Statistical Methods in AI/ML
Recap

Vibhav Gogate
The University of Texas at Dallas

What we studied in a nutshell?

- Representation
  - Graphical
  - Template-based
  - Logical
- Inference
  - Given a statistical model and a query, answer the query
- Learning
  - Given data, learn a representation

Probability Theory

- $0 \leq \Pr(x) \leq 1$
- $\Pr(x) = \sum_{i=1}^{n} \Pr(x \wedge a_i)$
  - $a_1, \ldots, a_n$ is a set of mutually exclusive and exhaustive events
- $\Pr(x \vee y) = \Pr(x) + \Pr(y) - \Pr(x \wedge y)$
- $\Pr(x \wedge y) = \Pr(x) \Pr(y)$
  - x and y are independent events
- $Product\ or\ Chain\ rule$:
  - $\Pr(x_1 \wedge \cdots \wedge x_n) = \prod_{i=1}^{n} \Pr(x_i | x_1 \wedge \cdots \wedge x_{i-1})$
- $Bayes\ rule$: $\Pr(x|y) = \frac{\Pr(y|x)\Pr(x)}{\Pr(y)}$

Conditional Independence (CI) Properties

- $I(\mathbf{X},\mathbf{Z},\mathbf{Y})$: $Pr(\mathbf{X},\mathbf{Y}|\mathbf{Z}) = Pr(\mathbf{X}|\mathbf{Z})Pr(\mathbf{Y}|\mathbf{Z})$
- Symmetry: $I(\mathbf{X},\mathbf{Z},\mathbf{Y}) \Rightarrow I(\mathbf{Y},\mathbf{Z},\mathbf{X})$
- Decomposition: $I(\mathbf{X},\mathbf{Z},\mathbf{Y}\cup\mathbf{W}) \Rightarrow I(\mathbf{X},\mathbf{Z},\mathbf{Y})$
- Weak Union: $I(\mathbf{X},\mathbf{Z},\mathbf{Y}\cup\mathbf{W}) \Rightarrow I(\mathbf{X},\mathbf{Z}\cup\mathbf{W},\mathbf{Y})$
- Contraction:
  - $I(\mathbf{X},\mathbf{Y}\cup\mathbf{Z},\mathbf{W})$ & $I(\mathbf{X},\mathbf{Z},\mathbf{Y}) \Rightarrow I(\mathbf{X},\mathbf{Z},\mathbf{Y}\cup\mathbf{W})$
- If distribution is positive, Intersection:
  - $I(\mathbf{X},\mathbf{Z}\cup\mathbf{W},\mathbf{Y})$ & $I(\mathbf{X},\mathbf{Z}\cup\mathbf{Y},\mathbf{W}) \Rightarrow I(\mathbf{X},\mathbf{Z},\mathbf{Y}\cup\mathbf{W})$

Representation: Graphical Representations

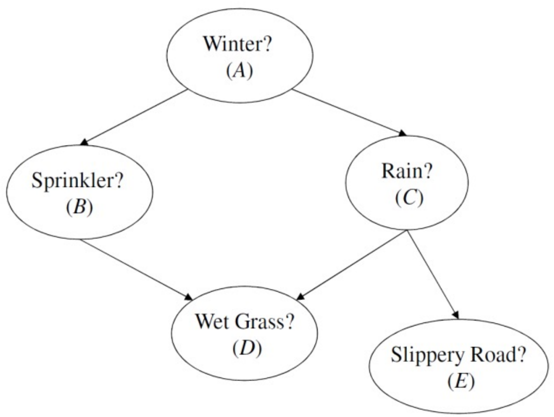
- From a probability distribution to a graph
- Graph properties vs conditional independence properties
- A graph can be viewed as:
  - View 1: A data structure for compactly representing a joint distribution
  - View 2: Compact representation for a set of conditional independence assumption
  - Both views are equivalent
- Bayesian networks (Directed Acyclic graph)
- Markov networks (Undirected graph)

Concept of I-map, D-map and P-map

- A graph represents conditional independence assumptions
- A graph G is an I-map of Pr if $I(G) \subseteq I(Pr)$
- A graph G is a D-map of Pr if $I(Pr) \subseteq I(G)$
- A graph G is a P-map of Pr if $I(G)=I(Pr)$
- Minimal I-maps
  – Remove an edge from G and it ceases to be an I-map.

