베이즈망
Bayesian Networks

한국정보과학회 인공지능소사이어티

제10회 패턴인식 및 기계학습 겨울학교

딥러닝 기초, 이론 및 응용

2016년 1월 21일 (목) 16:00-18:00

숭실대학교 기계학습연구실

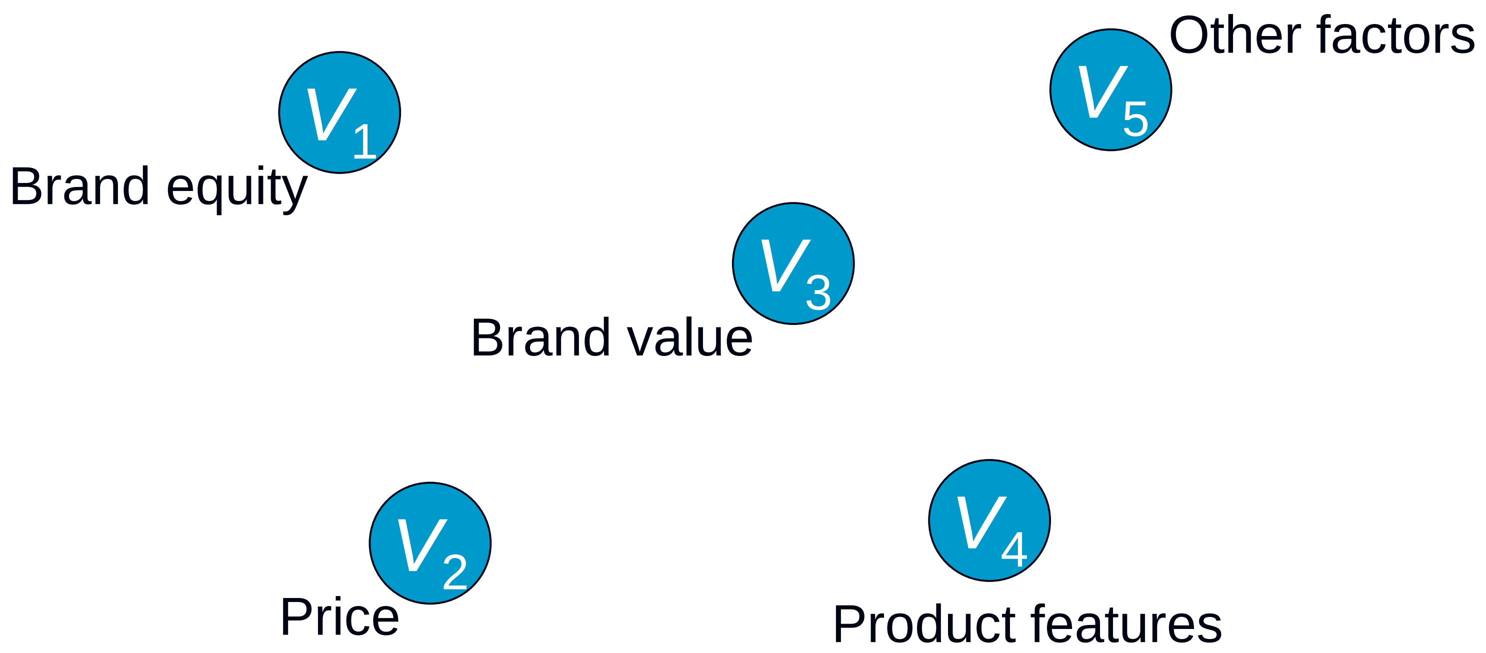
황규백

kbhwang@ssu.ac.kr; http://ml.ssu.ac.kr

- Basic Concepts of Bayesian Networks
- Inference in Bayesian Networks
- Learning Bayesian Networks
  - Parametric Learning
  - Structural Learning
- Conclusion

Our Problem Domain

- Discrete random variables



Joint Probability Distribution

- $P(V_1, V_2, V_3, V_4, V_5)$ can be represented as a table.

V_1, V_2, V_3, V_4, V_5	$P(V_1, V_2, V_3, V_4, V_5)$
Value 1	Probability 1
Value 2	Probability 2
…	…

Probabilistic Inference

- P(Brand value| Brand equity, Price)?

$$P(V_3 | V_1, V_2) = \frac{P(V_1, V_2, V_3)}{P(V_1, V_2)}$$

Marginalization

$$= \frac{\sum_{V_4, V_5} P(V_1, V_2, V_3, V_4, V_5)}{\sum_{V_3, V_4, V_5} P(V_1, V_2, V_3, V_4, V_5)}$$

- Any conditional probabilities can be calculated in principle.
  - Exponential time complexity

It is too expensive to store all joint probabilities.

