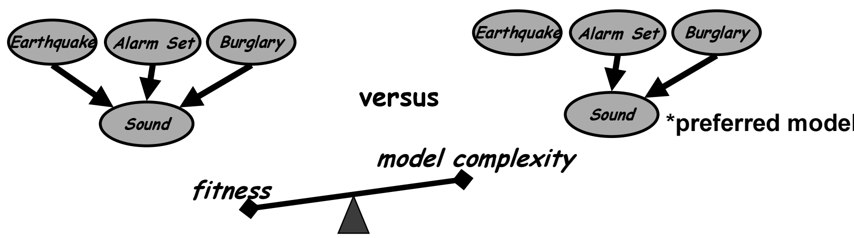
- Global structure:
  - Enables decomposability of the score
    - **Search is feasible**
- Local structure:
  - Reduces the number of parameters to be fitted
    - **Better estimates**
    - **More accurate global structure!**

Local Structure ⇒ More Accurate Global Structure

Without local structure...

Adding an arc may imply an exponential increase on the number of parameters to fit, independently of the relation between the variables

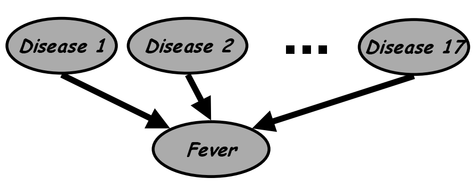
The score balancing act



MP1-149

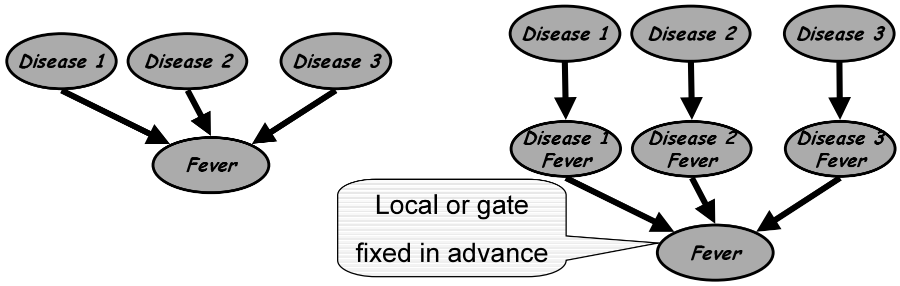
Local structure: Noisy Or

- Intuition: **Causal Independence**
  - Many possible causes that do not interact
    - Several diseases can cause fever;
      If one “succeeds”, the patient has the symptom



MP1-150

Local structure: Noise-Or decomposition



- Benefits:
  - Linear number of parameters
  - Good approximation for many domains
- Training:
  - Using missing data methods
  - Or gate parameters are fixed and not retrained

Other Types of Local Structure

- Extensions of trees: Graphs
- Extensions of Noisy-or: Noisy-max, Causal independence
- Regression
- Neural nets
- Continuous representations, such as Gaussians

Any type of representation that reduces the number of parameters to fit

- To "plug in" a different representation, we need the following
  - Sufficient Statistics
  - Estimation of parameters
  - Marginal likelihood

Outline

- ◆Introduction
- ◆Bayesian networks: a review
- ◆Parameter learning: Complete data
- ◆Parameter learning: Incomplete data
- ◆Structure learning: Complete data
- »Application: classification
- ◆Learning causal relationships
- ◆Structure learning: Incomplete data
- ◆Conclusion

MP1-153

The Classification Problem

- ◆From a data set describing objects by vectors of *features* and a *class*

Age Sex ChestPain RestBP Cholesterol BloodSugar ECG MaxHeartRt Angina OldPeak Heart Disease

$Vector_1$= <49, 0, 2, 134, 271, 0, 0, 162, 0, 0, 2, 0, 3> Presence
$Vector_2$= <42, 1, 3, 130, 180, 0, 0, 150, 0, 0, 1, 0, 3> Presence
$Vector_3$= <39, 0, 3, 94, 199, 0, 0, 179, 0, 0, 1, 0, 3 > Presence
$Vector_4$= <41, 1, 2, 135, 203, 0, 0, 132, 0, 0, 2, 0, 6 > Absence
$Vector_5$= <56, 1, 3, 130, 256, 1, 2, 142, 1, 0.6, 2, 1, 6 > Absence
$Vector_6$= <70, 1, 2, 156, 245, 0, 2, 143, 0, 0, 1, 0, 3 > Presence
$Vector_7$= <56, 1, 4, 132, 184, 0, 2, 105, 1, 2.1, 2, 1, 6 > Absence

- ◆Find a function *F*: *features* $\rightarrow$ *class* to <u>*classify*</u> a new object

MP1-154

Examples

- Predicting heart disease
  - Features: cholesterol, chest pain, angina, age, etc.
  - Class: {present, absent}
- Finding lemons in cars
  - Features: make, brand, miles per gallon, acceleration,etc.
  - Class: {normal, lemon}
- Digit recognition
  - Features: matrix of pixel descriptors
  - Class: {1, 2, 3, 4, 5, 6, 7, 8, 9, 0}
- Speech recognition
  - Features: Signal characteristics, language model
  - Class: {pause/hesitation, retraction}

Approaches

- Memory based
  - Define a distance between samples
  - Nearest neighbor, support vector machines

- Decision surface
  - Find best partition of the space
  - CART, decision trees

- Generative models
  - Induce a model and impose a decision rule
  - Bayesian networks

Generative Models

- Bayesian classifiers
  - Induce a probability describing the data $P(F_1,\ldots,F_n,C)$
  - Impose a decision rule. Given a new object < $f_1,\ldots,f_n$ > $c = argmax_C\, P(C = c \mid f_1,\ldots,f_n)$
- We have shifted the problem to learning $P(F_1,\ldots,F_n,C)$
- Learn a Bayesian network representation for $P(F_1,\ldots,F_n,C)$

MP1-157

Optimality of the decision rule Minimizing the error rate...

- Let c_i be the ***true*** class, and let l_j be the class returned by the classifier.