Bayesian networks: Compact Representation of the Joint distribution

- $\Pr(x_1, \ldots, x_n) = \prod_{i=1}^{n} \Theta_{x_i|pa(xi)}$



A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

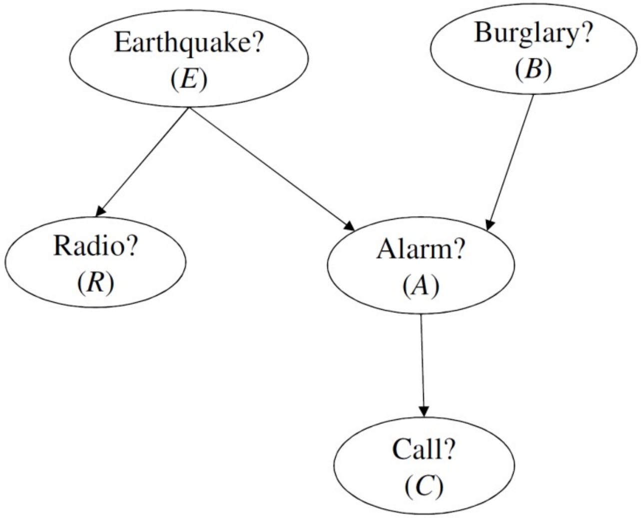
A	Θ_A
true	.6
false	.4

B	C	D	$\Theta_{D\|B,C}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

Bayesian networks: Compact representation of Conditional Independence assumptions

- Derive others using CI properties.



Markovian assumptions, Markov(G):

$$I(C, A, \{B, E, R\})$$
$$I(R, E, \{A, B, C\})$$
$$I(A, \{B, E\}, R)$$
$$I(B, \emptyset, \{E, R\})$$
$$I(E, \emptyset, B)$$

Variables B and E have no parents, hence, they are marginally independent of their non-descendants.

D-separation

- Graphical test of conditional independence
- $I(G)=\text{d-sep}_G$

Deciding $\mathrm{dsep}_G(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$ is equivalent to testing whether $\mathbf{X}$ and $\mathbf{Y}$ are disconnected in a new DAG G' obtained by pruning DAG G

- Delete any leaf node W from DAG G as long as W not in $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$. Repeat until no more nodes can be deleted.
- Delete all edges outgoing from nodes in $\mathbf{Z}$.

Decided in time and space that are linear in the size of DAG G

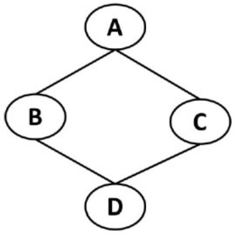
Constructing Minimal I-maps

Given a distribution Pr, how can we construct a DAG G which is guaranteed to be a minimal I-MAP of Pr?

Given an ordering $X_1, \ldots, X_n$ of the variables in Pr:

- Start with an empty DAG G (no edges)
- Consider the variables X_i one by one, for $i = 1, \ldots, n$
- For each variable X_i, identify a minimal subset $\mathbf{P}$ of the variables in $X_1, \ldots, X_{i-1}$ such that
  - $I_{\mathrm{Pr}}(X_i, \mathbf{P}, \{X_1, \ldots, X_{i-1}\} \setminus \mathbf{P})$

Markov networks: compact representation of the joint distribution



- Normalized product of all factors (called the Gibbs distribution).
- $\Pr(a, b, c, d) = \frac{1}{Z}\phi(a, b) \times \phi(b, c) \times \phi(c, d) \times \phi(a, d)$
- Z is a normalizing constant, often called the partition function
- $Z = \sum_{a,b,c,d} \phi(a, b) \times \phi(b, c) \times \phi(c, d) \times \phi(a, d)$

Example: What is the distribution represented by:
$\phi(a, b) = \phi(b, c) = (10, 1, 1, 10)$
$\phi(b, c) = \phi(c, d) = (5, 1, 1, 5)$

Markov networks: Compact representation of CI assumptions

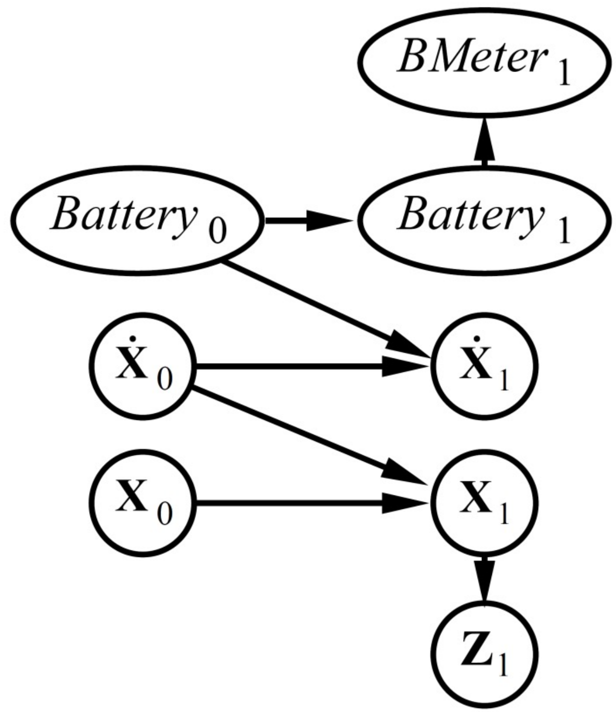
- Simpler: Graph separation I(**X**,**Z**,**Y**) if **X** and **Y** become disconnected after removing **Z**
- Converting a Bayesian network to a Markov network
- Converting a Markov network to a Bayesian network
  - Make the Markov network Chordal
- Chordal graphs lie at the intersection of the two.

Other Representations

- Factor Graphs
- Formula-based Representations
  - Formulas with weights attached to them
- Log-Linear models
  - $\Pr(\boldsymbol{x}) = \frac{1}{Z}\exp(\sum_i w_i f_i(\boldsymbol{x}))$
  - f_i is a formula or a feature
  - w_i is the weight of the formula = log(potential-value)

Dynamic Bayesian networks

- A template for generating a Bayesian network
  - Parameter: # of time-slices



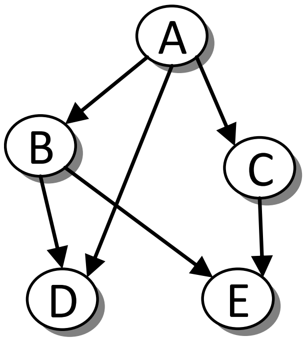
Answering Queries: Inference

- Queries
  - (PE) Probability of Evidence (Partition function)
  - (MAR) Posterior Marginals: $P(X_i|e)$
  - (MPE) Most Probable Explanation
  - (MAP) Maximum a Posteriori

Exact Algorithms for PE and MAR: Elimination

- Bucket/Variable Elimination for PE
- Junction tree algorithm for MAR
  - Sum-product message passing
- Complexity Analysis
  - Time and Space exponential in the treewidth of the primal/interaction graph
    - Make the graph chordal
    - Construct a tree decomposition
  - No difference at inference time between Bayesian and Markov networks!

Bucket Elimination [Dechter96]



<u>Query:</u> $P(a \mid e=0) \propto P(a, e=0)$ <u>Elimination Order</u>: d,e,b,c

$$P(a,e=0) = \sum_{c,b,e=0,d} P(a)P(b \mid a)P(c \mid a)P(d \mid a,b)P(e \mid b,c)$$

$$= P(a)\sum_{c} P(c \mid a)\sum_{b} P(b \mid a)\sum_{e=0} P(e \mid b,c)\sum_{d} P(d \mid a,b)$$

<u>Original Functions</u>

D: $P(d \mid a,b)$

E: $P(e \mid b,c)$

B: $P(b \mid a)$

C: $P(c \mid a)$

A: $P(a)$

<u>Messages</u>

$$f_D(a,b) = \sum_{d} P(d \mid a,b)$$

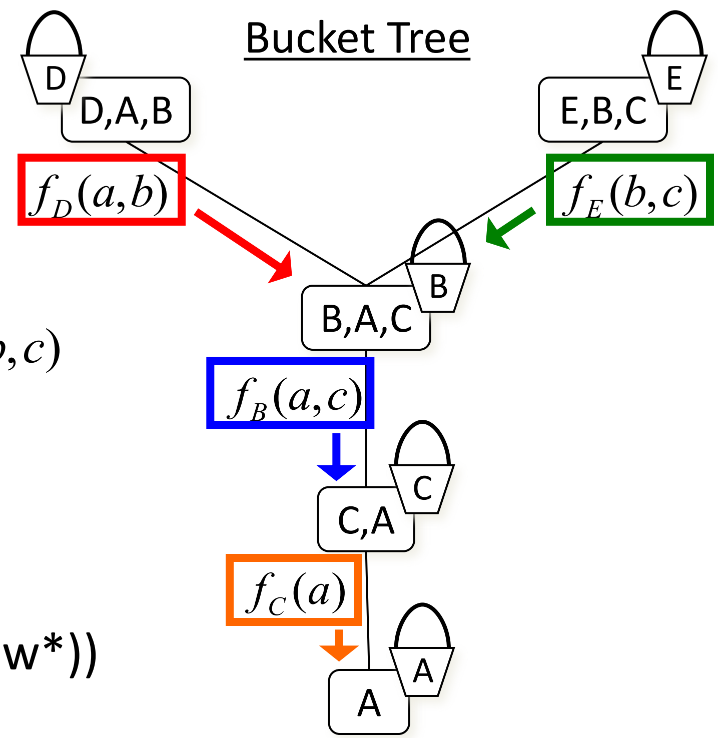
$$f_E(b,c) = P(e=0 \mid b,c)$$

$$f_B(a,c) = \sum_{b} P(b \mid a) f_D(a,b) f_E(b,c)$$

$$f_C(a) = \sum_{c} P(c \mid a) f_B(a,c)$$

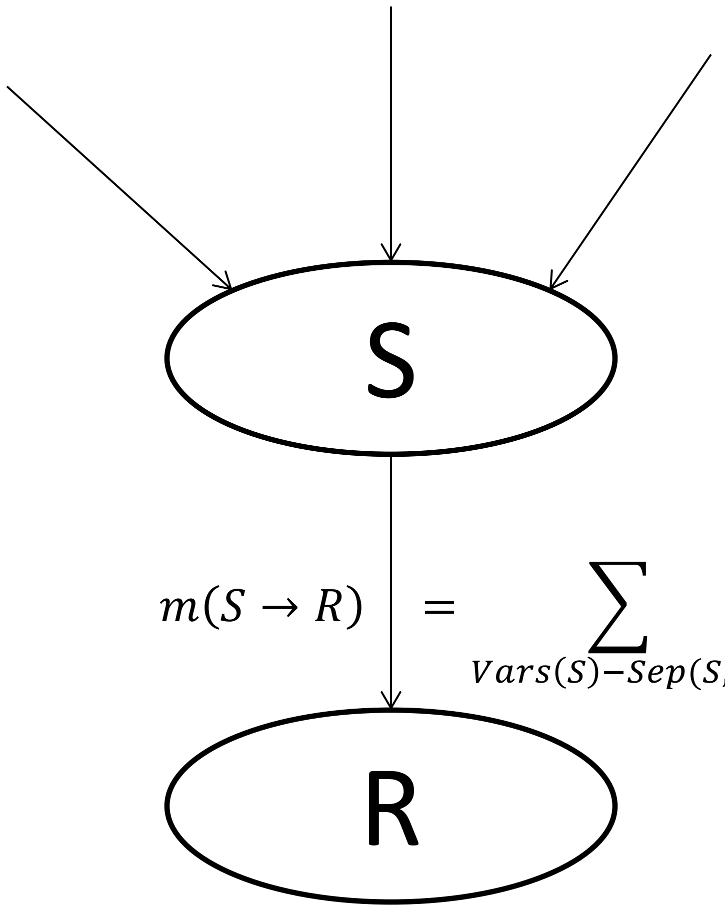
$$P(a,e=0) = p(A) f_C(a)$$

<u>Bucket Tree</u>



Time/Space is O(exp(w*))

Message passing Equations



- Multiply all received messages except from R
- Multiply all functions
- Sum-out all variables except the separator

$$m(S \rightarrow R) = \sum_{Vars(S)-Sep(S,R)} \prod_{f \in functions(S)} f \prod_{G \in Neighbors(S)-R} m(\mathrm{G} \rightarrow \mathrm{R})$$

Exact Algorithms for PE and MAR: Search

- AND/OR Search spaces
  - Time and Space tradeoffs
  - Pseudo Tree and Context
  - Tree vs Graph Search
- w-cutset conditioning
- Formula-based Probabilistic Inference
  - Weighted model counting
  - Determinism and Context Specific independence
  - Unit propagation and logical inference

AND/OR Tree DFS Algorithm

$P(E \mid A, B)$

A	B	E=0	E=1
0	0	.4	.6
0	1	.5	.5
1	0	.7	.3
1	1	.2	.8

Evidence: E=0

$P(B \mid A)$

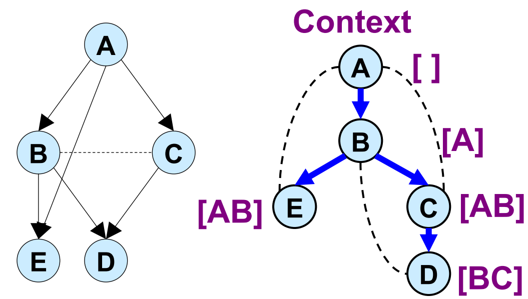
A	B=0	B=1
0	.4	.6
1	.1	.9

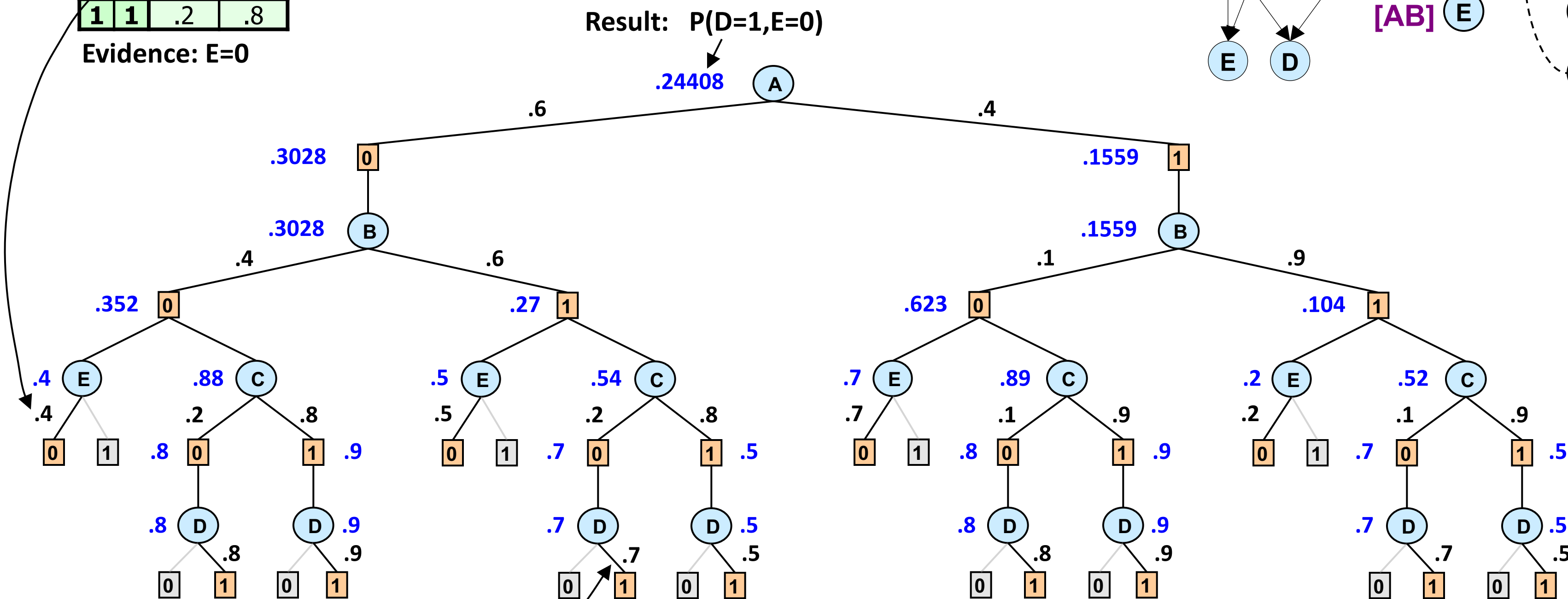
$P(C \mid A)$

A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4





$P(D \mid B, C)$

B	C	D=0	D=1
0	0	.2	.8
0	1	.1	.9
1	0	.3	.7
1	1	.5	.5

Evidence: D=1

OR node: Marginalization operator (summation)

AND node: Combination operator (product)

Value of node = updated belief for sub-problem below

AND/OR Graph DFS Algorithm

$P(E \mid A, B)$

A	B	E=0	E=1
0	0	.4	.6
0	1	.5	.5
1	0	.7	.3
1	1	.2	.8

Evidence: E=0

$P(B \mid A)$

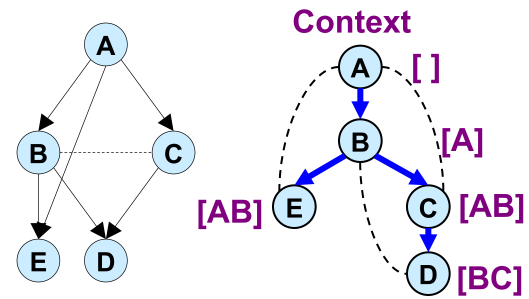
A	B=0	B=1
0	.4	.6
1	.1	.9

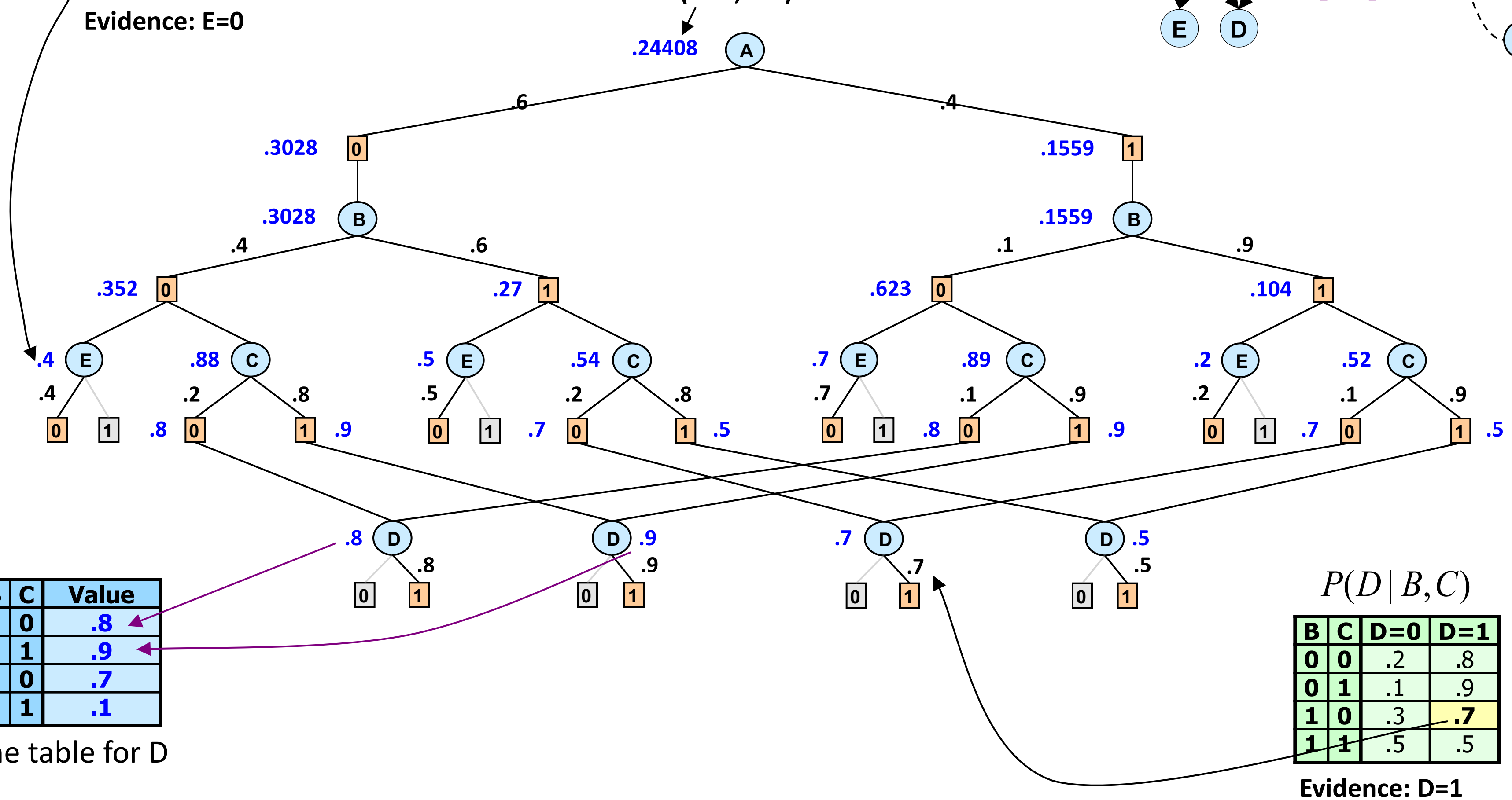
$P(C \mid A)$

A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4





B	C	Value
0	0	.8
0	1	.9
1	0	.7
1	1	.1

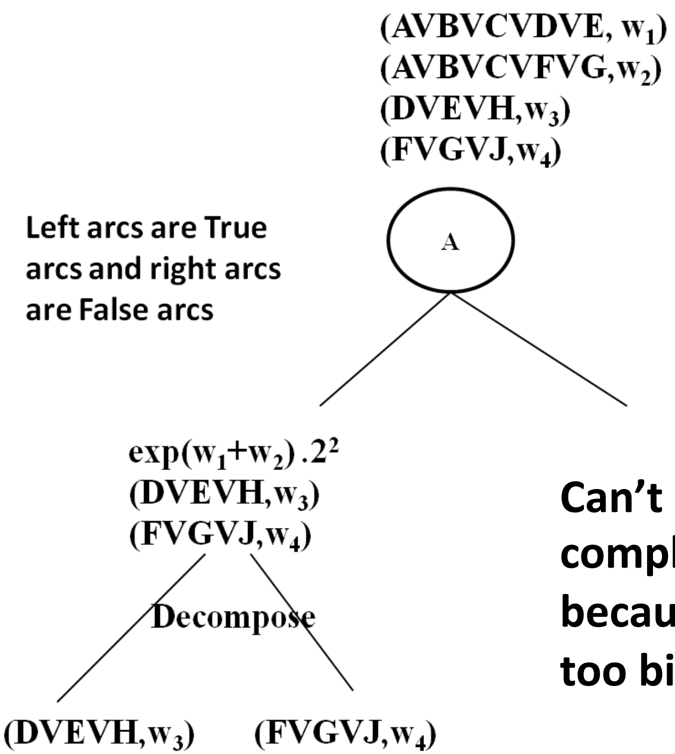
Cache table for D

$P(D \mid B, C)$

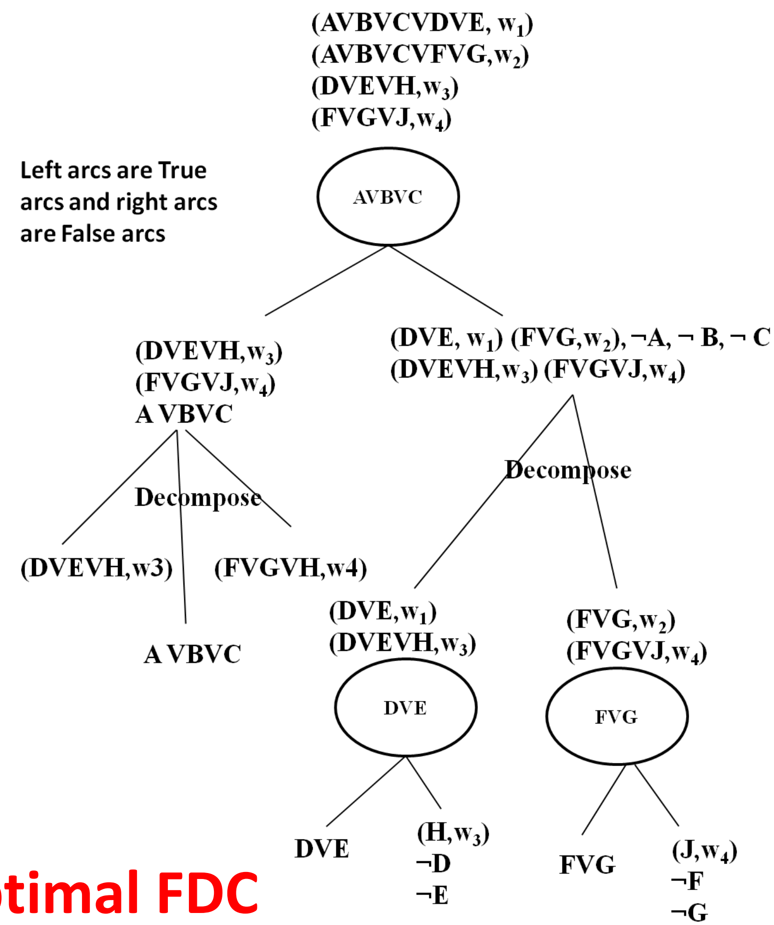
B	C	D=0	D=1
0	0	.2	.8
0	1	.1	.9
1	0	.3	.7
1	1	.5	.5

Evidence: D=1

Efficiency: Example



Can't draw it completely because it is too big



Optimal VDC explores 12 leaf nodes.

Optimal FDC explores 7 leaf nodes.

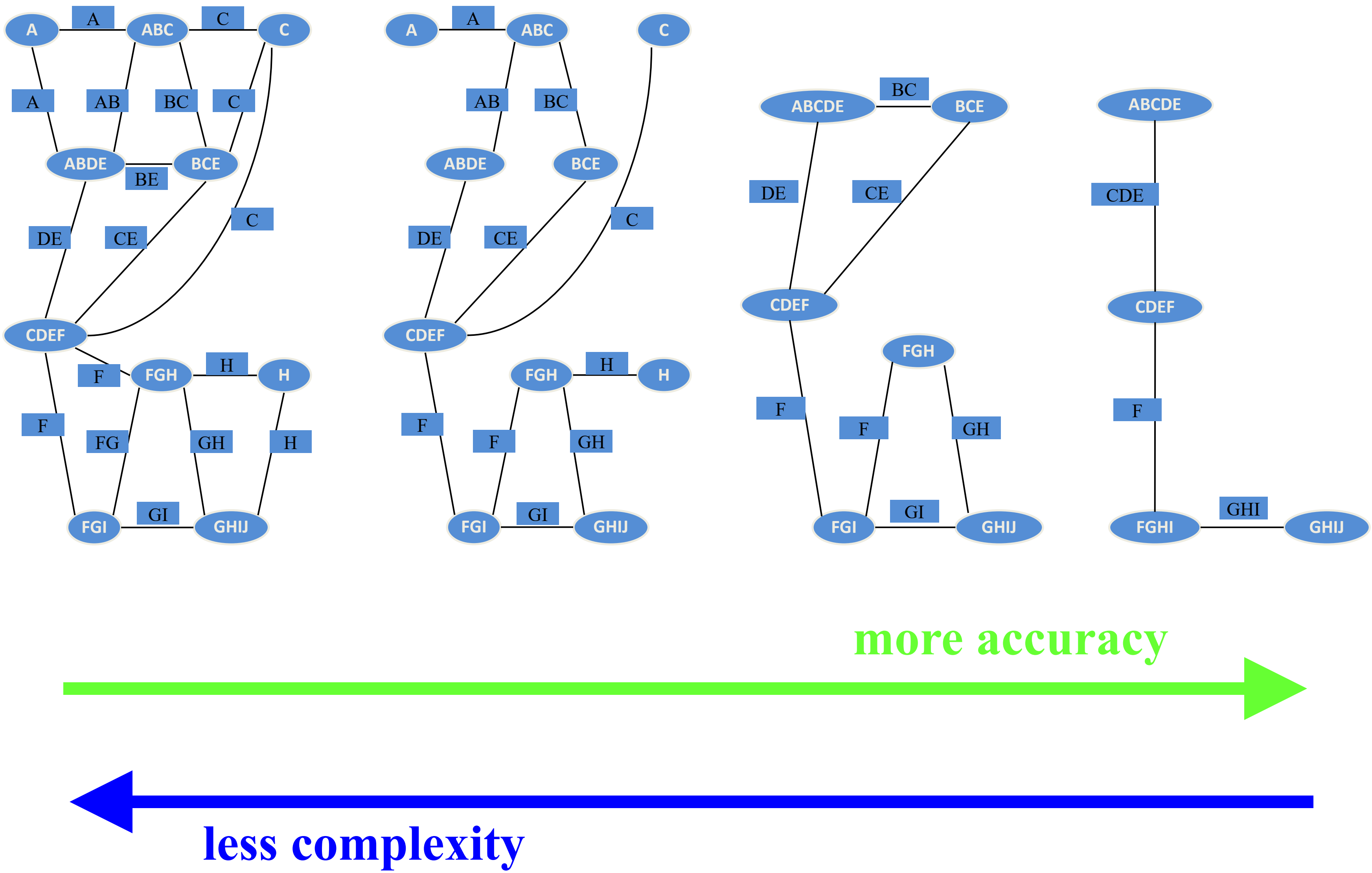
Exact Algorithms for MPE and MAP

- Exact Algorithms (MPE)
  - Bucket elimination (Replace sum by max)
  - DFS search
  - Branch and Bound Search
    - Lower bounds computed using **Mini-buckets**
- Exact Algorithms (MAP)
  - Constrained Bucket elimination (sum then max)
  - Branch and Bound search

Approximate Inference

- Propagation-based Inference
  - Belief Propagation
  - Iterative Join Graph Propagation
    - Constructing arc-minimal join graphs
    - Convergence
- Sampling-based Algorithms
  - Importance Sampling
    - Likelihood weighting
  - Metropolis Hastings
  - Gibbs sampling

Join-graphs


A
ABC
C
ABDE
BCE
CDEF
FGH
H
FGI
GHIJ
AB
BC
BE
DE
CE
F
FG
GH
GI
ABCDE
CDE
FGHI
GHI
more accuracy
less complexity

Approximate Inference for MPE/MAP

- Branch and Bound algorithm
- Local search
- Max-product Belief Propagation (did not cover)

Inference in Dynamic Probabilistic models

- Forward-Backwards algorithm
  - Slice by Slice Variable elimination (forward pass)
- Viterbi algorithm
  - MPE-type inference
- Slice by Slice Likelihood weighting
- Particle Filtering

Learning Graphical models

- Maximum Likelihood vs Bayesian approach
- Fully observable vs Partially Observable data
- Structure vs Parameter Learning
- Bayesian vs Markov networks

Learning Concepts

- Maximizing likelihood will decrease the KL divergence between the learned model and data-generating distribution
- Overfitting
- Generalization
- Bias-Variance tradeoff
- Regularization
- Training vs Test set
- K-fold Cross validation

Learning Bayesian networks Maximum likelihood approach

- Parameter learning
  - FOD: easy (ratio of counts)
  - POD case is tricky. Requires inference
    - EM and Gradient Ascent.
    - Variations
- Structure learning
  - FOD: for trees is easy (Chow-Liu algorithm)
  - FOD: for general Bayesian networks is hard
    - Need to add a penalty term. Why?
    - Local Search
  - POD: Structural EM (not covered)

Learning Bayesian networks Bayesian approach

- Bayesians: They integrate prior knowledge into the learning process and reduce learning to a problem of inference.
- Concept of the meta-network
- Discrete vs Dirichlet priors
- Parameter learning
  - FOD case: Closed form equations in which we need not explicitly construct the meta-network
  - POD case: EM algorithm (again we need not explicitly construct the meta-network. It requires inference however)
- Bayesian Structure learning (not covered in detail)

Learning Markov networks

- Hard and complicated because we have to compute the partition function which requires inference.
  - Even FOD case does not have a closed form.
- Structure learning is relatively easier because we do not have to worry about cycles

Software Resources

- BNT (Kevin Murphy)
- Alchemy (See my webpage)
- Vibhav Gogate's software page
- Adnan Darwiche's group software (http://reasoning.cs.ucla.edu/)
- Rina Dechter's software page (graphmod.ics.uci.edu)
- JavaBayes
- Hugin (commercial software)
- PNL (intel's library)
- Joris Mooij's libdai (http://cs.ru.nl/~jorism/)
- Blog (Brian Milch's statistical relational learning library)
- Smile Genie (http://genie.sis.pitt.edu/)
- FastInf
  - (http://compbio.cs.huji.ac.il/FastInf/fastInf/FastInf_Homepage.html)