- Probability table size is exponential to the number of variables.
  - If all variables are binary, the table size amounts to $(2^n – 1)$ where *n* is the number of variables.
  - Space and time complexity for storing probabilities and marginalization is formidable in practice.
- Probabilistic independence can facilitate the use of joint probability distribution.

In the Extreme Case

- Let us assume that all variables are independent from one another.

by chain rule

$$P(V_1, V_2, V_3, V_4, V_5)$$
$$= P(V_1) \cdot P(V_2 | V_1) \cdot P(V_3 | V_1, V_2) \cdot P(V_4 | V_1, V_2, V_3) \cdot P(V_5 | V_1, V_2, V_3, V_4)$$
$$= \boxed{P(V_1) \cdot P(V_2) \cdot P(V_3) \cdot P(V_4) \cdot P(V_5)}$$

- Table size comparison in binary case

# of variables	*1*	*2*	*3*	*4*	*5*	*6*
n	**1**	**2**	**3**	**4**	**5**	**6**
$2^n - 1$	**1**	**3**	**7**	**15**	**31**	**63**

Between the Two Extremes

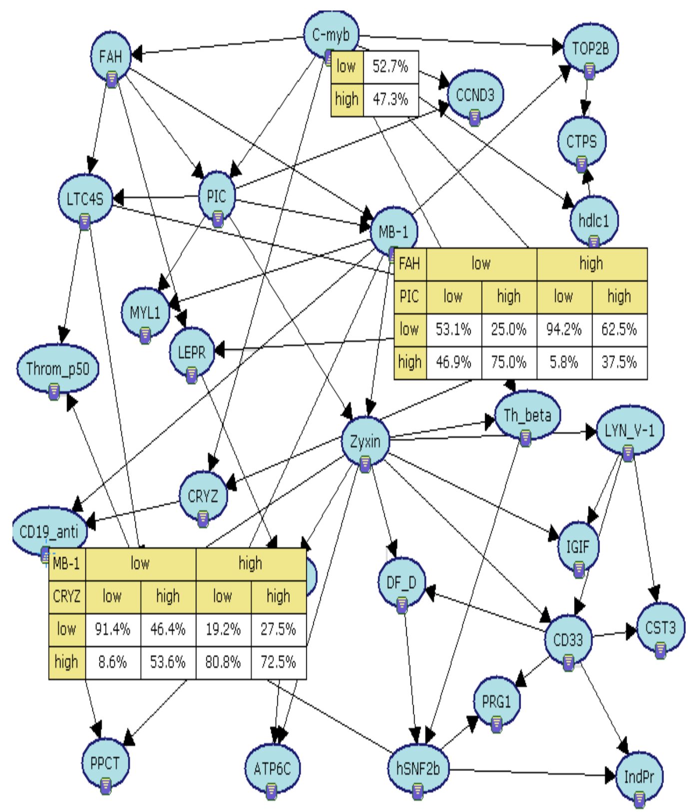
- Too complicated
  - All variables are dependent on each other.
- Too simple
  - All variables are independent from each other.
- A reasonable compromise
  - Some variables are dependent on other variables.
    - *Conditional independence*