  A decision by the classifier is correct if $c_i=l_j$, and in error if $c_i \neq l_j$.
- The error incurred by choose label l_j is

$$E(c_i \mid L) = \sum_{j=1}^{n} \lambda(c_i \mid l_i) P(l_j \mid \vec{f}) = 1 - P(l_i \mid \vec{f})$$

- Thus, had we had access to *P,* we minimize error rate by choosing l_i when

$$P(l_i \mid \vec{f}) > P(l_j \mid \vec{f}) \forall j \neq i$$

which is the decision rule for the Bayesian classifier

MP1-158

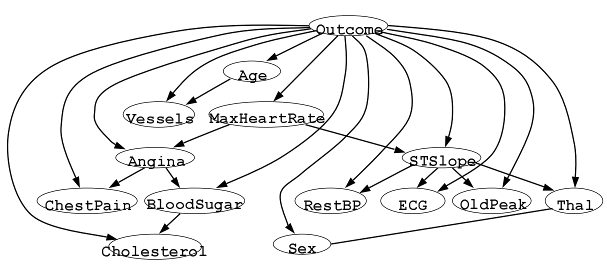
Advantages of the Generative Model Approach

- Output: Rank over the outcomes---likelihood of present vs. absent
- Explanation: What is the profile of a “typical” person with a heart disease
- Missing values: both in training and testing
- Value of information: If the person has high cholesterol and blood sugar, which other test should be conducted?
- Validation: confidence measures over the model and its parameters
- Background knowledge: priors and structure

Advantages of Using a Bayesian Network

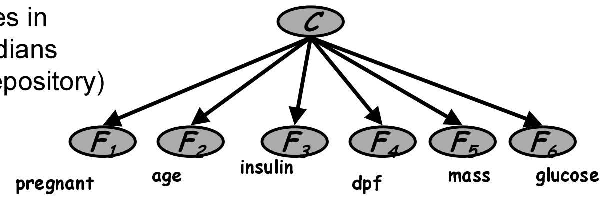
- Efficiency in learning and query answering
  - Combine knowledge engineering and statistical induction
  - Algorithms for decision making, value of information, diagnosis and repair

Heart disease
Accuracy = 85%
Data source
UCI repository



The Naïve Bayesian Classifier

Diabetes in
Pima Indians
(from UCI repository)



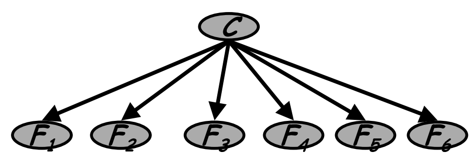
- Fixed structure encoding the assumption that features are independent of each other given the class.

$$P(C \mid F_1, \ldots, F_6) \propto P(F_1 \mid C) \bullet P(F_2 \mid C) \bullet \cdots \bullet P(F_6 \mid C) \bullet P(C)$$

- Learning amounts to estimating the parameters for each $P(F_i|C)$ for each F_i.

MP1-161

The Naïve Bayesian Classifier (cont.)



- Common practice is to estimate

$$\hat{\theta}_{a_i|c} = \frac{N(a_i, c)}{N(c)}$$

- These estimate are identical to MLE for multinomials
- Estimates are robust consisting of low order statistics requiring few instances
- Has proven to be a powerful classifier

MP1-162

Improving Naïve Bayes

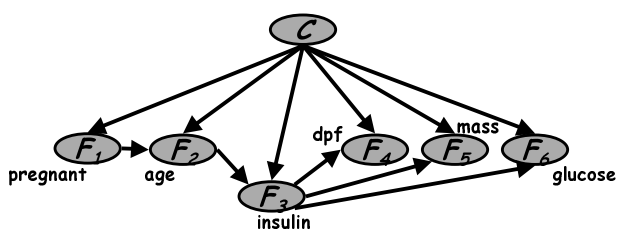
- Naïve Bayes encodes assumptions of independence that may be unreasonable:

  Are *pregnancy* and *age* independent given *diabetes*?

  Problem: same evidence may be incorporated multiple times
- The success of naïve Bayes is attributed to
  - Robust estimation
  - Decision may be correct even if probabilities are inaccurate
- **Idea**: improve on naïve Bayes by weakening the independence assumptions

  Bayesian networks provide the appropriate mathematical language for this task

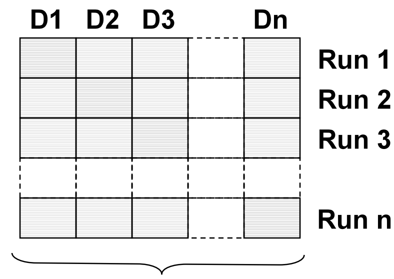
Tree Augmented Naïve Bayes (TAN)



$$P(C \mid F_1,\ldots,F_6) \propto P(F_1 \mid C) \bullet P(F_2 \mid F_1,C) \bullet \cdots \bullet P(F_6 \mid F_3,C) \bullet P(C)$$

- Approximate the dependence among features with a tree Bayes net
- Tree induction algorithm
  - Optimality: maximum likelihood tree
  - Efficiency: polynomial algorithm
- Robust parameter estimation

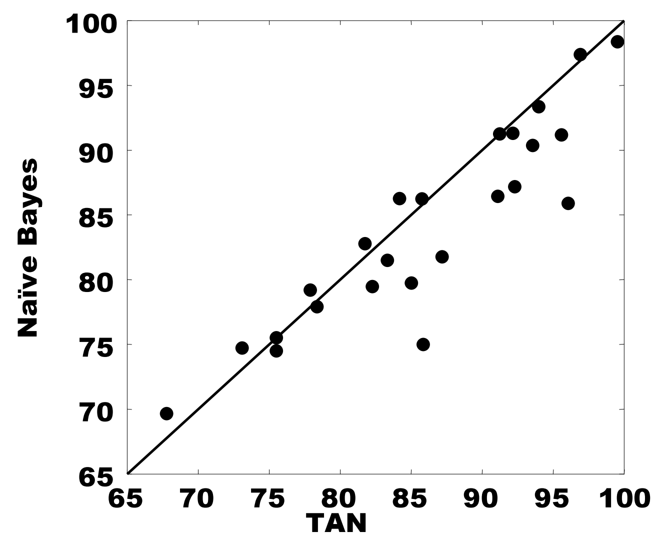
Evaluating the performance of a classifier: n-fold cross validation



