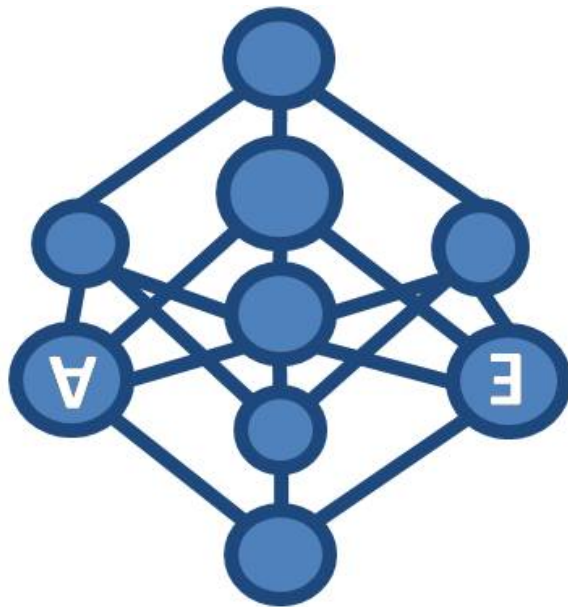


Probabilistic Graphical Models*

Bayesian Networks - Learning



TECHNISCHE
UNIVERSITÄT
DARMSTADT



*Thanks to Carlos Guestrin, Pedro Domingos and many others for making their slides publically available



So far

Representation and Inference ...

... but where do the numbers come from?



What's next

Learning Bayesian networks from data

1. **Parameter Estimation**
2. Model Selection aka **Structure Learning**

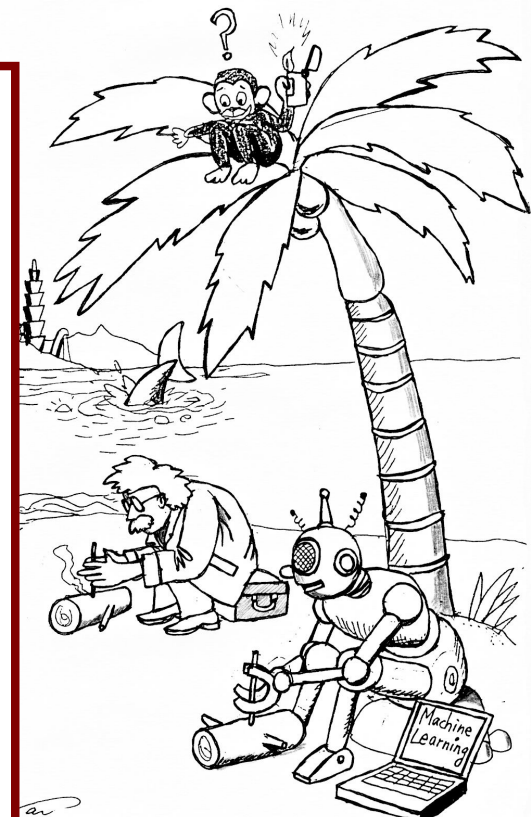


What is Learning?

Agents are said to learn if they improve their performance over time based on experience.

The problem of understanding intelligence is said to be the greatest problem in science today and “the” problem for this century – as deciphering the genetic code was for the second half of the last one... is the problem of learning represents a gateway to understanding intelligence in man and machines.

[Tomasso Poggio and Steven Smale, 2003]



Why bothering with learning?

- **Bottleneck of knowledge aquisition**
 - Expensive, difficult
 - Normally, no expert is around
- **Data is cheap !**
 - Huge amount of data available, e.g.
 - Literature Databases
 - Web mining, e.g. log files
 -

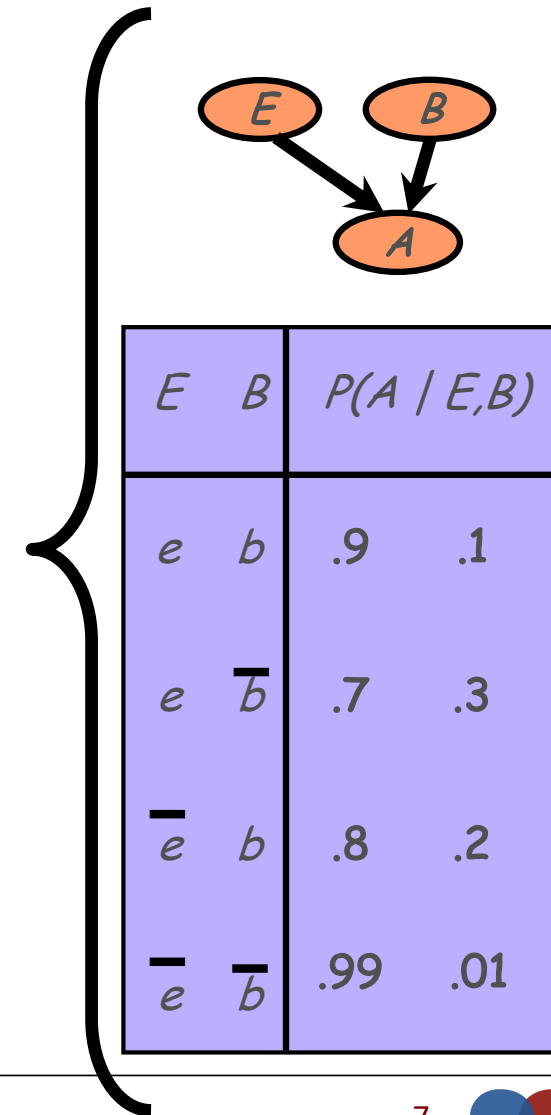
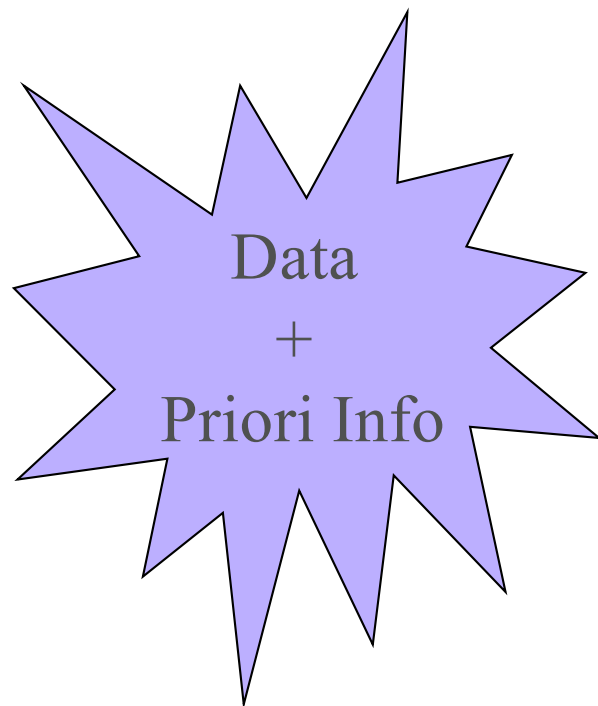


Why Learning Bayesian Networks?