Conditional Independence

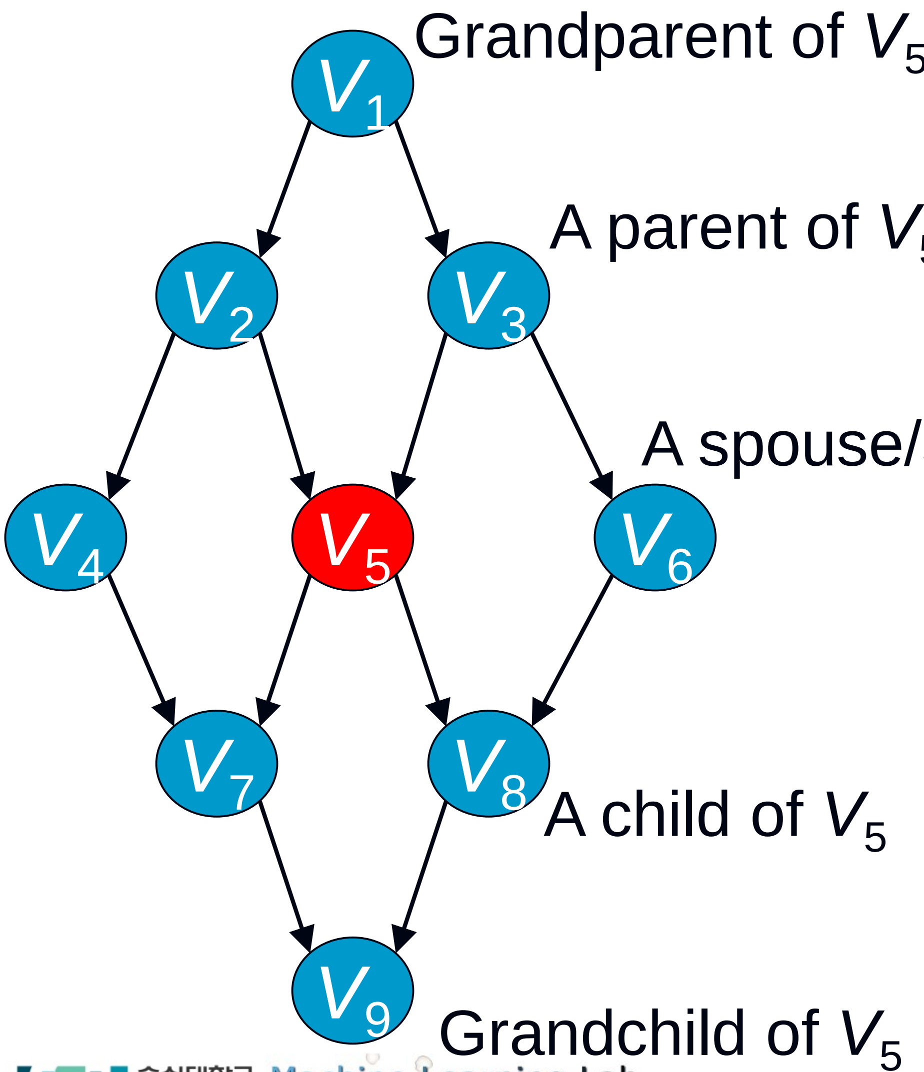
- Probabilistic independence
  - X and Y are independent from each other.
  - $P(X, Y) = P(X) \cdot P(Y)$
- Conditional (probabilistic) independence
  - X and Y are conditionally independent from each other given the value of Z.
  - $P(X, Y|Z) = P(X|Z) \cdot P(Y|Z)$
- How to describe dependencies among variables efficiently?

The Bayesian Network

- Compact representation of joint probability distribution
- Qualitative part: graph theory
  - Directed acyclic graph (DAG)
  - Vertices (nodes): variables
  - Edges: dependency or influence of a variable on another.
- Quantitative part: probability theory
  - Set of (conditional) probabilities for all variables
- Naturally handles the problem of complexity and uncertainty.