Original data set

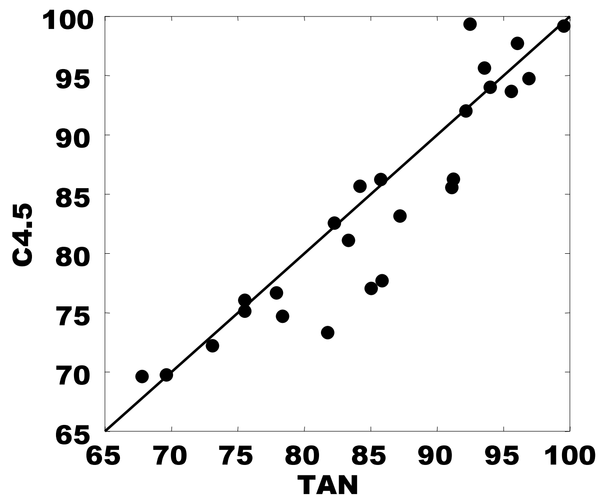
- Partition the data set in *n* segments
- Do *n* times
  - Train the classifier with the green segments
  - Test accuracy on the red segments
- Compute statistics on the *n* runs
  - Variance
  - Mean accuracy
- Accuracy: on test data of size *m*
  - Acc = $\frac{\sum_{k=1}^{m} \lambda_k (c_i \mid l_j)}{m}$

Performance: TAN vs. Naïve Bayes



- 25 Data sets from UCI repository
  - Medical
  - Signal processing
  - Financial
  - Games
- Accuracy based on 5-fold cross-validation
- No parameter tuning

Performance: TAN vs C4.5



- 25 Data sets from UCI repository
  - Medical
  - Signal processing
  - Financial
  - Games
- Accuracy based on 5-fold cross-validation
- No parameter tuning

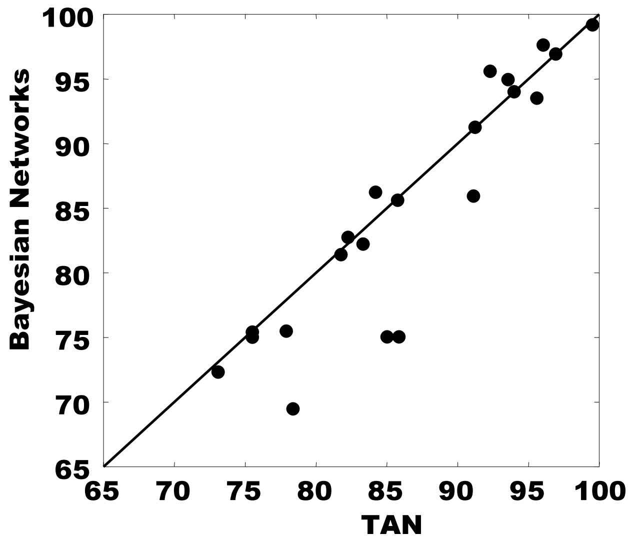
MP1-167

Beyond TAN

- Can we do better by learning a more flexible structure?
- Experiment: learn a Bayesian network without restrictions on the structure

MP1-168

Performance: TAN vs. Bayesian Networks



- 25 Data sets from UCI repository
  - Medical
  - Signal processing
  - Financial
  - Games
- Accuracy based on 5-fold cross-validation
- No parameter tuning

MP1-169

What is the problem?

- Objective function
  - Learning of arbitrary Bayesian networks optimizes $P(C, F_1,\ldots,F_n)$
  - It may learn a network that does a **great** job on $P(F_i,\ldots,F_j)$ but a **poor** job on $P(C \mid F_1,\ldots,F_n)$
    (Given *enough* data… No problem…)
  - We want to optimize classification accuracy or at least the *conditional likelihood* $P(C \mid F_1,\ldots,F_n)$
    - Scores based on this likelihood do not decompose
      $\Rightarrow$ learning is computationally expensive!
    - Controversy as to the correct form for these scores
- Naive Bayes, Tan, etc circumvent the problem by forcing a structure where all features are connected to the *class*

MP1-170

Classification: Summary

- Bayesian networks provide a useful language to improve Bayesian classifiers
  - Lesson: we need to be aware of the task at hand, the amount of training data vs dimensionality of the problem, etc
- Additional benefits
  - Missing values
  - Compute the tradeoffs involved in finding out feature values
  - Compute misclassification costs
- Recent progress:
  - Combine generative probabilistic models, such as Bayesian networks, with decision surface approaches such as Support Vector Machines

MP1-171

Outline

- Introduction
- Bayesian networks: a review
- Parameter learning: Complete data
- Parameter learning: Incomplete data
- Structure learning: Complete data
- Application: classification
- »Learning causal relationships
  - Causality and Bayesian networks
  - Constraint-based approach
  - Bayesian approach
- Structure learning: Incomplete data
- Conclusion

MP1-172

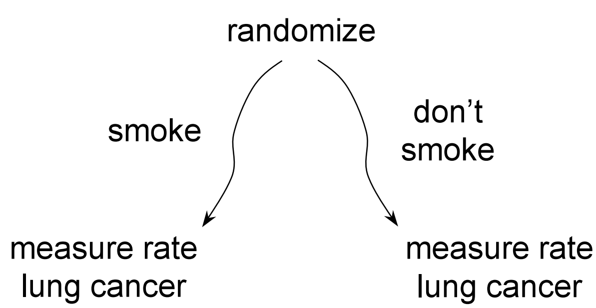
Learning Causal Relations

(Thanks to David Heckerman and Peter Spirtes for the slides)

- Does smoking cause cancer?
- Does ingestion of lead paint decrease IQ?
- Do school vouchers improve education?
- Do Microsoft business practices harm customers?

MP1-173

Causal Discovery by Experiment



Can we discover causality from observational data alone?

MP1-174

What is “Cause” Anyway?

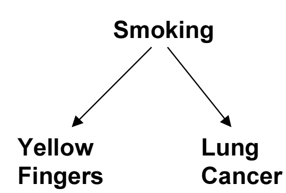
Probabilistic question:
What is p(lung cancer | yellow fingers) ?

Causal question:
What is p(lung cancer | set(yellow fingers)) ?

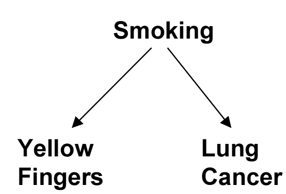
MP1-175

Probabilistic vs. Causal Models

Probabilistic question:
What is p(lung cancer | yellow fingers) ?