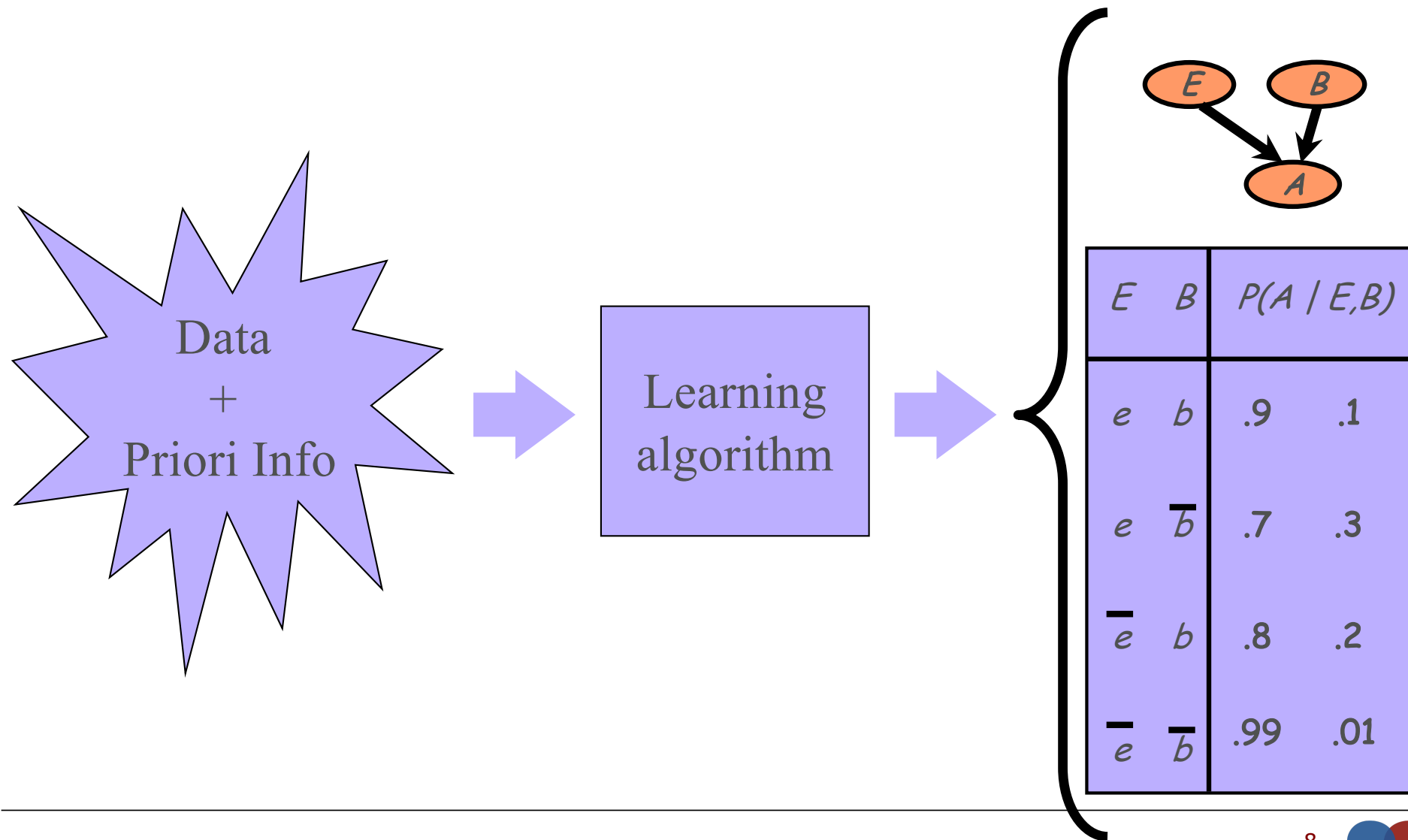
- Conditional independencies and graphical language capture structure of many real-world distributions
- **Graph structure provides much insight** into domain: “knowledge discovery”
- **Learned model can be used for many tasks**
- Automatically **dealing with missing data** and **hidden variables**



Learning With Bayesian Networks



Learning With Bayesian Networks



What does the data look like?

attributes/variables

complete data set

A1	A2	A3	A4	A5	A6	
true	true	false	true	false	false	X1
false	true	true	true	false	false	X2
...	⋮
true	false	false	false	true	true	XM

data cases



What does the data look like?

incomplete data set

A1	A2	A3	A4	A5	A6
true	true	?	true	false	false
?	true	?	?	false	false
...
true	false	?	false	true	?

- **Real-world data:**
states of some
random variables are
missing
 - E.g. medical diagnose:
not all patient are
subjects to all test
 - Parameter reduction,
e.g. clustering, ...



What does the data look like?

incomplete data set

A1	A2	A3	A4	A5	A6
true	true	?	true	false	false
?	true	?	?	false	false
...
true	false	?	false	true	?

- **Real-world data:**
states of some
random variables are
missing

- E.g. medical diagnose:
not all patient are
subjects to all test
- Parameter reduction,
e.g. clustering, ...

missing value



What does the data look like?

**hidden/
latent**

incomplete data set

A1	A2	A3	A4	A5	A6
true	true	?	true	false	false
?	true	?	?	false	false
...
true	false	?	false	true	?

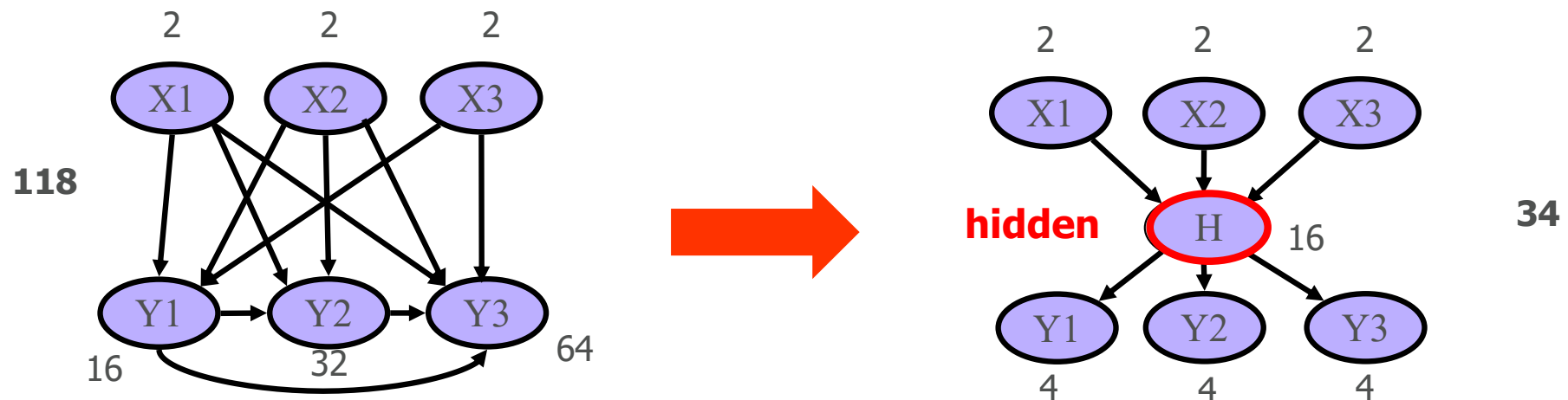
- **Real-world data:**
states of some
random variables are
missing

- E.g. medical diagnose:
not all patient are
subjects to all test
- Parameter reduction,
e.g. clustering, ...

missing value



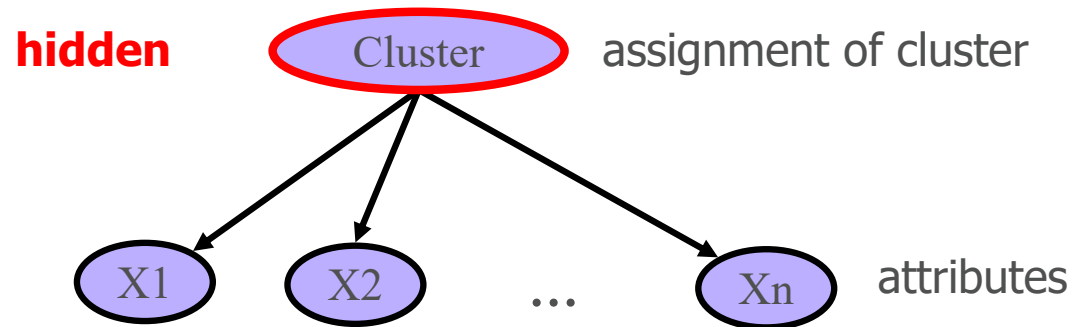
Hidden variable: Parameter Reduction



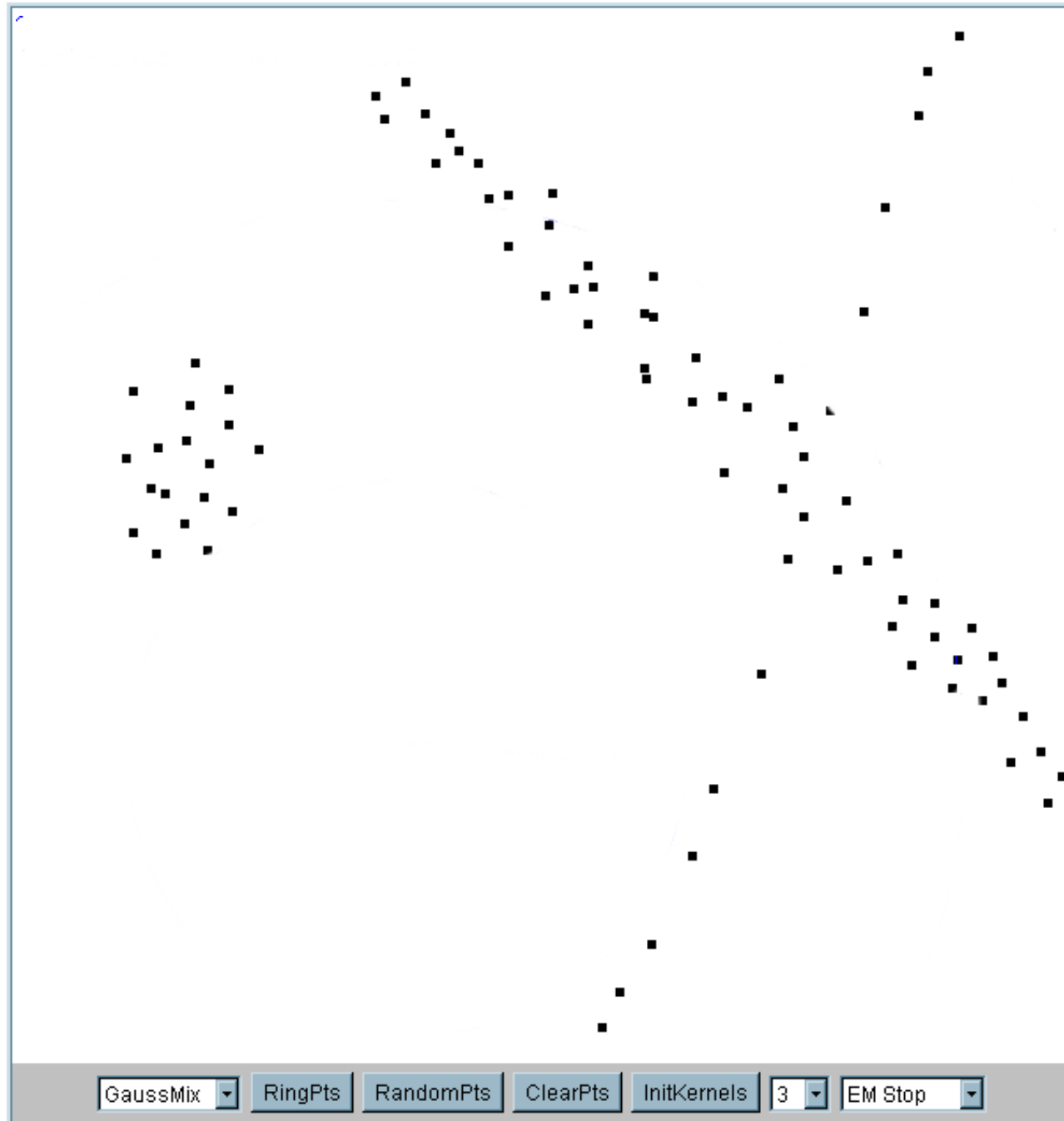
Hidden = latent = never observed



Hidden variable: Clustering

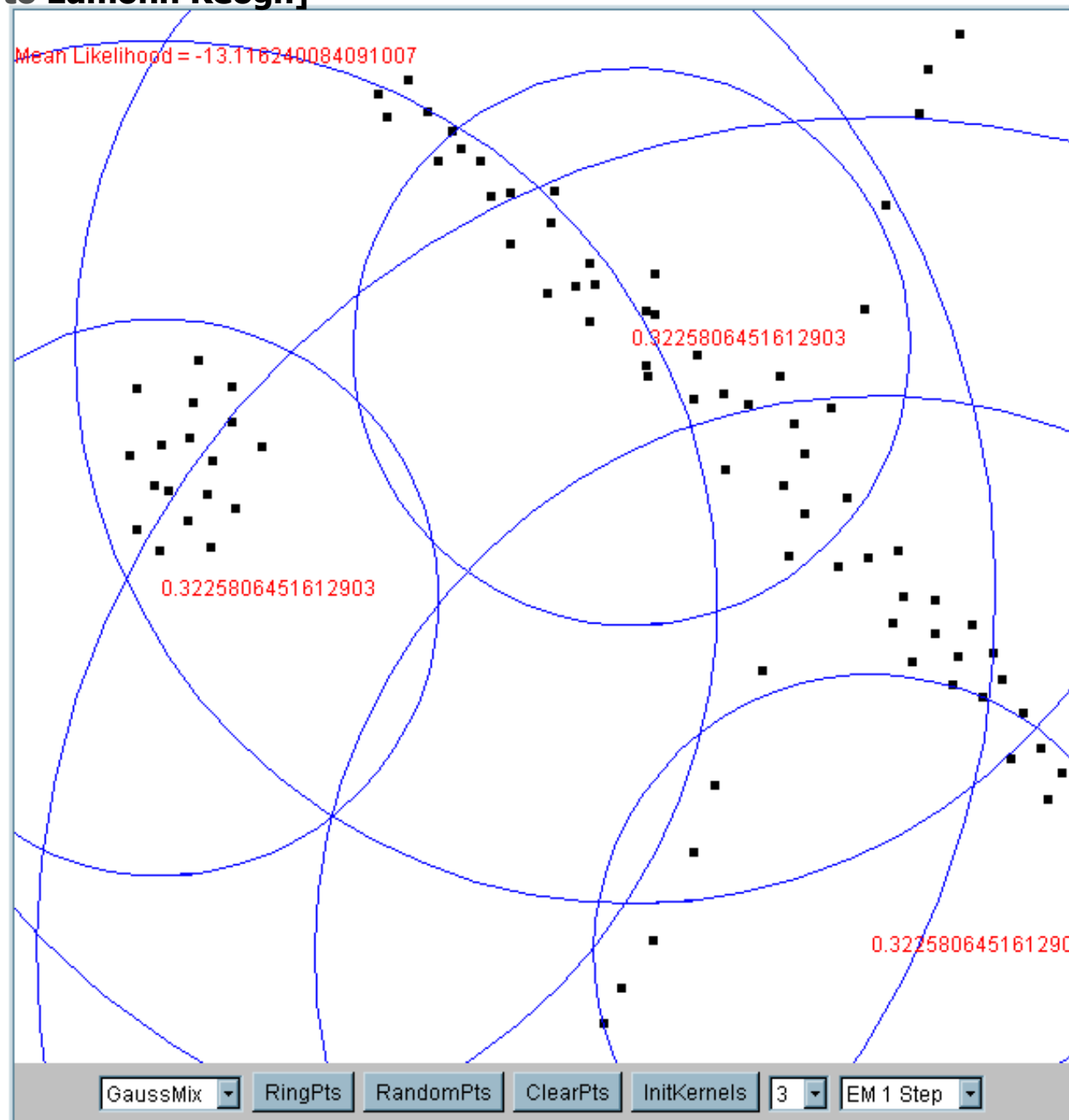


- Hidden variables also appear in **clustering**
- **Autoclass/Naïve Bayes/kMeans** model used by NASA for deep space exploration:
 - Hidden variable assigns class labels
 - Observed attributes are independent given the class

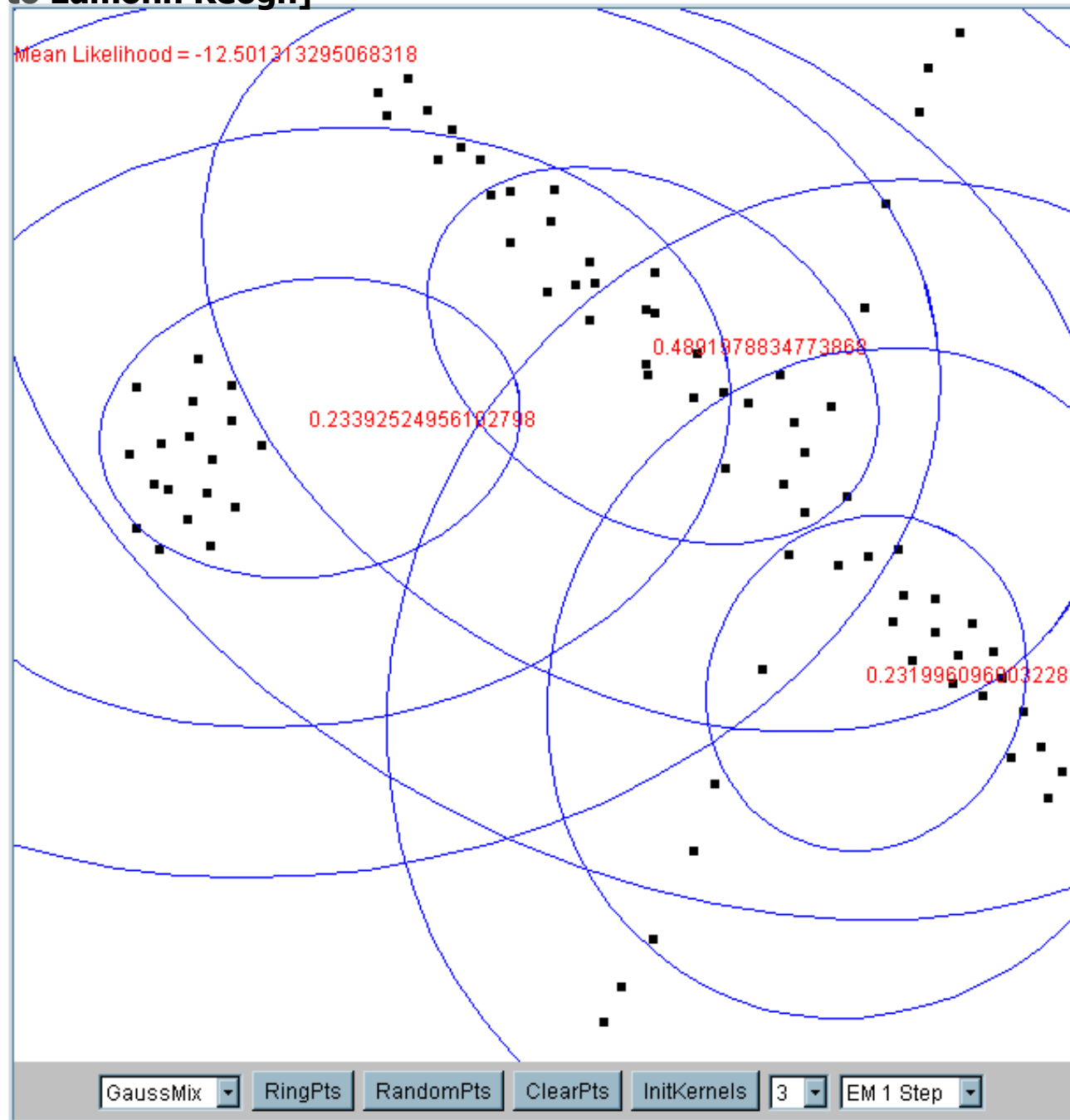


Iteration 1

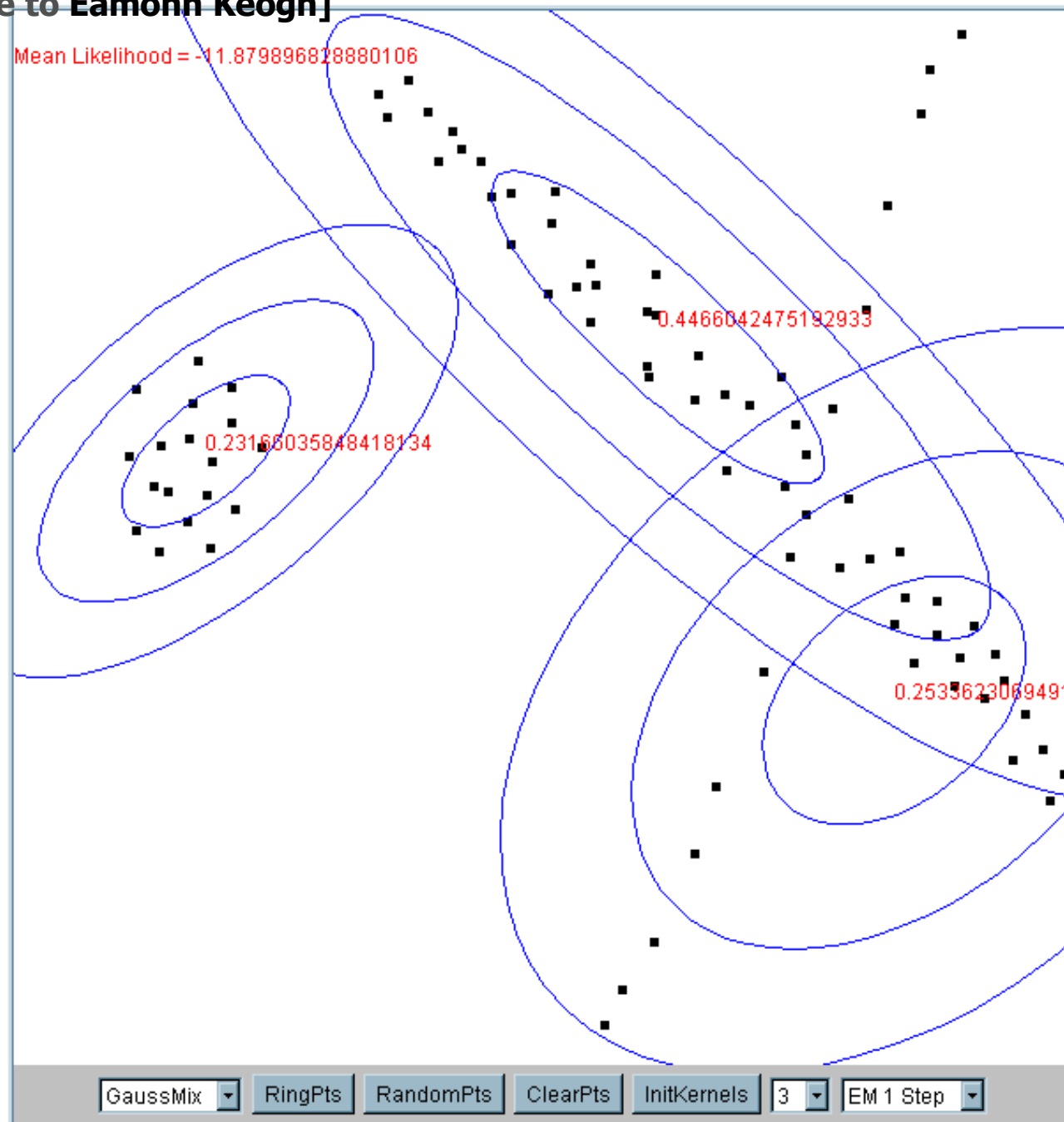
The cluster
means are
randomly
assigned



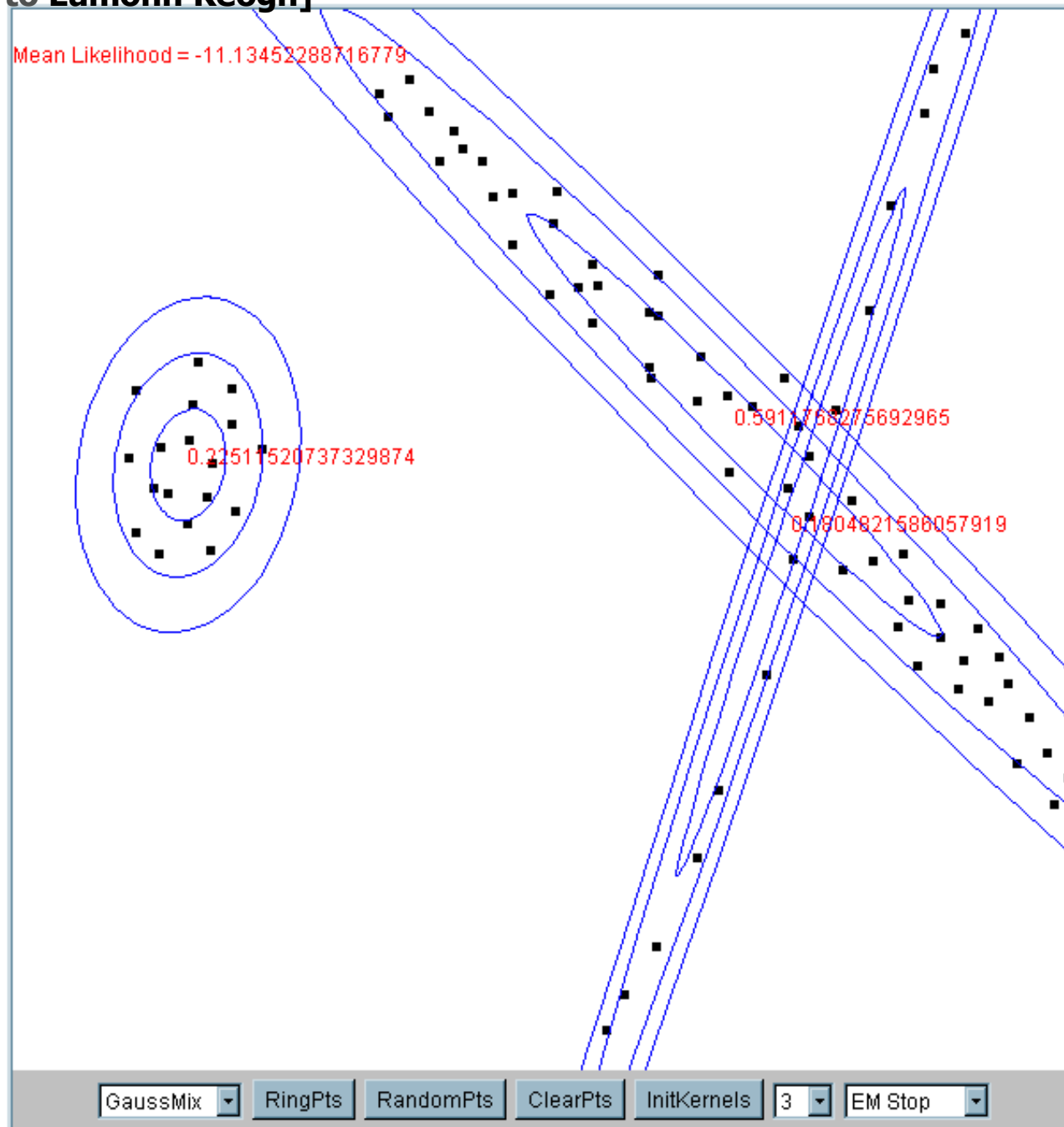
Iteration 2



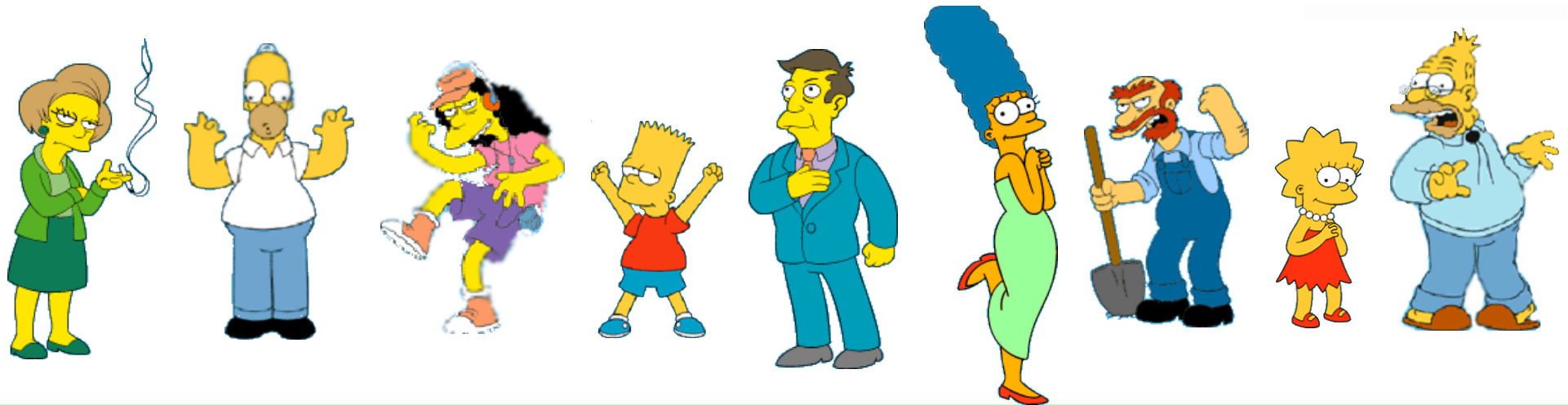
Iteration 5



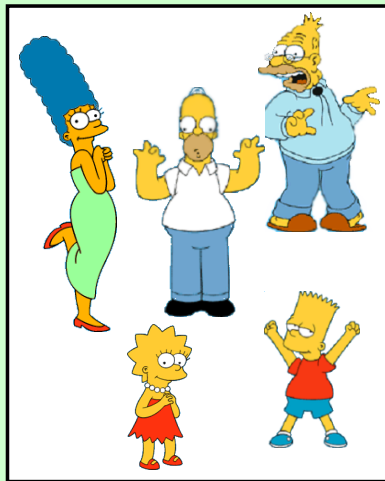
Iteration 25



What is a natural grouping among these objects?



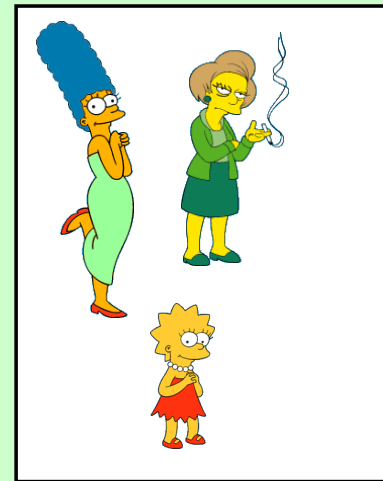
Clustering is subjective



Simpson's Family



School Employees



Females






Males

... and depends on your taste of similarity



„We know it when we see it“

Learning With Bayesian Networks

		Fixed structure 	Fixed variables 	Hidden variables 
observed	fully	Easiest problem counting	Selection of arcs New domain with no domain expert Data mining	
	Partially	Numerical, nonlinear optimization, Multiple calls to BNs, Difficult for large networks	Encompasses to difficult subproblem, „Only“ Structural EM is known	Scientific discovery



Parameter Estimation and IID

- Let $\mathcal{X} = \{X_1, X_2, \dots, X_n\}$ set of data over m RVs
- $X_i \in \mathcal{X}$ is called a *data case*
- **iid** - assumption:
 - All data cases are **i**ndependently sampled from **i**dentical **d**istributions

Find:

Parameters Θ of CPDs which match the data best



Maximum Likelihood - Parameter Estimation

What does „best matching“ mean ?

Find parameters Θ which have most likely produced the data



Maximum Likelihood - Parameter Estimation

- What does „best matching“ mean ?

1. MAP parameters $\Theta^* = \arg \max_{\Theta} P(\Theta|\mathcal{X})$

$$= \arg \max_{\Theta} P(\mathcal{X}|\Theta) \cdot \frac{P(\Theta)}{P(\mathcal{X})}$$

2. Data is equally likely for all parameters

3. All parameters are apriori equally likely



Maximum Likelihood - Parameter Estimation

- What does „best matching“ mean ?

Find:

ML parameters

Taking the log does not
affect the maximum

$$\Theta^* = \arg \max_{\Theta} P(\mathcal{X}|\Theta)$$

Likelihood $\mathcal{L}(\Theta|\mathcal{X})$: the params given the data

$$\Theta^* = \arg \max_{\Theta} \log P(\mathcal{X}|\Theta)$$

Log-Likelihood $\mathcal{LL}(\Theta|\mathcal{X})$







Maximum Likelihood

- One of the most commonly used estimators in statistics
- **Intuitively appealing**
- **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

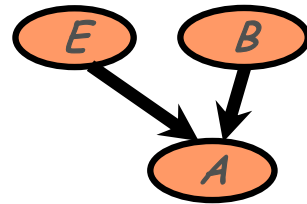


Learning With Bayesian Networks

		Fixed structure 	Fixed variables 	Hidden variables 
observed	fully	Easiest problem counting 	Selection of arcs New domain with no domain expert Data mining	
	Partially	Numerical, nonlinear optimization, Multiple calls to BNs, Difficult for large networks	Encompasses to difficult subproblem, „Only“ Structural EM is known	Scientific discovery

Known Structure, Complete Data

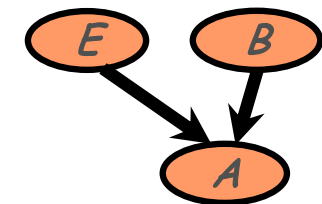
E, B, A
 <Y,N,N>
 <Y,N,Y>
 <N,N,Y>
 <N,Y,Y>
 .
 .
 <N,Y,Y>



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

Learning
algorithm

- Network structure is specified
 - Only estimation of parameters
- No missing data values



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

ML Parameter Estimation

A1	A2	A3	A4	A5	A6
true	true	false	true	false	false
false	true	true	true	false	false
...
true	false	false	false	true	true

$$\begin{aligned}
 \mathcal{LL}(\Theta|\mathcal{X}) &= \log P(X_1, X_2, \dots, X_n|\Theta) \\
 &\stackrel{\text{(iid)}}{=} \log \prod_{i=1}^n P(X_i|\Theta) \\
 &\stackrel{\log \prod}{=} \sum \log = \sum_{i=1}^n \log P(X_i|\Theta) = \sum_{i=1}^n \log P(x_i^1, x_i^2, \dots, x_i^m|\Theta) \\
 &= \sum_{i=1}^n \log \left(\prod_{j=1}^m P(x_i^j | \text{pa}(x_i^j), \Theta) \right) \quad \text{(BN semantics)} \\
 &\stackrel{\log \prod}{=} \sum \log = \sum_{i=1}^n \sum_{j=1}^m \log P(x_i^j | \text{pa}(x_i^j), \Theta) \\
 &= \sum_{j=1}^m \sum_{i=1}^n \log P(x_i^j | \text{pa}(x_i^j), \Theta_j) \quad \text{Only local parameters of family of } A_j \text{ involved} \\
 &= \sum_{j=1}^m \mathcal{LL}(\Theta_j|\mathcal{X}) \quad \text{Each factor individually !!}
 \end{aligned}$$

ML Parameter Estimation

$$\mathcal{LL}(\Theta|\mathcal{X}) = \log P(X_1, X_2, \dots, X_n|\Theta)$$

A1	A2	A3	A4	A5	A6
true	true	false	true	false	false
false	true	true	true	false	false
...
true	false	false	false	true	true

(iid)

$$= \log \prod_{i=1}^n P(X_i|\Theta)$$

Decomposability of the likelihood

$$= \sum \log \prod_{i=1}^n \sum_{j=1}^m \log P(x_i^j | \text{pa}(x_i^j), \Theta)$$

$$= \sum_{j=1}^m \sum_{i=1}^n \log P(x_i^j | \text{pa}(x_i^j), \Theta_j)$$

$$= \sum_{j=1}^m \mathcal{LL}(\Theta_j|\mathcal{X})$$

**Only local
parameters
of family of
 A_j involved**

**Each factor
individually !!**

Decomposability of Likelihood

If data set is **complete/fully observed** (i.e. no “?”)

- we can maximize each local likelihood function **independently**, and
- then **combine** the solutions to get an MLE solution
- This **decomposition** of the global problem to independent, local sub-problems allows us to come up with efficient solutions to the MLE problem



Likelihood for Multinominals

- Random variable V with $1, \dots, K$ values

$$P(V = k) = \theta_k \quad \sum_{k=1}^K \theta_k = 1$$

This constraint implies that the choice on θ_i influences the choice on θ_j ($i \neq j$)

- $\mathcal{LL}(\Theta_v | \mathcal{X}) = \sum_{k=1}^K \log \theta_k^{N_k} = \sum_{k=1}^K N_k \cdot \log \theta_k$

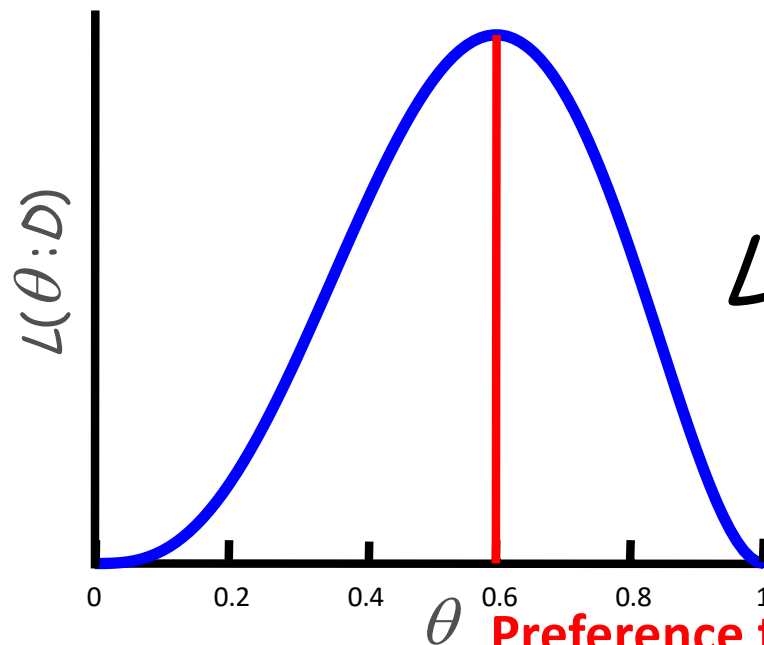
where N_k denotes the number of times we observe state k in the data (**the counts**)



Likelihood Function: Multinomials

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

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



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

Preference towards heads

General case:

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

Count of k^{th} outcome in D

Probability of
 k^{th} outcome

Likelihood for Binominals (2 states only)

- **Compute partial derivative**

$$\begin{aligned}\frac{\partial}{\partial \theta_i} \mathcal{LL}(\Theta_v | \mathcal{X}) &= \frac{\partial}{\partial \theta_i} (N_1 \log \theta_1 + N_2 \log(1 - \theta_1)) \\ &= \frac{N_1}{\theta_1} + \frac{N_2}{1 - \theta_1}\end{aligned}$$

$$\theta_1 + \theta_2 = 1$$

- **Set partial derivative zero**

$$\frac{\partial}{\partial \theta_i} \mathcal{LL}(\Theta_v | \mathcal{X}) = 0 \Leftrightarrow \frac{N_1}{\theta_1} + \frac{N_2}{1 - \theta_1} = 0$$

=> MLE is

$$\theta_1^* = \frac{N_1}{N_1 + N_2}$$



Likelihood for Binominals (2 states only)

- **Compute partial derivative**

$$\begin{aligned}\frac{\partial}{\partial \theta_i} \mathcal{LL}(\Theta_v | \mathcal{X}) &= \frac{\partial}{\partial \theta_i} (N_1 \log \theta_1 + N_2 \log(1 - \theta_1)) \\ &= \frac{N_1}{\theta_1} + \frac{N_2}{1 - \theta_1}\end{aligned}$$

$$\theta_1 + \theta_2 = 1$$

- **Set partial derivative zero**

In general, for multinomials (>2 states),
the MLE is

$$\theta_i^* = \frac{N_i}{\sum_j N_j}$$



Likelihood for Conditional Multinomials

- $P(V = k | \text{pa}(V) = \text{pa})$ binomial for each joint state pa of the parents of V :

$$P(k|1, 1), P(k|1, 2), P(k|2, 1), P(k|2, 2)$$

- $\mathcal{LL}(\Theta_v | \mathcal{X})$

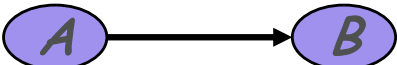




$$= \sum_{\text{pa}} \sum_{k=1}^K \log \theta_{k|\text{pa}}^{N_{k,\text{pa}}} = \sum_{\text{pa}} \sum_{k=1}^K N_{k,\text{pa}} \cdot \theta_{k|\text{pa}}$$

- MLE

$$\theta_{k|\text{pa}}^* = \frac{N_{k,\text{pa}}}{N_{\text{pa}}}$$

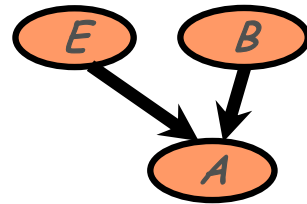


Learning With Bayesian Networks

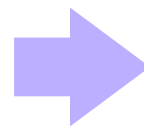
		Fixed structure	Fixed variables	Hidden variables
				
observed	fully	<p>Easiest problem</p> <p>counting</p> 	<p>Selection of arcs</p> <p>New domain with no domain expert</p> <p>Data mining</p>	
	Partially	<p>Numerical, nonlinear optimization,</p> <p>Multiple calls to BNs,</p> <p>Difficult for large networks</p> 	<p>Encompasses to difficult subproblem,</p> <p>„Only“ Structural EM is known</p>	<p>Scientific discovery</p>

Known Structure, Incomplete Data

E, B, A
 $\langle Y, ?, N \rangle$
 $\langle Y, N, ? \rangle$
 $\langle N, N, Y \rangle$
 $\langle N, Y, Y \rangle$
 \cdot
 \cdot
 $\langle ?, Y, Y \rangle$



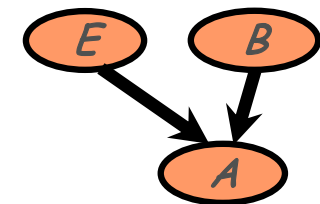
E	B	$P(A E, B)$	
e	b	?	?
e	\bar{b}	?	?
\bar{e}	b	?	?
\bar{e}	\bar{b}	?	?



Learning
algorithm



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



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

EM Idea

- If **data is complete**, ML parameter estimation is easy:
 - **simple counting** (1 iteration)
- But what if there are missing values, i.e., we are facing **incomplete data**?

1. **Complete data** (Imputation)

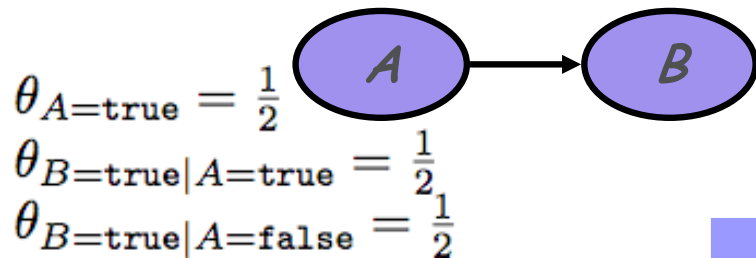
— most probable?, average?, ... value

2. **Count**

3. **Iterate**



EM Idea: complete the data



complete

$$P(B = \text{true} | A = \text{true}) = 0.5$$

$$P(B = \text{true} | A = \text{false}) = 0.5$$

incomplete data

A	B
true	true
true	?
false	true
true	false
false	?



complete data

expected counts



A	B	N
true	true	1.5
true	false	1.5
false	true	1.5
false	false	0.5

maximize

iterate

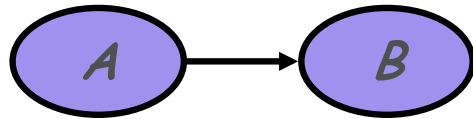
$$\theta_{A=\text{true}} = \frac{1.5+1.5}{1.5+1.5+1.5+0.5} = 0.6$$

$$\theta_{B=\text{true}|A=\text{true}} = \frac{1.5}{1.5+1.5} = 0.5$$

$$\theta_{B=\text{true}|A=\text{false}} = \frac{1.5}{1.5+0.5} = 0.75$$



EM Idea: complete the data



$$\theta_{A=\text{true}} = 0.6$$

$$\theta_{B=\text{true}|A=\text{true}} = 0.5$$

$$\theta_{B=\text{true}|A=\text{false}} = 0.875$$

complete

$$P(B = \text{true} | A = \text{true}) = 0.5$$

$$P(B = \text{true} | A = \text{false}) = 0.875$$

incomplete data

A	B
true	true
true	?
false	true
true	false
false	?



complete data

expected counts



A	B	N
true	true	1.5
true	false	1.5
false	true	1.875
false	false	0.125

maximize

iterate

$$\theta_{A=\text{true}} = \frac{1.5+1.5}{1.5+1.5+1.875+0.125} = 0.6$$

$$\theta_{B=\text{true}|A=\text{true}} = \frac{1.5}{1.5+1.5} = 0.5$$

$$\theta_{B=\text{true}|A=\text{false}} = \frac{1.875}{1.875+0.125} = 0.9375$$



Complete-data likelihood

incomplete-data likelihood

$$\Theta^* = \arg \max_{\Theta} \mathcal{L}(\Theta|\mathcal{X})$$

A1	A2	A3	A4	A5	A6
true	true	?	true	false	false
?	true	?	?	false	false
...
true	false	?	false	true	?

Assume complete data $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$ exists with

$$P(\mathcal{Z}|\Theta) = P(\mathcal{X}, \mathcal{Y}|\Theta) = P(\mathcal{Y}|\mathcal{X}, \Theta) \cdot P(\mathcal{X}|\Theta)$$

complete-data likelihood

$$\mathcal{L}(\Theta|\mathcal{Z}) = \mathcal{L}(\Theta|\mathcal{X}, \mathcal{Y}) = P(\mathcal{X}, \mathcal{Y}|\Theta)$$

$$\mathcal{LL}(\Theta|\mathcal{Z}) = \mathcal{LL}(\Theta|\mathcal{X}, \mathcal{Y}) = \log P(\mathcal{X}, \mathcal{Y}|\Theta)$$

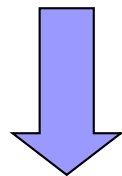


EM Algorithm - Abstract



Expectation Step

$$Q(\Theta, \Theta^{i-1}) = E [\mathcal{L}(\mathcal{Z}|\Theta) | \mathcal{X}, \Theta^{i-1}]$$

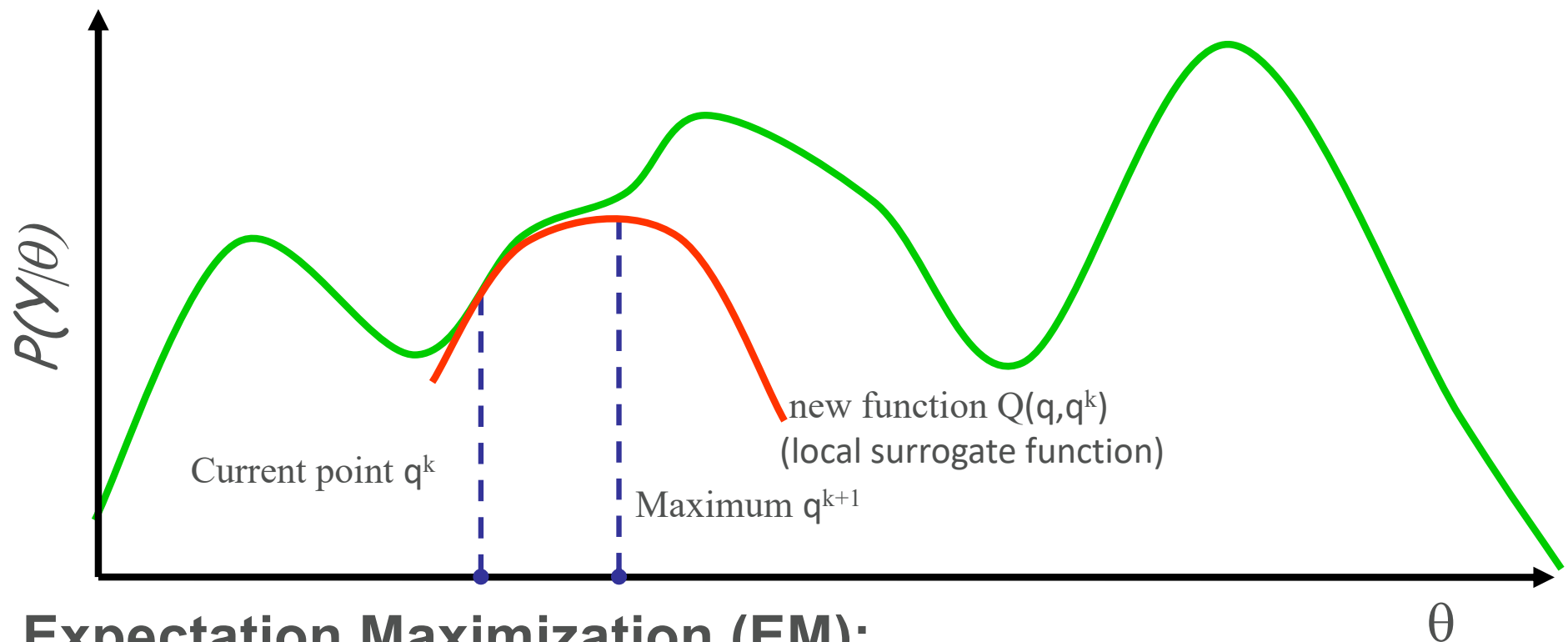


Maximization Step

$$\Theta^i = \arg \max_{\Theta} Q(\Theta, \Theta^{i-1})$$



EM Algorithm - Principle



Expectation Maximization (EM):

Construct an new function based on the “current point” (which “behaves well”)

Property: The maximum of the new function has a better scoring then the current point.



EM for Multinominals

- Random variable V with $1, \dots, K$ values

$$P(V = k) = \theta_k \quad \sum_{k=1}^K \theta_k = 1$$

- $Q(\Theta_v, \Theta') = \sum_{k=1}^K \log \theta_k^{EN_k} = \sum_{k=1}^K \log EN_k \cdot \theta_k$

where EN_k are the **expected counts** of state k in the data, i.e.

$$EN_k = \sum_{i=1}^m P(k|X_i)$$

- „MLE“:

$$\frac{EN_i}{\sum_k EN_k}$$

EM for Conditional Multinominals

- $P(V = k | \text{pa}(V) = \text{pa})$ multinomial for each joint state pa of the parents of V :

$$P(k|1, 1), P(k|1, 2), P(k|2, 1), P(k|2, 2)$$

$$Q(\Theta_v, \Theta')$$

$$= \sum_{\text{pa}} \sum_{k=1}^K \log \theta_{k|\text{pa}}^{EN_{k,\text{pa}}} = \sum_{\text{pa}} \sum_{k=1}^K EN_{k,\text{pa}} \cdot \theta_{k|\text{pa}}$$

- „MLE“

$$\theta_{k|\text{pa}}^* = \frac{EN_{k,\text{pa}}}{EN_{\text{pa}}}$$

Learning Parameters: incomplete data



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Non-decomposable likelihood (missing value, hidden nodes)

Initial parameters

Current model

Expectation

Inference:
 $P(S|X=0,D=1,C=0,B=1)$

Data

S	X	D	C	B
<?>	0	1	0	1>
<1>	1	?>	0	1>
<0>	0	0	?>	?>
<?>	?>	0	?>	1>

Expected counts

S	X	D	C	B
1	0	1	0	1
1	1	1	0	1
0	0	0	0	0
1	0	0	0	1

Maximization

Update parameters
(ML, MAP)

EM-algorithm:
iterate until convergence

$$\sum_{i=1}^m P(k, \text{pa} | X_i)$$

Learning Parameters using EM: incomplete data

1. Initialize parameters
2. Compute pseudo counts for each variable

$$\theta_{k|\text{pa}}^* = \frac{\sum_{i=1}^m P(k, \text{pa} | X_i)}{\sum_{i=1}^m P(\text{pa} | X_i)}$$

junction tree
algorithm

3. Set parameters to the (completed) ML estimates
4. If not converged, iterate to 2



Monotonicity

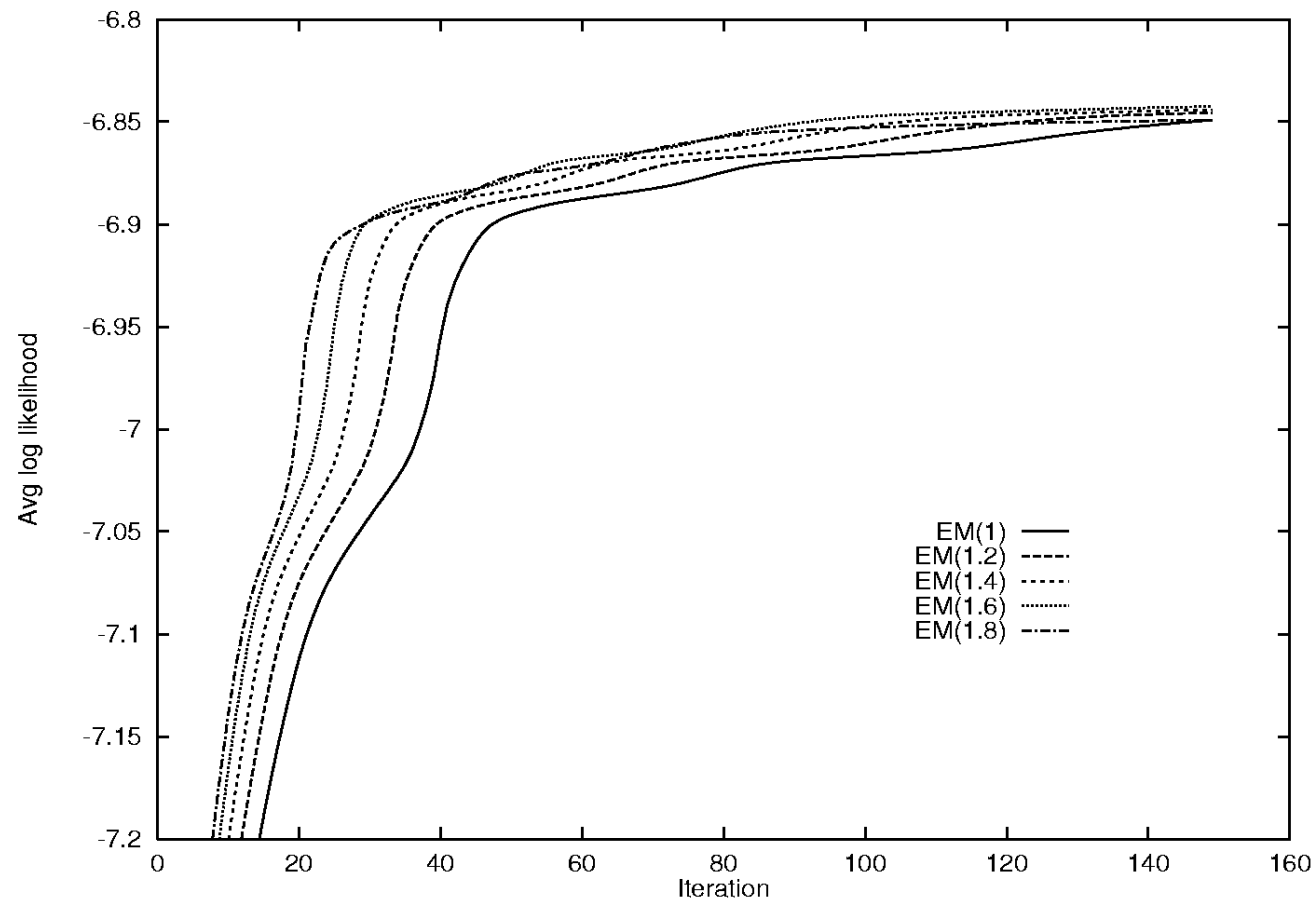
- (Dempster, Laird, Rubin '77): the incomplete-data likelihood function is not decreased after an EM iteration

$$\mathcal{L}(\Theta^i | \mathcal{X}) \geq \mathcal{L}(\Theta^{i-1} | \mathcal{X})$$

- (discrete) Bayesian networks: for any initial, non-uniform value the EM algorithm converges to a (local or global) maximum.



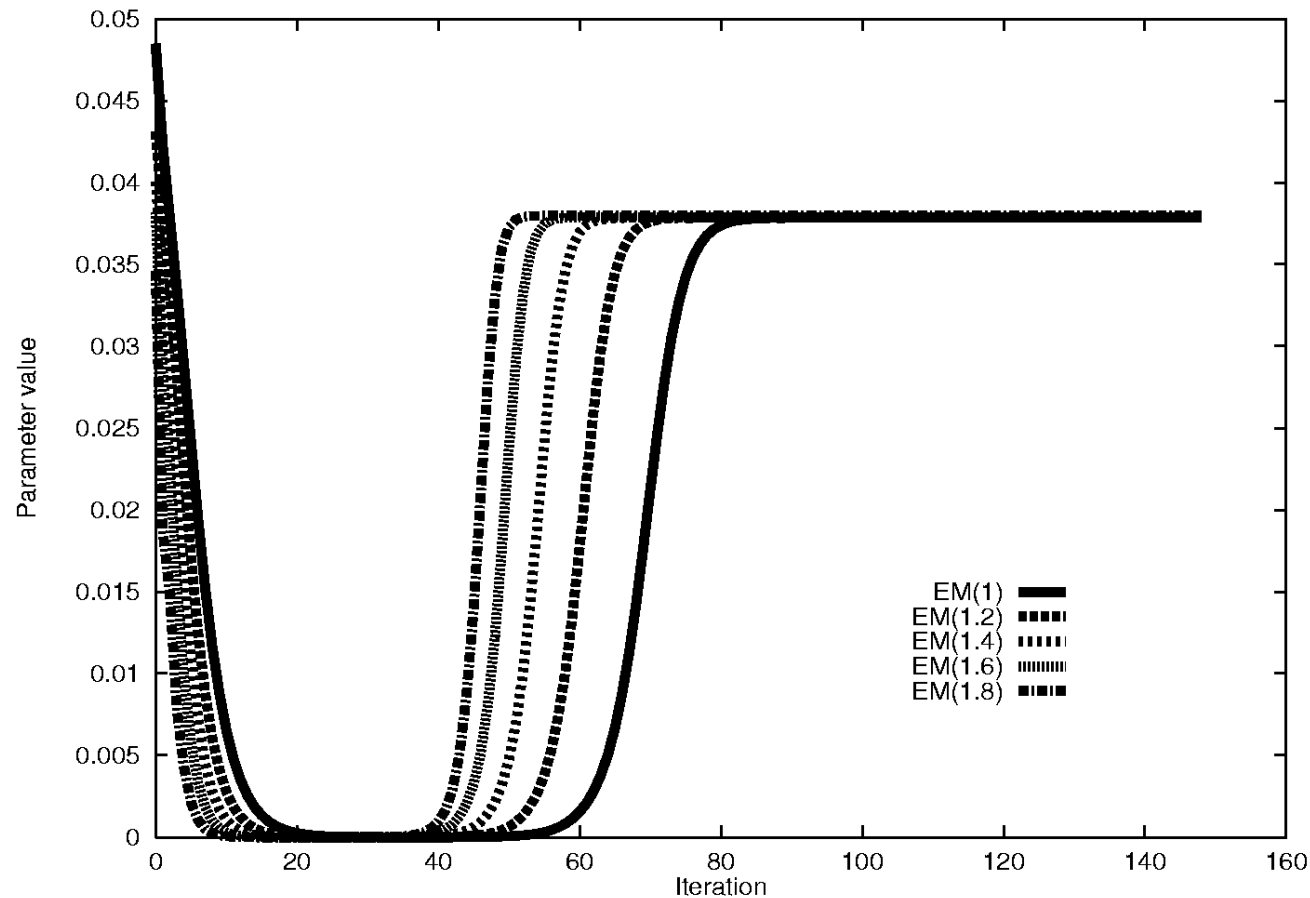
LL on training set (Alarm)



Experiment by Bauer, Koller and Singer [UAI97]



Parameter value (Alarm)



Experiment by Bauer, Koller and Singer [UAI97]



EM in Practice

Initial parameters:

- Randomly
- “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**



Gradient Ascent

- Main result

$$\frac{\partial \mathcal{L}(\Theta | \mathcal{X})}{\partial \theta_{k|pa}} = \frac{1}{\theta_{k|pa}} \sum_{j=1}^m \log P(k, pa | X_j, \Theta)$$

- Requires same BN inference computations as EM

- **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



What you need to know

- Parameter estimation is a basic task for learning with Bayesian networks
- Due to missing values non-linear optimization
 - EM, Gradient, ...
- EM for multi-nominal random variables
 - Fully observed data: counting
 - Partially observed data: pseudo counts
- Junction tree to do multiple inference