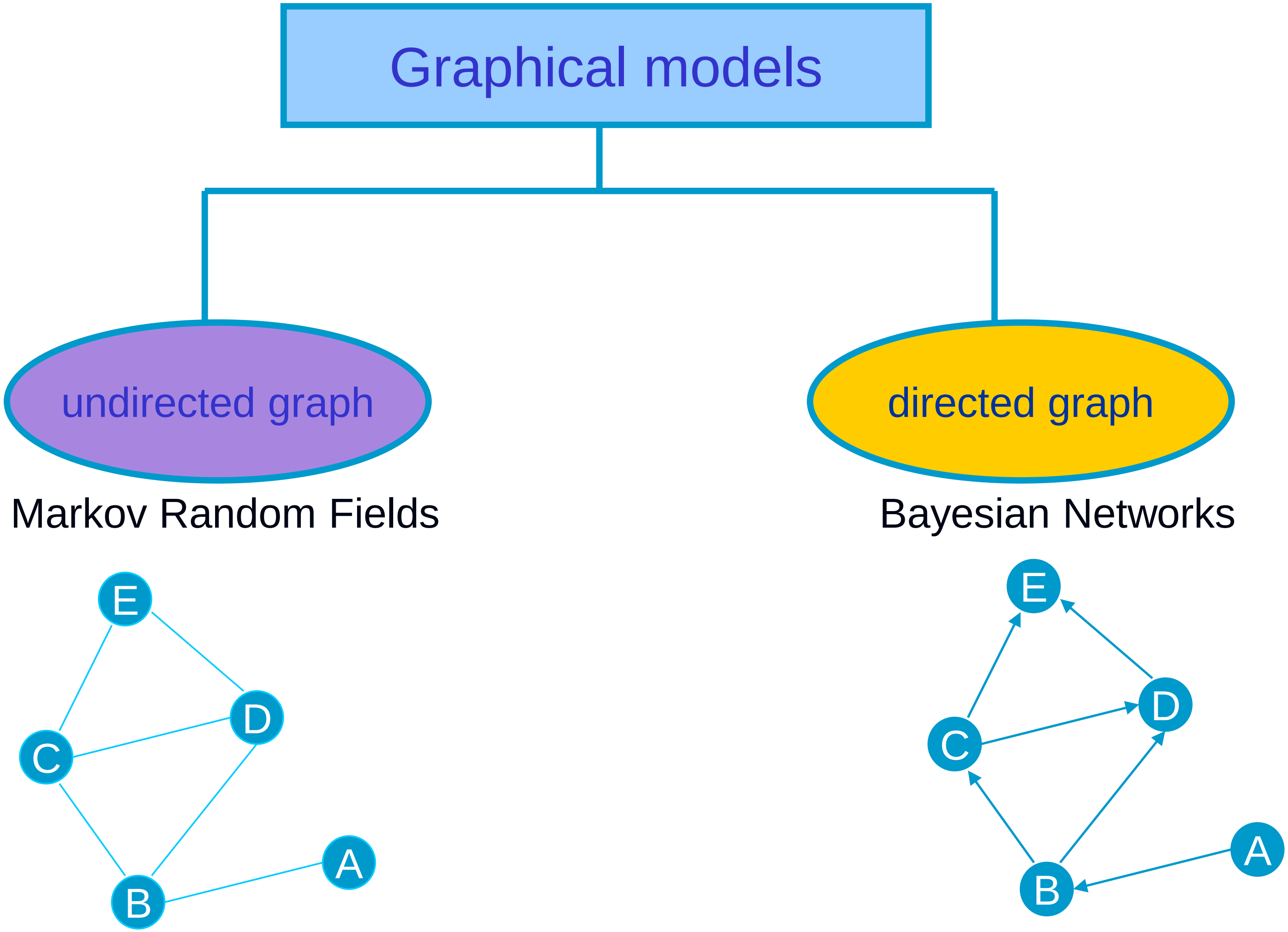
Directed Acyclic Graph Structures



A partial topological order over the nodes could be specified:

V1, [V3, V2], [V6, V5, V4], [V8, V7], V9.

Probabilistic Graphical Models

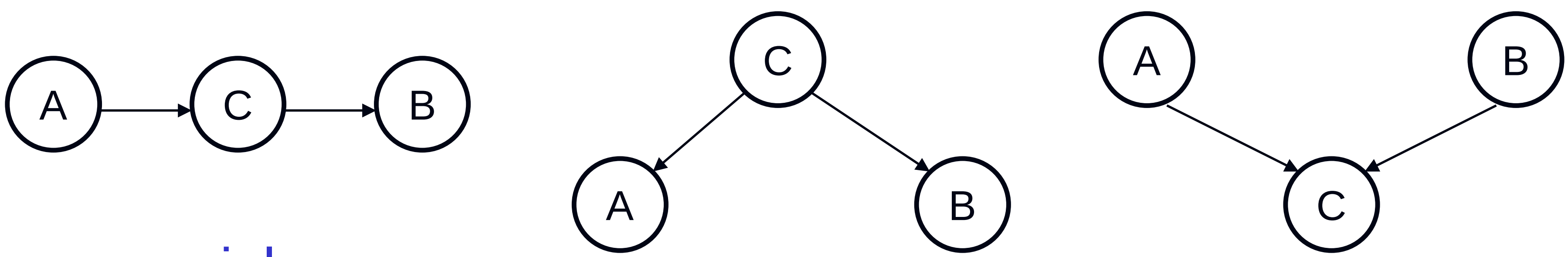


DAG for Encoding Conditional Independencies

d-separation:

- Two nodes (variables) in a DAG are *d-separated* if for all paths between them, there is an intermediate node *C* such that,
  - the connection is "*serial*" or "*diverging*" and the state of *C* is known or
  - the connection is "*converging*" and neither *C* nor any of *C*'s descendants have received evidence.

Connections in DAGs

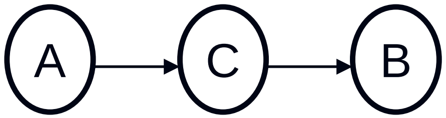


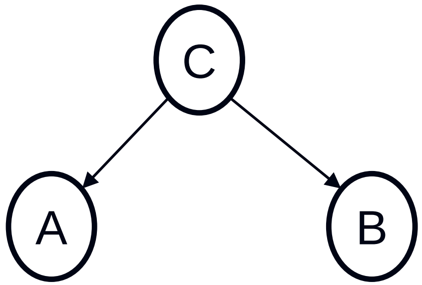
serial diverging converging

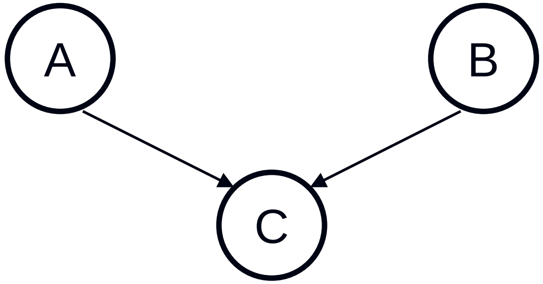
d-Separation and Conditional Independence

- Two random variables are conditionally independent from each other if the corresponding vertices in the DAG are *d*-separated.

d-Separation Example 1

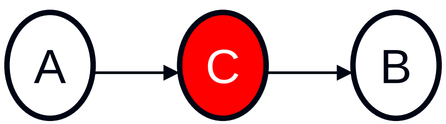


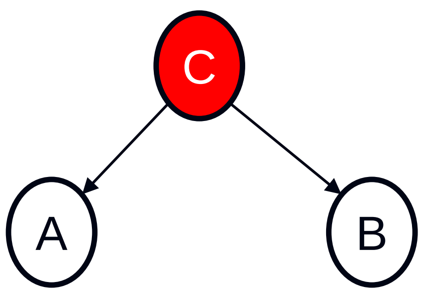


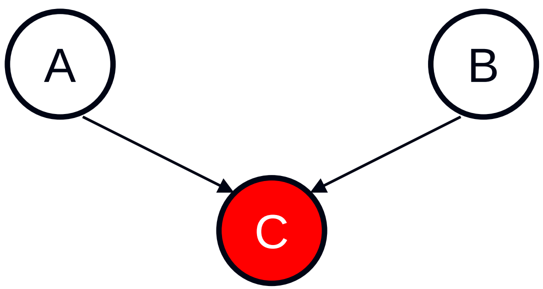


A and B is *marginally* dependent.

A and B is *marginally* independent.



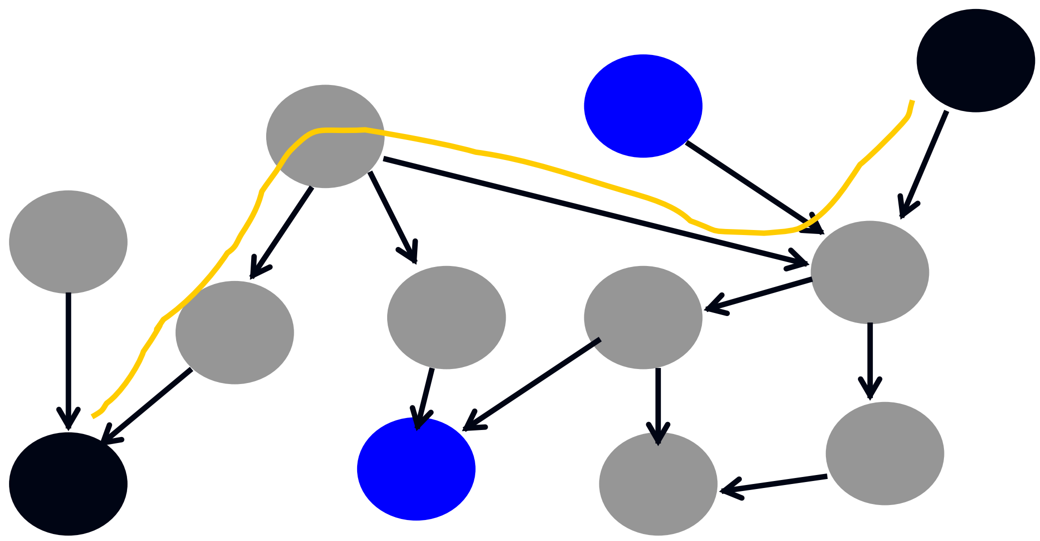




A and B is *conditionally* independent.

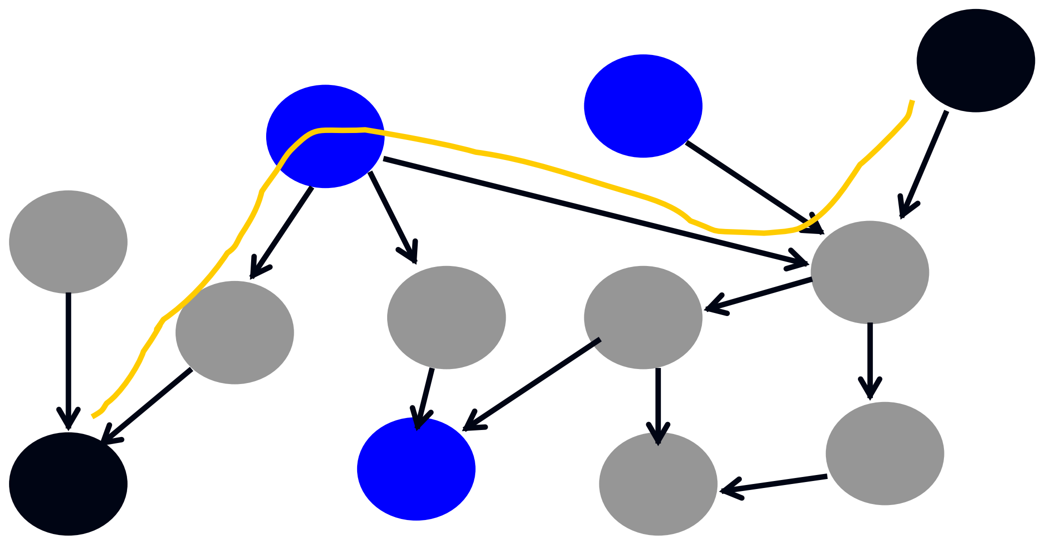
A and B is *conditionally* dependent.

d-Separation Example 2



There exists a non-blocked path. Hence, two black nodes (variables) are not *d*-separated and dependent on each other.

d-Separation Example 2

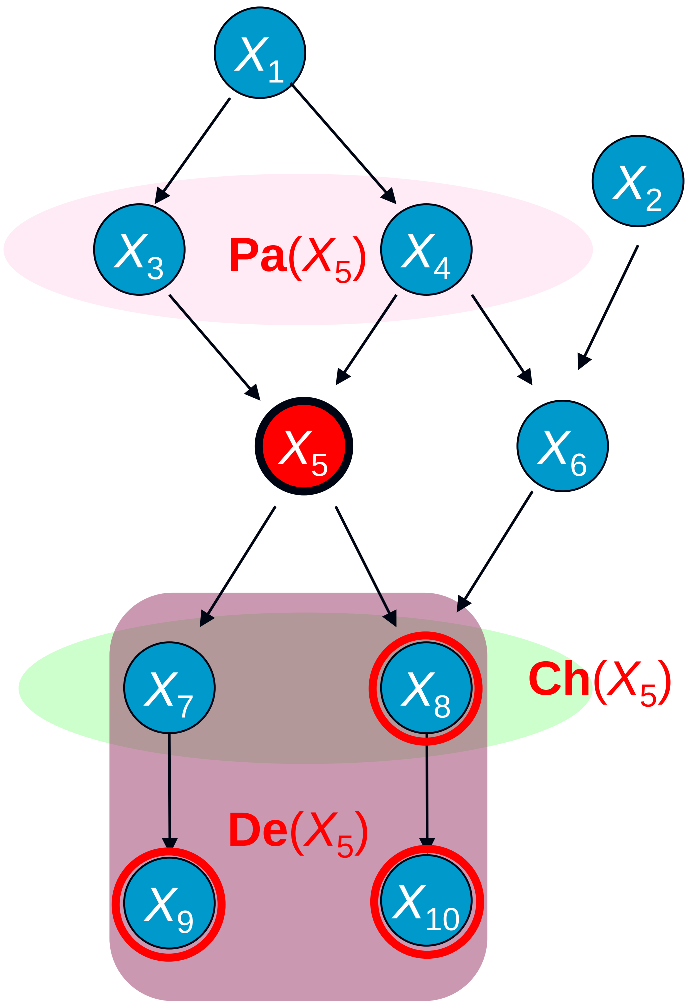


Every path is blocked now. