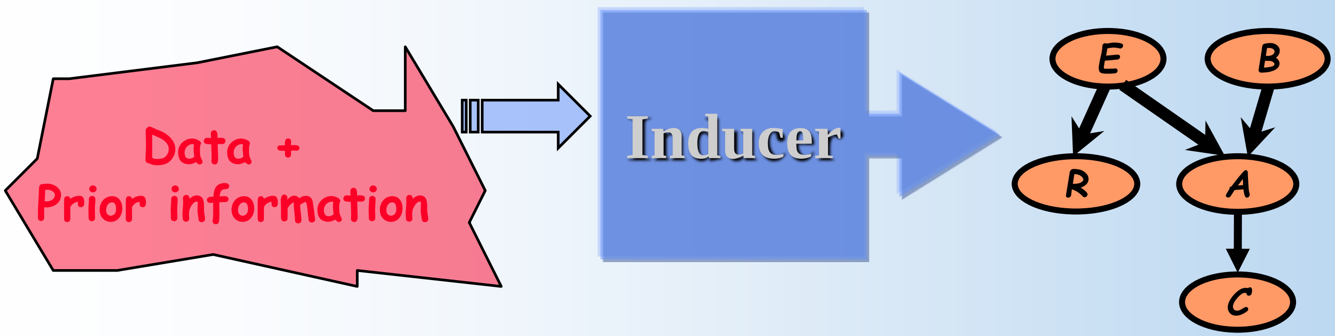
Learning Bayesian Networks

Prof. Dr. Peter Haddawy
Faculty of ICT
Mahidol University

Some material adopted from slides by Nir Friedman, Jack Breese, and Daphne Koller.

Learning Bayesian networks



E	B	P(A | E,B)	
e	b	.9	.1
e	$\bar{b}$	.7	.3
$\bar{e}$	b	.8	.2
$\bar{e}$	$\bar{b}$	.99	.01

The Learning Problem

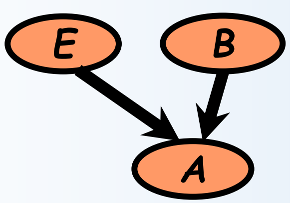
	Known Structure	**Unknown Structure**
Complete Data	Statistical parametric estimation (closed-form eq.)	Discrete optimization over structures (discrete search)
Incomplete Data	Parametric optimization (EM, gradient descent...)	Combined (Structural EM, mixture models…)

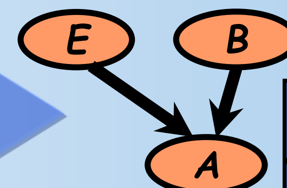
Learning Problem

	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…)

E	B	P(A | E,B)	
e	b	?	?
e	$\bar{b}$	?	?
$\bar{e}$	b	?	?
$\bar{e}$	$\bar{b}$	?	?