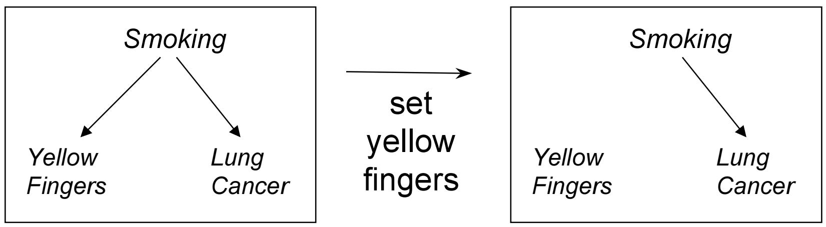


Causal question:
What is p(lung cancer | set(yellow fingers)) ?



MP1-176

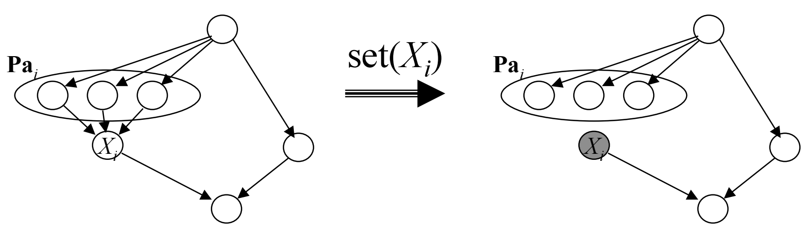
To Predict the Effects of Actions: Modify the Causal Graph



p(lung cancer | set(yellow fingers)) = p(lung cancer)

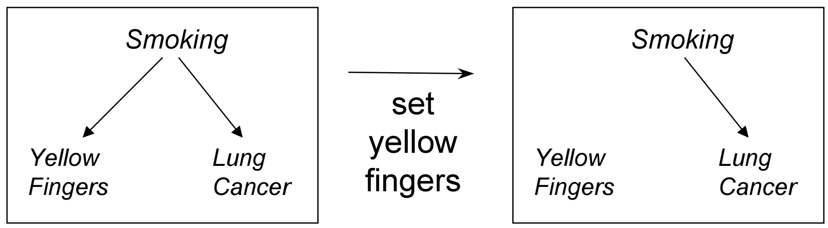
MP1-177

Causal Model



MP1-178

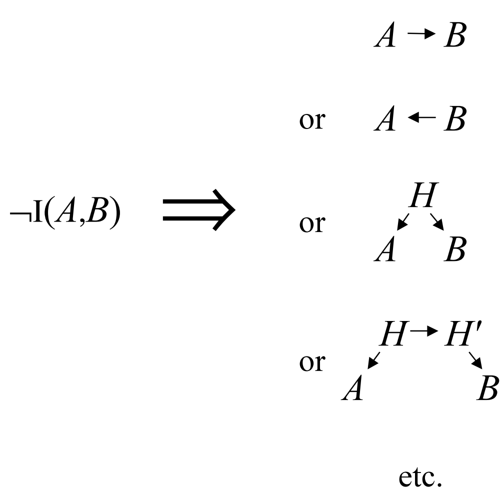
Ideal Interventions



- Pearl: ideal intervention is primitive, defines cause
- Spirtes et al.: cause is primitive, defines ideal intervention
- Heckerman and Shachter: from decision theory one could define both

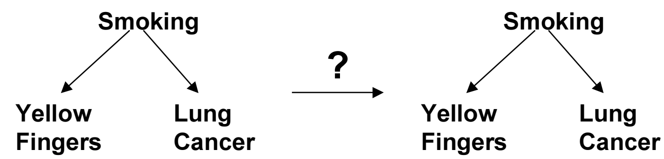
MP1-179

How Can We Learn Cause and Effect from Observational Data?



MP1-180

Learning Cause from Observations: Constraint-Based Approach



(cond independencies from data)

(causal assertions)

Bridging assumptions:

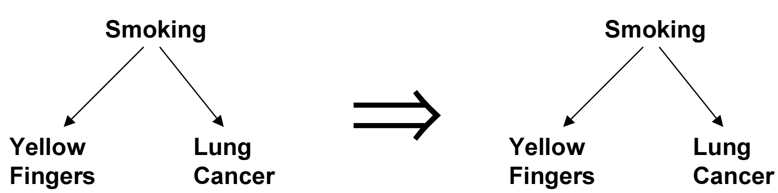
◆Causal Markov assumption

◆Faithfulness

MP1-181

Causal Markov assumption

We can interpret the causal graph as a probabilistic one



i.e.: absence of cause ⇒ conditional independence

Faithfulness

There are no accidental independencies

E.g., cannot have:

Smoking → **Lung Cancer** and **I(smoking,lung cancer)**

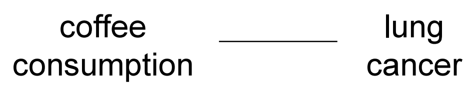
i.e.: conditional independence $\Rightarrow$ absence of cause

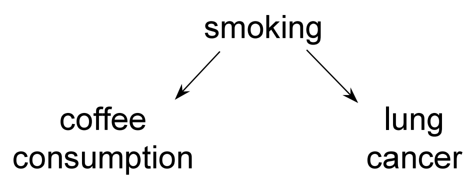
Other assumptions

- All models under consideration are causal
- All models are acyclic

All models under consideration are causal

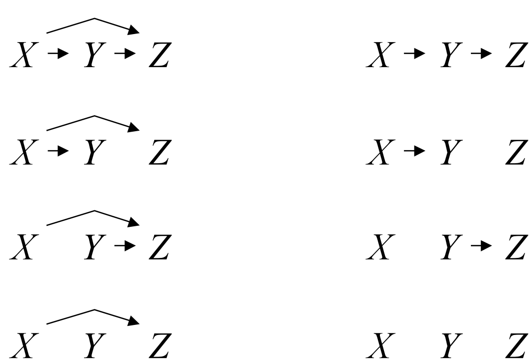
No unexplained correlations





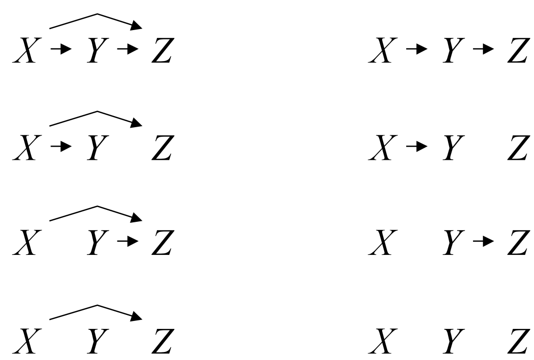
MP1-185

Learning Cause from Observations: Constraint-based method



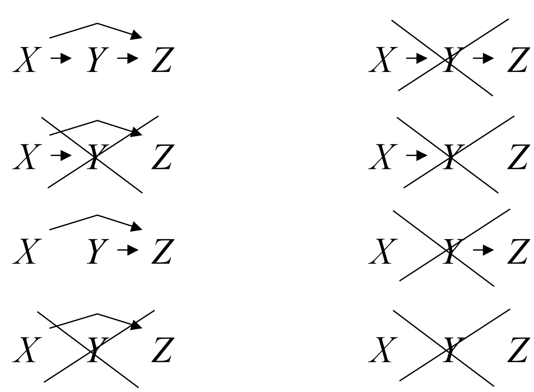
Assumption: These are all the possible models

Learning Cause from Observations: Constraint-based method



Data: The only independence is I(X,Y)

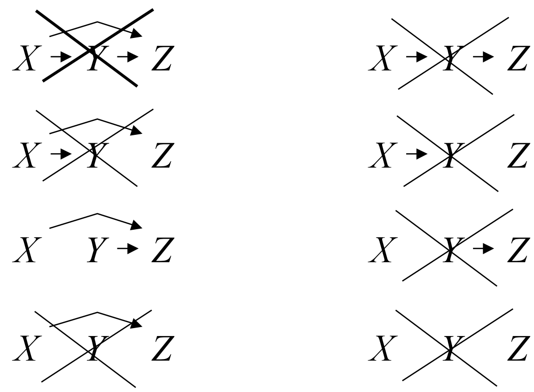
Learning Cause from Observations: Constraint-based method



only I(X,Y)

CMA: Absence of cause ⇒ conditional independence

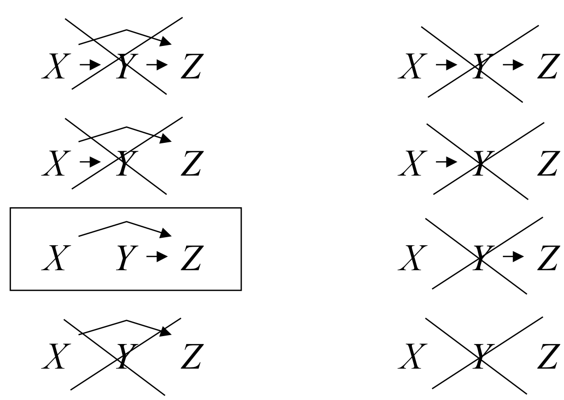
Learning Cause from Observations: Constraint-based method



I(X,Y)

Faithfulness: Conditional independence ⇒ absence of cause

Learning Cause from Observations: Constraint-Based Method

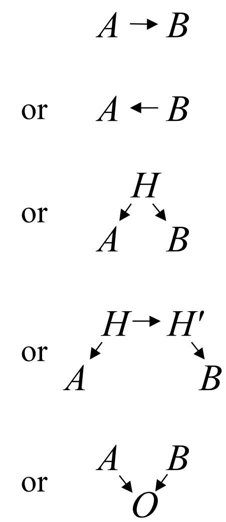


only I(X,Z)

Conclusion: X and *Y* are causes of *Z*

Cannot Always Learn Cause

$\neg I(A,B) \Rightarrow$



hidden common causes

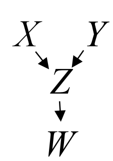
selection bias

etc.

MP1-191

But with four (or more) variables…

Suppose we observe the independencies & dependencies consistent with



that is...

$I(X,Y)$

$I(X \& Y, W|Z)$

$\neg I(X,Y|Z)$

etc.

Then, in <u>every</u> acyclic causal structure not excluded by CMA and faithfulness, there is a directed path from Z to W.

Z causes *W*

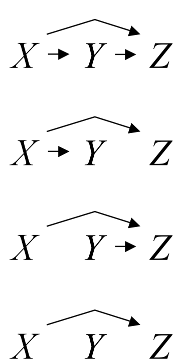
MP1-192

Constraint-Based Approach

- Algorithm based on the systematic application of
  - Independence tests
  - Discovery of “Y” and “V” structures
- Difficulties:
  - Need infinite data to learn independence with certainty
    - What significance level for independence tests should we use?
    - Learned structures are susceptible to errors in independence tests

MP1-193

The Bayesian Approach



$p(G_1) = 0.25$ $\quad p(G_1 \mid \mathbf{d}) = 0.01$

$p(G_2) = 0.25$ $\quad p(G_2 \mid \mathbf{d}) = 0.1$

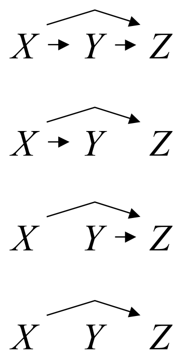
Data $\mathbf{d}$ ⟹

$p(G_3) = 0.25$ $\quad p(G_3 \mid \mathbf{d}) = 0.8$

$p(G_4) = 0.25$ $\quad p(G_4 \mid \mathbf{d}) = 0.09$

One conclusion: $p(X \text{ and } Y \text{ cause } Z|\mathbf{d})=0.01+0.8=0.81$

The Bayesian approach



$p(G_1) = 0.25$ — $p(G_1 \mid \mathbf{d}) = 0.01$

$p(G_2) = 0.25$ — $p(G \mid \mathbf{d}) = 0.1$

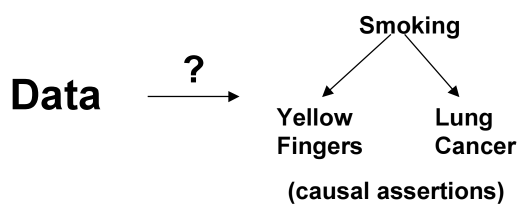
Data $\mathbf{d}$ ⟹

$p(G_3) = 0.25$ — $p(G_3 \mid \mathbf{d}) = 0.8$

$p(G_4) = 0.25$ — $p(G_4 \mid \mathbf{d}) = 0.09$

$$p(Z \mid set(Y), \mathbf{d}) = \sum_G p(Z \mid set(Y), \mathbf{d}, G)\, p(G / \mathbf{d})$$

Assumptions



- Causal Markov assumption
- Faithfulness
- All models under consideration are causal
- etc.

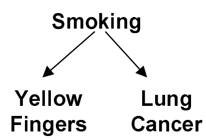
MP1-196

Definition of Model Hypothesis G

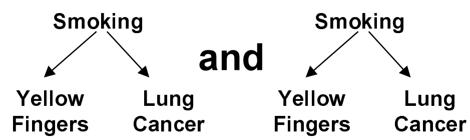
The hypothesis corresponding to DAG model G:

- m is a causal model
- (+CMA) the true distribution has the independencies implied by m

DAG model G:



Hypothesis G:



MP1-197

Faithfulness

$p(\theta \mid G)$ is a probability density function for every G

$$\Downarrow$$

the probability that faithfulness is violated = 0

Example: DAG model G: X->Y

$$p(X \perp Y \mid G) = 0$$

MP1-198

Causes of publishing productivity

Rodgers and Maranto 1989

ABILITY	**Measure of ability (undergraduate)**
GPQ	**Graduate program quality**
PREPROD	**Measure of productivity**
QFJ	**Quality of first job**
SEX	**Sex**
CITES	**Citation rate**
PUBS	**Publication rate**

Data: 86 men, 76 women

MP1-199

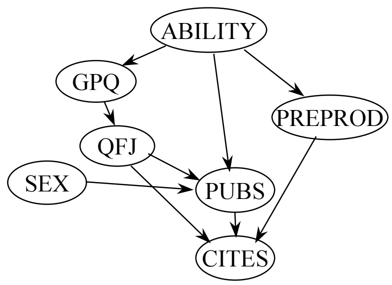
Causes of publishing productivity

Assumptions:

- No hidden variables
- Time ordering:
- Otherwise uniform distribution on structure
- Node likelihood: linear regression on parents

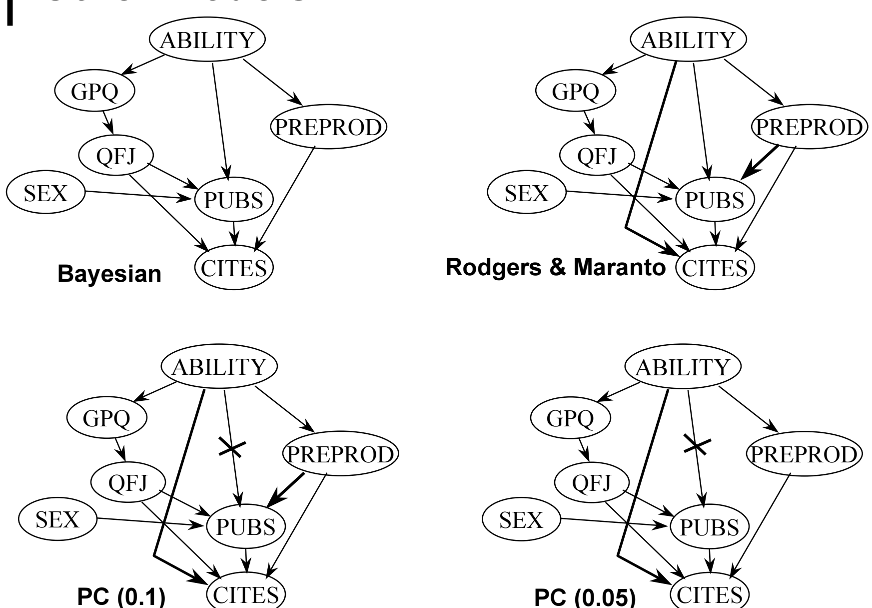
MP1-200

Results of Greedy Search...



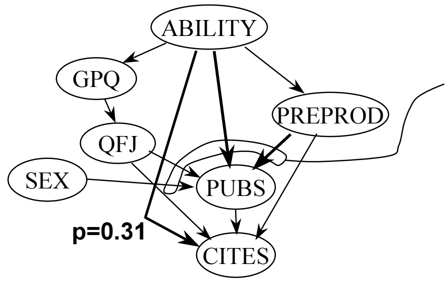
MP1-201

Other Models



MP1-202

Bayesian Model Averaging



parents of PUBS	prob
SEX, QFJ, ABILITY, PREPROD	0.09
SEX, QFJ, ABILITY	0.37
SEX, QFJ, PREPROD	0.24
SEX, QFJ	0.30

MP1-203

Challenges for the Bayesian Approach

- Efficient selective model averaging / model selection
- Hidden variables and selection bias
  - Prior assessment
  - Computation of the score (posterior of model)
  - Structure search
- Extend simple discrete and linear models

MP1-204

Benefits of the Two Approaches

Bayesian approach:
- ◆Encode uncertainty in answers
- ◆Not susceptible to errors in independence tests

Constraint-based approach:
- ◆More efficient search
- ◆Identify possible hidden variables

MP1-205

Summary

- ◆The concepts of
  - ● Ideal manipulation
  - ● Causal Markov and faithfulness assumptions

  enable us to use Bayesian networks as causal graphs for causal reasoning and causal discovery
- ◆Under certain conditions and assumptions, we can discover causal relationships from observational data
- ◆The constraint-based and Bayesian approaches have different strengths and weaknesses

MP1-206

Outline

	Known Structure	Unknown Structure
Complete data		
Incomplete data		

- ◆Introduction
- ◆Bayesian networks: a review
- ◆Parameter learning: Complete data
- ◆Parameter learning: Incomplete data
- ◆Structure learning: Complete data
- ◆Application: classification
- ◆Learning causal relationships
- »Structure learning: Incomplete data
- ◆Conclusion

Learning Structure for Incomplete Data

Distinguish:

- ◆Learning structure for given set of random variables
  - Hard search problem
- ◆Introducing new hidden variables
  - How to recognize the need for a new hidden variable?
  - Where to introduce the hidden variable in current structure?
  - Open ended...

Incomplete Data : Structure Scores

MDL

$$MDL(G:D) = l(G:D) - \frac{\log M}{2}\dim(G) - DL(G)$$

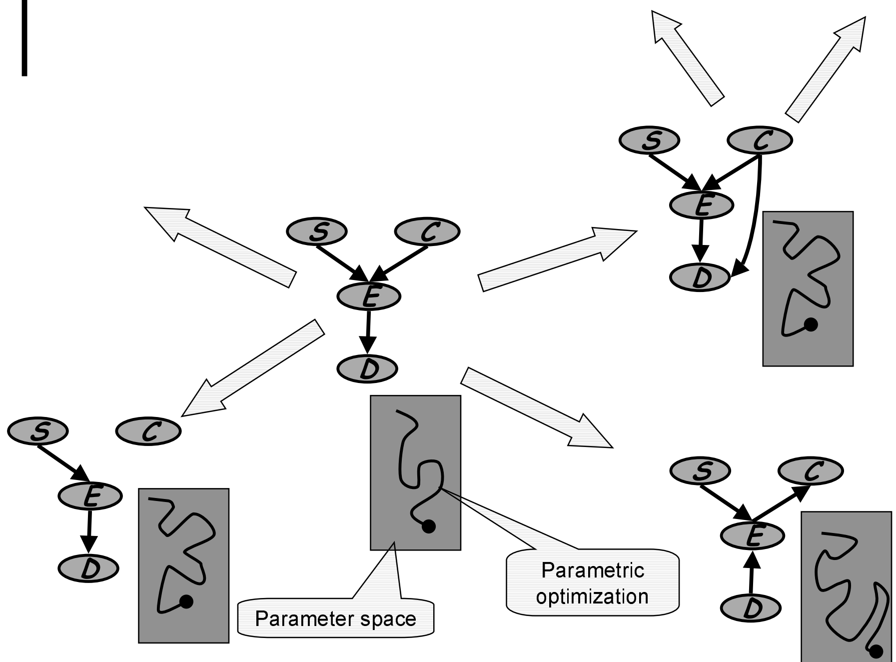
- Use same MDL formula with probability of the data
- Requires finding maximum likelihood parameters
  - Using methods for parameter learning (e.g., EM)
- Theoretical results show that penalty should be adjusted

Incomplete Data : Structure Scores (cont.)

Bayesian:

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

- We cannot evaluate the marginal likelihood
- We have to resort to approximations:
  - Asymptotic approximations
    - Evaluate score around MAP parameters
    - Need to find MAP parameters (e.g., EM)
  - Stochastic approximations
    - Apply stochastic integration methods
    - Much slower



Problem

Such procedures are computationally expensive!

- Computation of optimal parameters, per candidate, requires non-trivial optimization step
- Spend non-negligible computation on a candidate, even if it is a low scoring one

In practice, such learning procedures are feasible only when we consider small sets of candidate structures

Structural EM

- **Idea:** Use parameters found for previous structures to help evaluate new structures.
- **Scope**: searching over structures over the same set of random variables.

Outline:

- Perform search in (Structure, Parameters) space.
- Use EM-like iterations, using previously best found solution as a basis for finding either:
  - Better scoring parameters --- “parametric” EM step

  or
  - Better scoring structure --- “structural” EM step

Structural EM

- Recall, in complete data we had
  - Decomposition $\Rightarrow$ efficient search

Idea:

- Instead of optimizing the real score…
- Find an alternative score that is amenable to search
- Such that
  - We recover decomposability and sufficient statistics
  - Maximizing new score ➔ improvement in real score

Expected scores

Data:

X	Y	Z
H	?	T
?	?	H
H	?	?
T	?	?
H	?	T
H	?	H
T	?	H

H

O

- Let *O* denote the observed data
- Let *H* denote the hidden variables

- If we have a distribution *Q(H)*, then "complete" data

$$E_Q[Score(M : O, H)] = \sum_H Q(H) Score(M : O, H)$$

- Since *O, H* describe complete data

$$E_Q[Score(M \mid O, H)] = E_Q[\sum_i Score_{X_i|Pa_i^G}(N_{X_i, Pa_i^G})]$$

$$= \sum_i E_Q[Score_{X_i|Pa_i^G}(N_{X_i, Pa_i^G})]$$

- The expected score is decomposable!

How do we choose *Q(H)*?

Theorem: If $Q(H) = P(H|O,M_0)$ then

$$Score(M \mid O) - Score(M_0 \mid O) \geq$$
$$E_Q[Score(M \mid H, O)] - E_Q[Score(M_0 \mid H, O)]$$

Consequences:

- *M* is better than M_0 according to expected score,
  $\Rightarrow$ *M* is also better according to true score

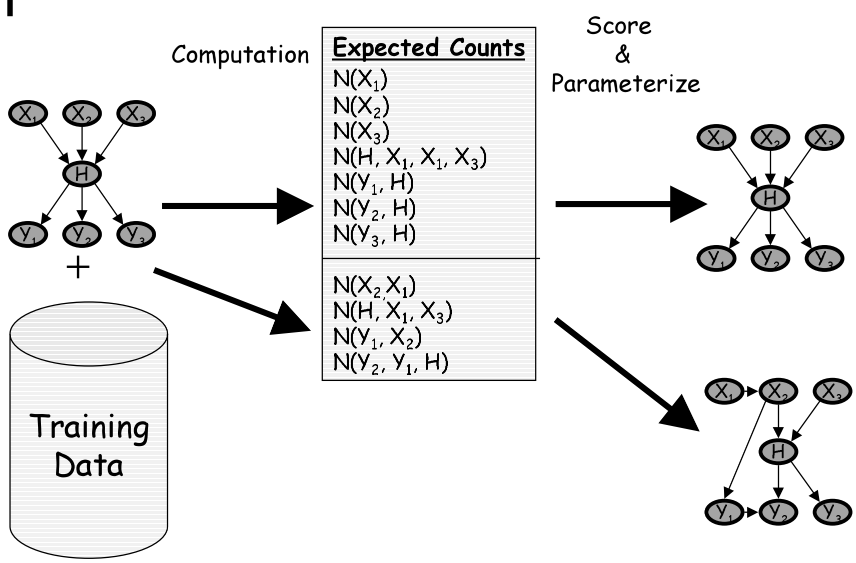
Structural EM for MDL

- For the MDL score, we get that

$$
\begin{aligned}
&E[\mathrm{MDL}(B : D^{+}) \mid D, B_0]\\
&\quad = E[\log P(D^{+} \mid B) \mid D, B_0] - \mathrm{Penalty}(B)\\
&\quad = E[\sum_i N(X_i, Pa_i) \log P(X_i \mid Pa_i) \mid D, B_0] - \mathrm{Penalty}(B)\\
&\quad = \sum_i E[N(X_i, Pa_i) \mid D, B_0] \log P(X_i \mid Pa_i) - \mathrm{Penalty}(B)
\end{aligned}
$$

Consequence:

- We can use complete-data methods, were we use expected counts, instead of actual counts



Structural EM in Practice

In theory:

- **E-Step**: compute expected counts for all candidate structures
- **M-Step**: choose structure that maximizes expected score

Problem: there are (exponentially) many structures

- We cannot computed expected counts for all of them in advance

Solution:

- **M-Step**: search over network structures (e.g., hill-climbing)
- **E-Step**: on-demand, for each structure G examined by M-Step, compute expected counts
- Use smart caching schemes to minimize overall computations

The Structural EM Procedure

Input: $B_0 = (G_0, \Theta_0)$
loop for $n = 0, 1, \ldots$ until convergence
Improve parameters:
$\Theta`_n =$ Parametric-EM (G_n, Θ_n)
let $B`_n = (G_n, \Theta`_n)$
Improve structure:
Search for a network $B_{n+1} = (G_{n+1}, \Theta_{n+1})$ s.t.
$E[Score(B_{n+1}:D) \mid B`_n] > E[Score(B`_n:D) \mid B`_n]$

- Parametric-EM() can be replaced by Gradient Ascent, Newton-Raphson methods, or accelerated EM.
- Early stopping parameter optimization stage avoids “entrenchment” in current structure.

Structural EM: Convergence Properties

Theorem: The SEM procedure converges in score:

The limit $\lim_{n\to\infty} Score(B_n : D)$ exists.

Theorem: Convergence point is a local maxima:

If $G_n = G$ infinitely often, then, Θ_G, the limit of the parameters in the subsequence with structure *G,* is a stationary point in the parameter space of G.

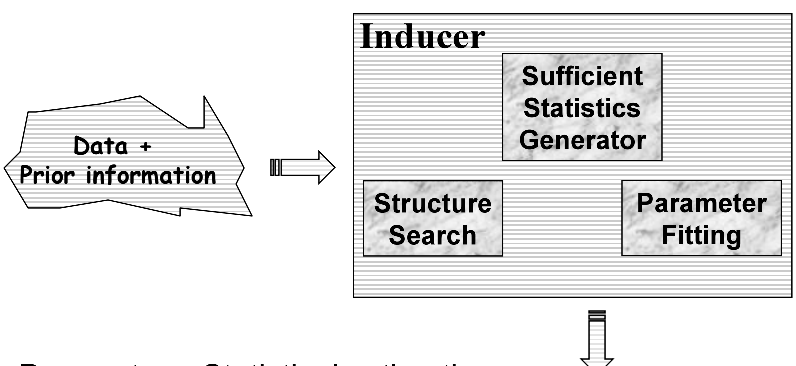
Learning Structure from Incomplete Data: Summary

- Hard problem!
- Initial progress:
  - EM-like search techniques
    - 6 CPU years ⇒ 6 CPU hours
- Problems:
  - Escaping local maxima
  - Inducing new variables

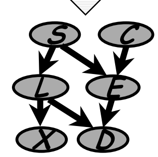
Outline

◆Introduction
◆Bayesian networks: a review
◆Parameter learning: Complete data
◆Parameter learning: Incomplete data
◆Structure learning: Complete data
◆Application: classification
◆Learning causal relationships
◆Structure learning: Incomplete data
»Conclusion

Summary: Learning Bayesian Networks



◆Parameters: Statistical estimation
◆Structure: Search optimization
◆Incomplete data: EM



Untouched issues

- Feature engineering
  - From measurements to features
- Feature selection
  - Discovering the relevant features
- Smoothing and priors
  - Not enough data to compute robust statistics
- Representing selection bias
  - All the subjects from the same population

Untouched Issues (Cont.)

- Unsupervised learning
  - Clustering and exploratory data analysis
- Incorporating time
  - Learning DBNs
- Sampling, approximate inference and learning
  - Non-conjugate families, and time constraints
- On-line learning, relation to other graphical models

Some Applications

- ◆Biostatistics -- Medical Research Council (Bugs)
- ◆Data Analysis -- NASA (AutoClass)
- ◆Collaborative filtering -- Microsoft (MSBN)
- ◆Fraud detection -- ATT
- ◆Classification -- SRI (TAN-BLT)
- ◆Speech recognition -- UC Berkeley

Systems

◆BUGS - Bayesian inference Using Gibbs Sampling
- Assumes fixed structure
- No restrictions on the distribution families
- Relies on Markov Chain Montecarlo Methods for inference
- www.mrc-bsu.com.ac.uk/bugs

◆AutoClass - Unsupervised Bayesian classification
- Assumes a naïve Bayes structure with hidden variable at the root representing the classes
- Extensive library of distribution families.
- ack.arc.nasa.gov/ic/projects/bayes-group/group/autoclass/

Systems (Cont.)

- MSBN - Microsoft Belief Netwoks
  - Learns both parameters and structure, various search methods
  - Restrictions on the family of distributions
  - www.research.microsoft.com/dtas/
- TAN-BLT - Tree Augmented Naïve Bayes for supervised classification
  - Correlations among features restricted to forests of trees
  - Multinomial, Gaussians, mixtures of Gaussians, and linear Gaussians
  - www.erg.sri.com/projects/LAS
- Many more - look into AUAI, Web etc.

Current Topics

- Time
  - Beyond discrete time and beyond fixed rate
- Causality
  - Removing the assumptions
- Hidden variables
  - Where to place them and how many?
- Model evaluation and active learning
  - What parts of the model are suspect and what and how much data is needed?

Perspective: What's Old and What's New

- **Old: Statistics and probability theory**
  - Provide the enabling concepts and tools for parameter estimation and fitting, and for testing the results
- **New: Representation and exploitation of domain structure**
  - Decomposability
    - Enabling scalability and computation-oriented methods
  - Discovery of causal relations and statistical independence
    - Enabling explanation generation and interpretability
  - Prior knowledge
    - Enabling a mixture of knowledge-engineering and induction

The Future...

- Progress will parallel and leverage on extensions to modeling
  - More expressive representation languages
  - Better continuous/discrete models
  - Increase cross-fertilization with neural networks
- Range of applications
  - Biology - DNA, control, financial, perception...
- Beyond current learning model
  - Feature discovery
  - Model decisions about the process: distributions, feature selection
  - Utilities
- Hybrid methods -- Bayesian networks as "glue"?

Many thanks to...

- Gil Bejerano
- Lise Getoor
- David Heckerman
- Daphne Koller
- Uri Lerner
- Ron Parr
- Peter Spirtes
- Bikash Sabata

…..And remember

For current slides, additional material, and reading list see http://www.cs.berkeley.edu/~nir/Tutorial