E, B, A
<Y,N,N>
<Y,Y,Y>
<N,N,Y>
<N,Y,Y>
.
.
<N,Y,Y>

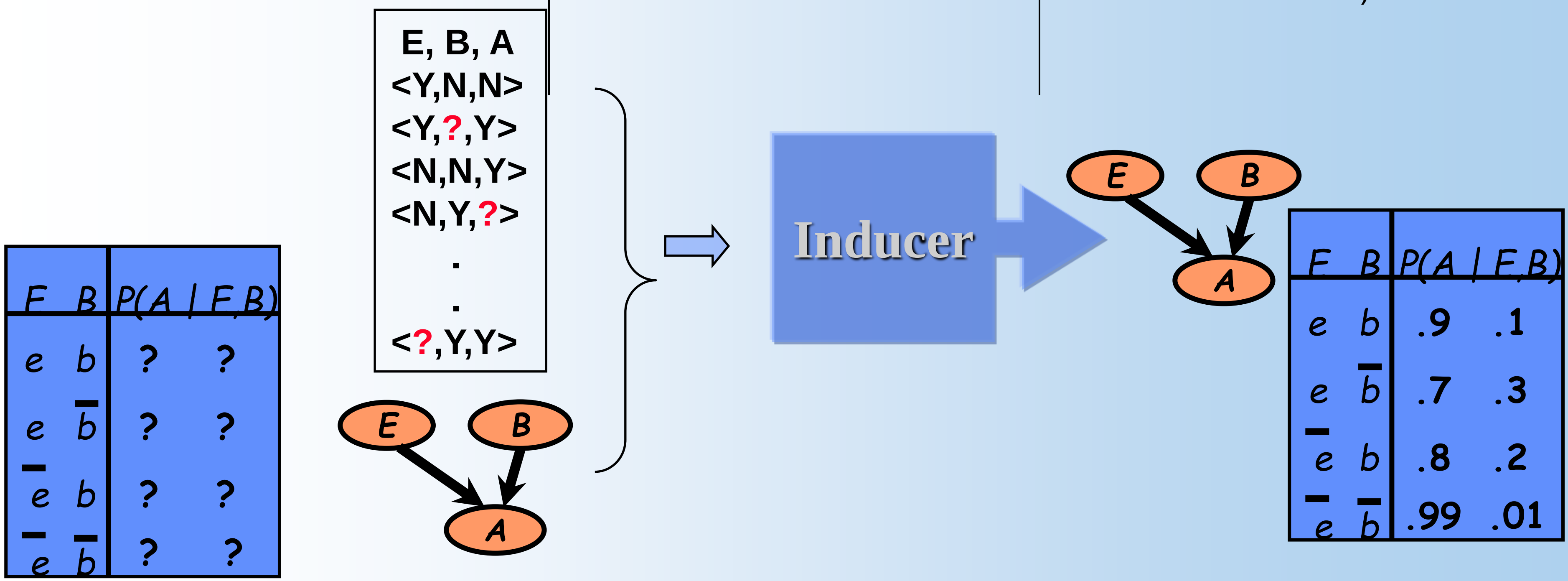




E	B	P(A | E,B)	
e	b	.9	.1
e	$\bar{b}$	.7	.3
$\bar{e}$	b	.8	.2
$\bar{e}$	$\bar{b}$	.99	.01

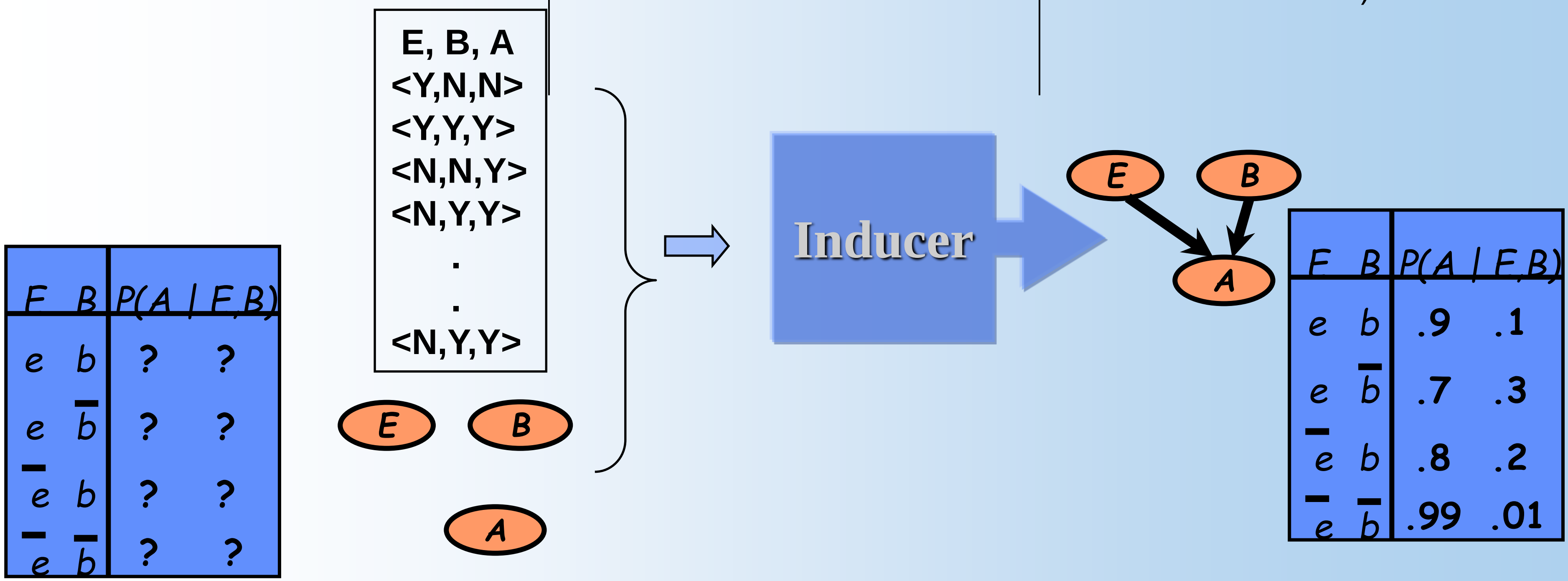
Learning Problem

	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…)



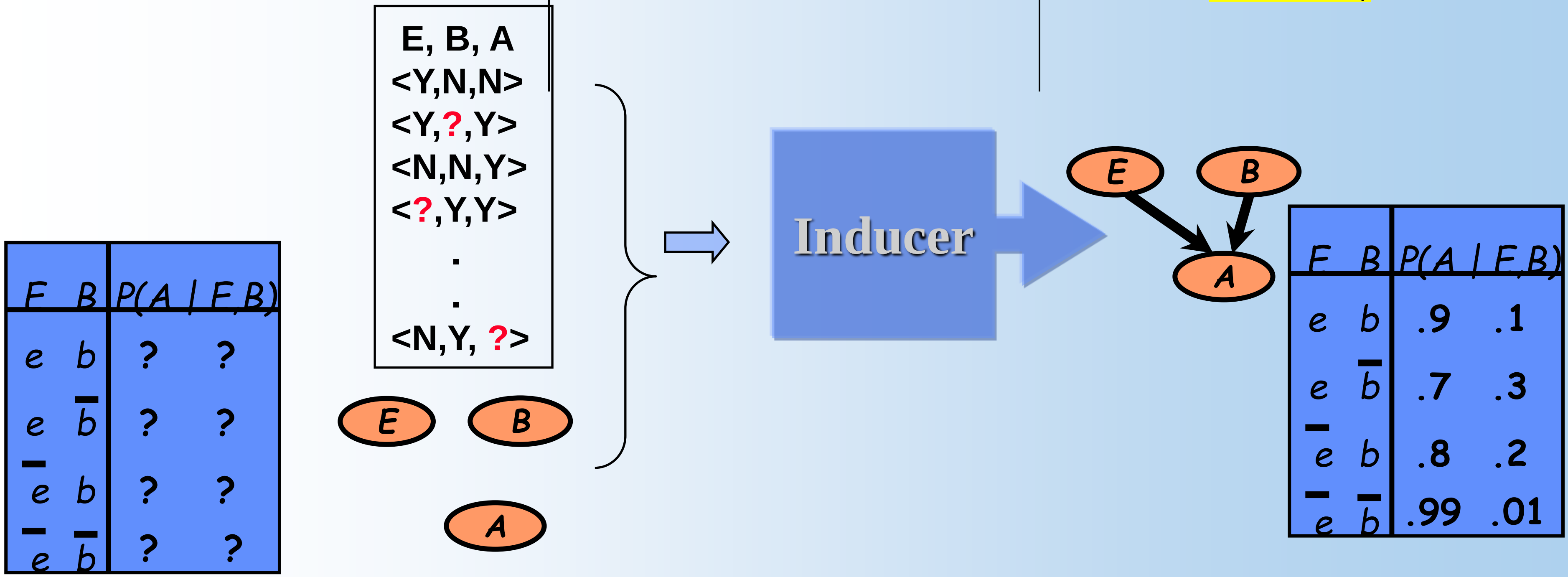
Learning Problem

	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…)

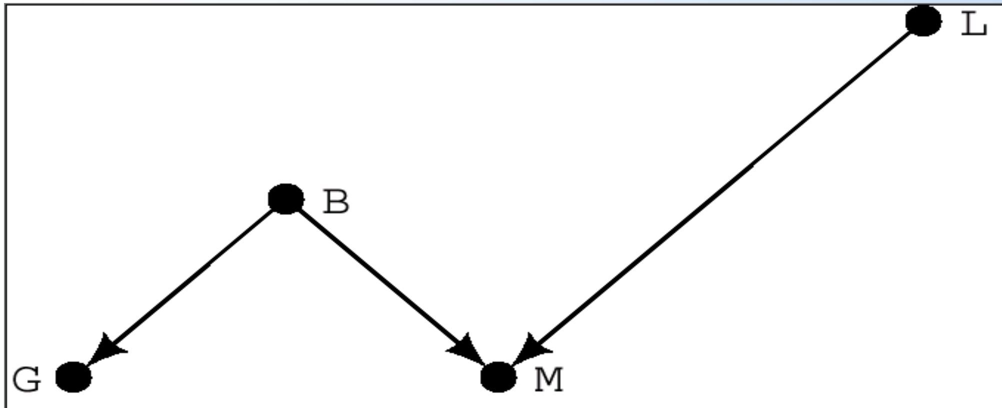


Learning Problem

	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…)



A Network and Training Data



G	M	B	L	Number of instances
True	*True*	*True*	*True*	54
True	*True*	*True*	*False*	1
True	*False*	*True*	*True*	7
True	*False*	*True*	*False*	27
False	*True*	*True*	*True*	3
False	*False*	*True*	*False*	2
False	*False*	*False*	*True*	4
False	*False*	*False*	*False*	2
				100

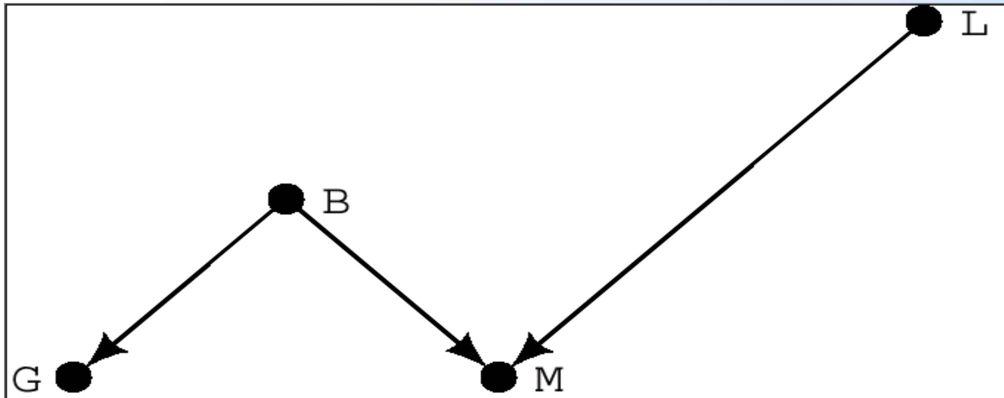
No Missing Data

- If we have an ample number of training samples, we have only to compute sample statistics for each node and its parents.
- CPT for some node *V* given its parents *Pa*(*V*)
  - The sample statistics for *V* and Pa(V):

$$\hat{p}(V = v_i \mid \mathrm{Pa} = \mathrm{p}_j) = \frac{n(V = v_i \wedge Pa = p_j)}{n(Pa = p_j)}$$

  - Given by the number of samples in D having $V = v_i$ and Pa(V)=p_j divided by the number of samples having Pa(V)=p_j

An Example for No Missing Data



G	M	B	L	Number of instances
True	*True*	*True*	*True*	54
True	*True*	*True*	*False*	1
True	*False*	*True*	*True*	7
True	*False*	*True*	*False*	27
False	*True*	*True*	*True*	3
False	*False*	*True*	*False*	2
False	*False*	*False*	*True*	4
False	*False*	*False*	*False*	2
				100

$$\hat{p}(B = True) = 0.94$$

$$\hat{p}(L = True) = 0.68$$

$$\hat{p}(M = True \mid B = True, L = False)$$

$$= \frac{1}{30} = 0.03$$

Laplace Smoothing

It is often useful to be able to combine expert opinion with data, particularly when data is scarce. This can be done if we can assign a virtual sample size to the expert's opinion.

$$\hat{P}(A_j = a_{jk} \mid C = c_i) = \frac{n_c + mp}{n + m}$$

n_c : number of training examples for which $A_j = a_{jk}$ and $\mathrm{C} = c_i$

n : number of training examples for which $C = c_i$

p : prior estimate (usually, $p = 1/t$ for t possible values of A_j)

m : weight to prior (number of "virtual" examples, $m \geq 1$)

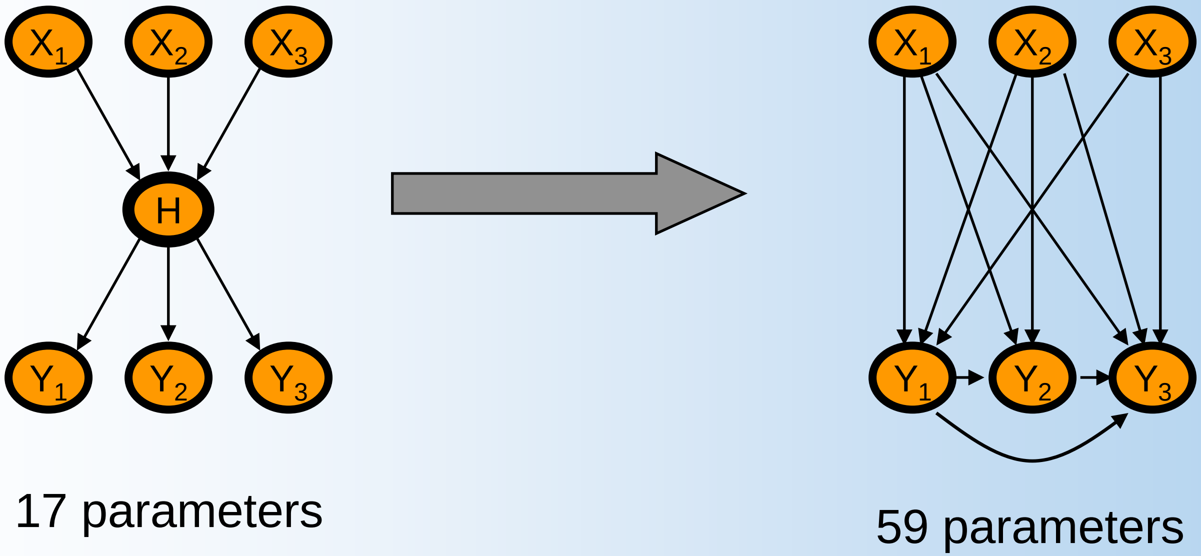
Incomplete Data

- Data is often **incomplete**
  - Some variables of interest are not assigned value

- This phenomena may happen when we have
  - Missing values
  - Hidden variables

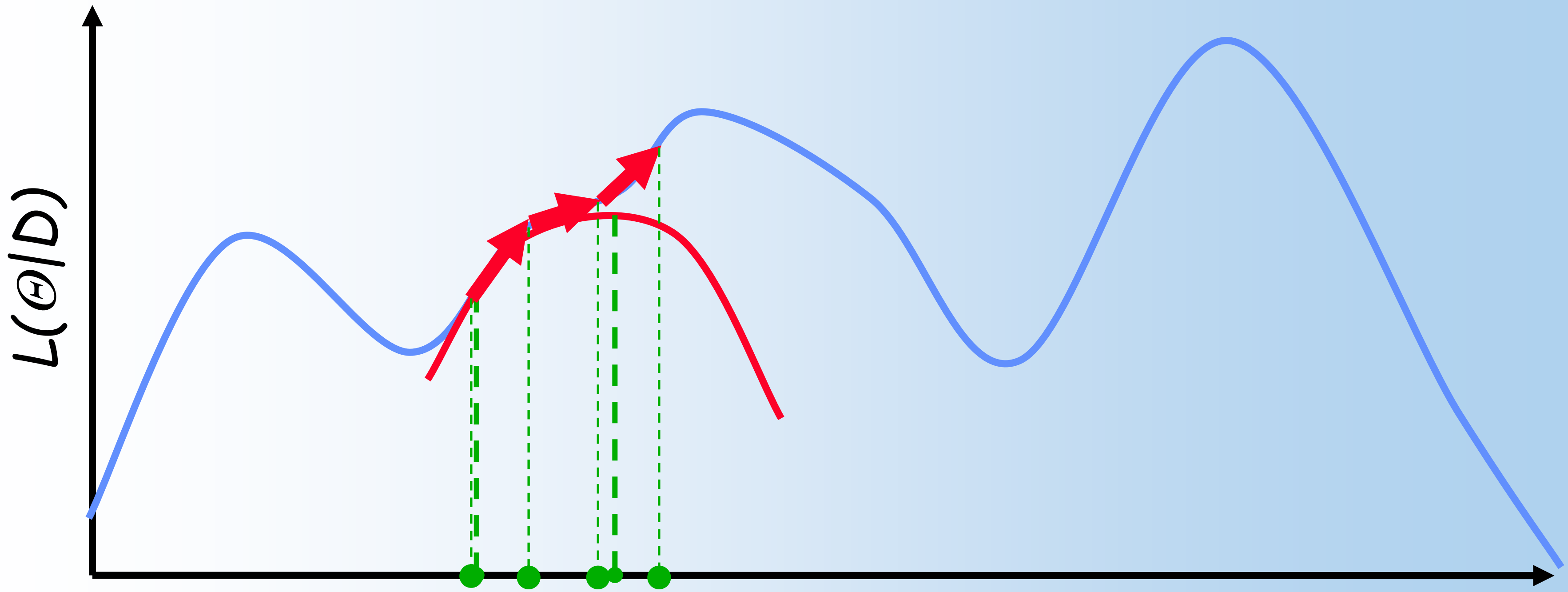
Hidden (Latent) Variables

- Attempt to learn a model with variables we never observe
- Why should we care about unobserved variables?
  - Limited data
  - Overfitting



MLE from Incomplete Data

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



Gradient Ascent:

Follow gradient of likelihood w.r.t. to parameters

Find local maxima

Require multiple restarts to find approx. to the global maximum

Expectation Maximization (EM):

Use "current point" to construct alternative function (which is "nice")

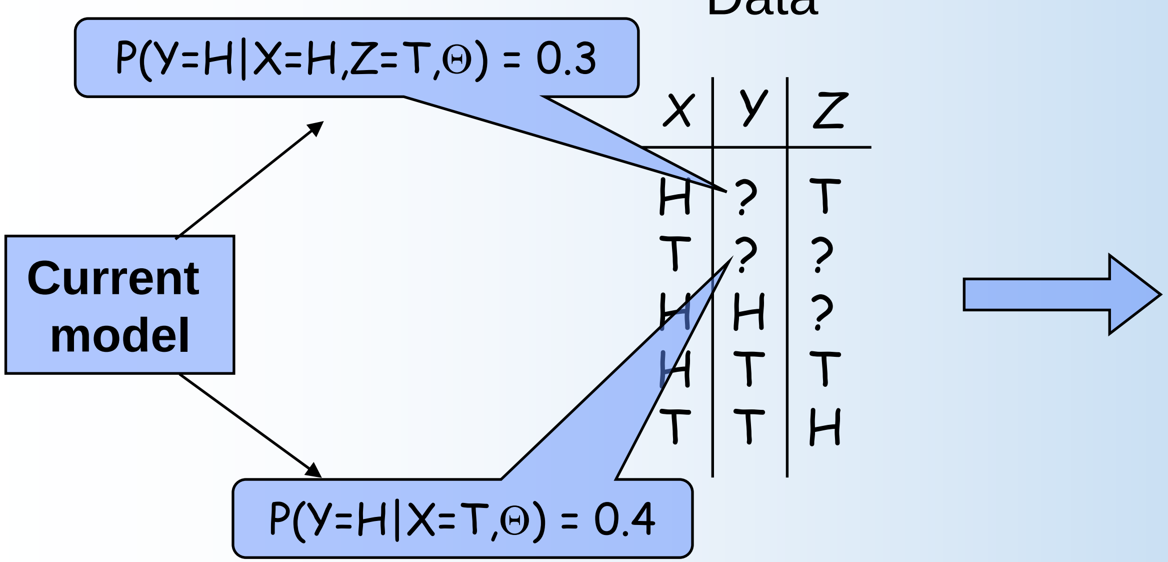
Guaranty: maximum of new function is better scoring than current point

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



Data

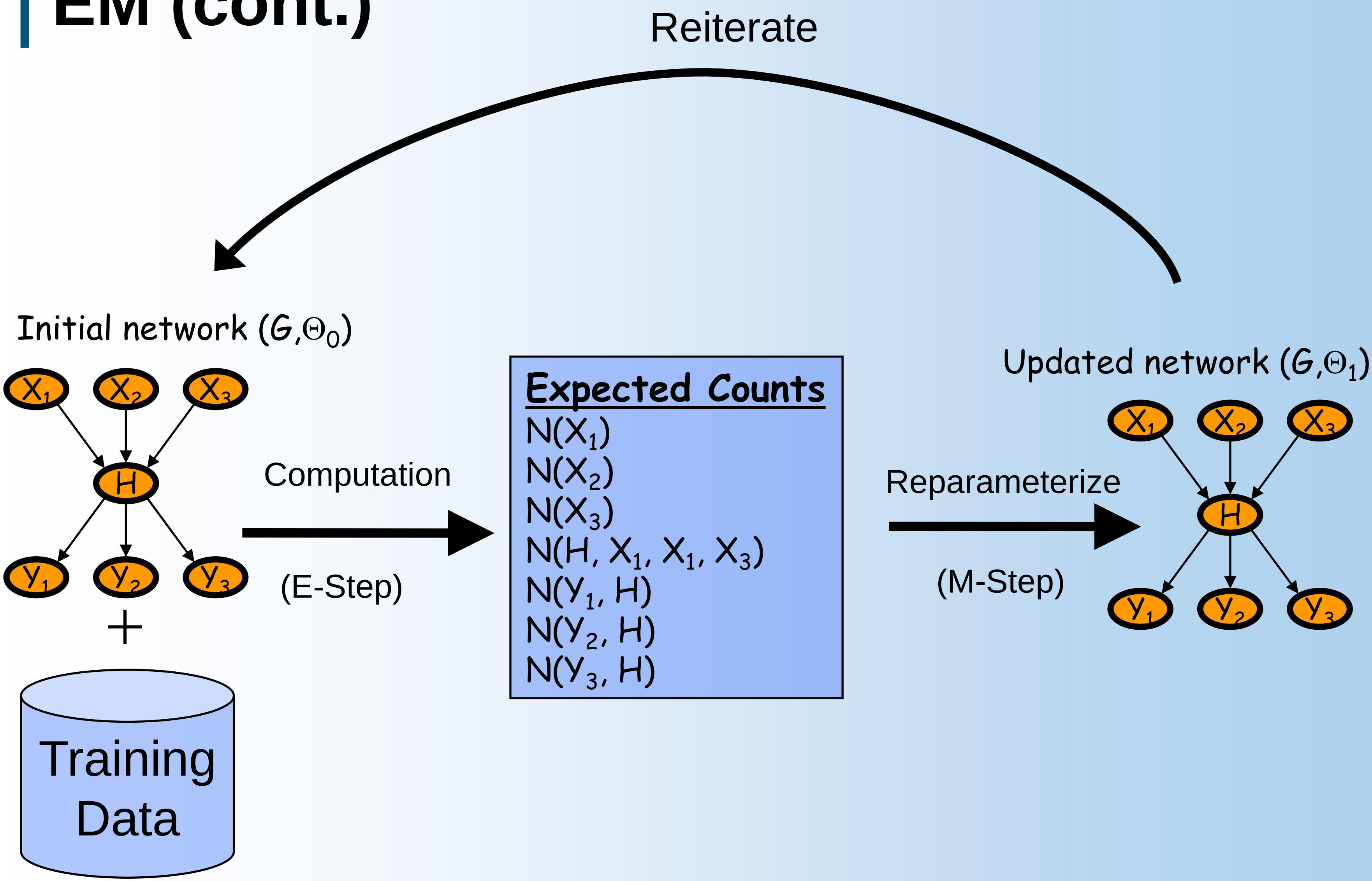
X	Y	Z
H	?	T
T	?	?
H	H	?
H	T	T
T	T	H

Expected Counts

N (X,Y)

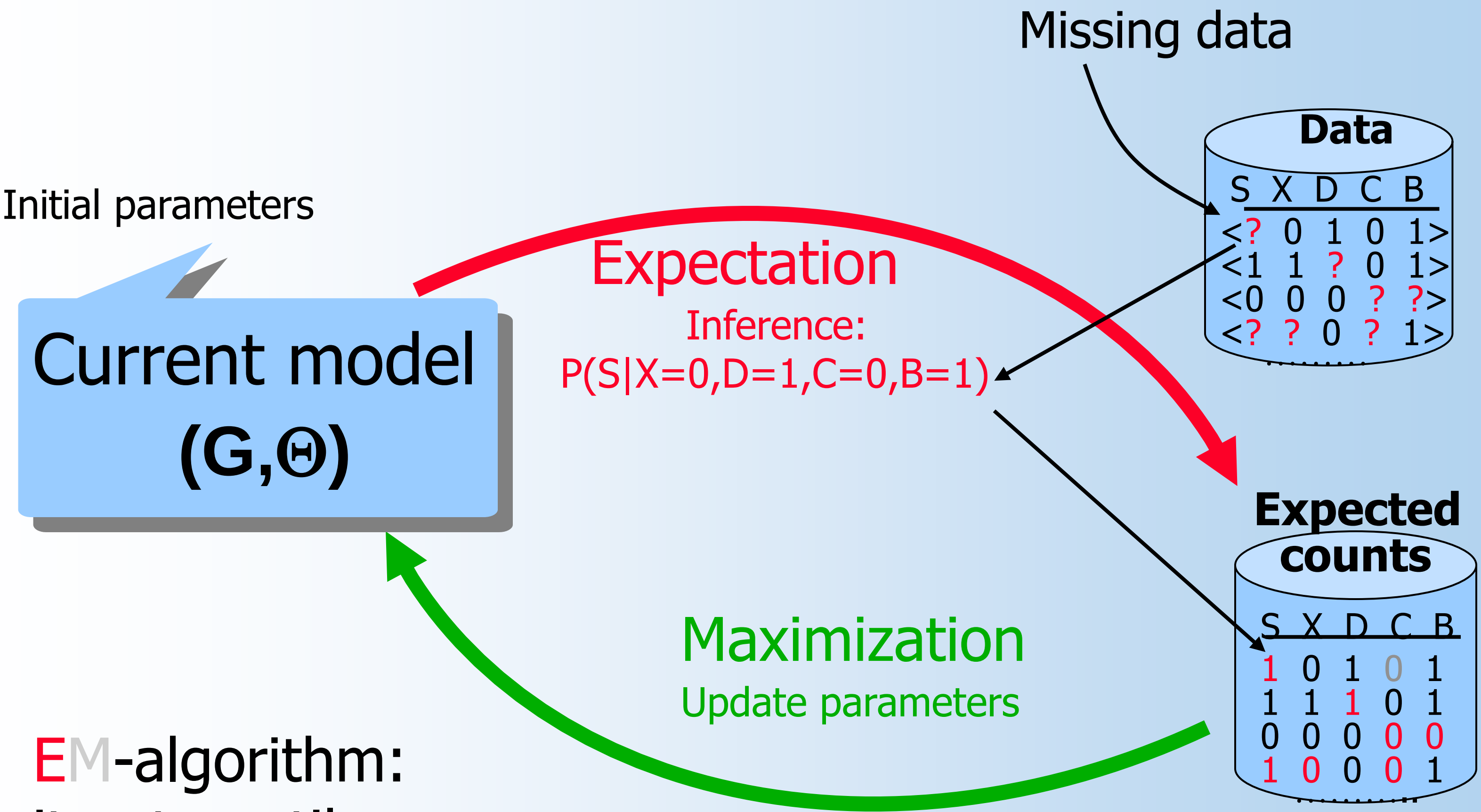
X	Y	#
H	H	1.3
T	H	0.4
H	T	1.7
T	T	1.6

EM (cont.)



Reiterate
Initial network (G,Θ_0)
X_1 X_2 X_3
H
Y_1 Y_2 Y_3
+
Training Data
Computation
(E-Step)
Expected Counts
$N(X_1)$
$N(X_2)$
$N(X_3)$
$N(H, X_1, X_1, X_3)$
$N(Y_1, H)$
$N(Y_2, H)$
$N(Y_3, H)$
Reparameterize
(M-Step)
Updated network (G,Θ_1)
X_1 X_2 X_3
H
Y_1 Y_2 Y_3

EM (cont.)



EM-algorithm:
iterate until convergence

The EM Algorithm for Bayes Nets

- First, random values are selected for the parameters in the CPTs for the entire network.
- Secondly, the needed weights are computed by using the Bayes net.
- Thirdly, these weights are in turn used to estimate new CPTs.
- Then, the second step and the third step are iterated until the CPTs converge.

EM in Practice

Initial parameters:
- Random parameter settings
- “Best” guess from other source

Stopping criteria:
- Small change in likelihood of data
- Small change in parameter values

Avoiding bad local maxima:
- Multiple restarts

Difficulties:
- More missing data $\Rightarrow$ many more local maxima
- Many hidden variables $\Rightarrow$ can result in over fitting the data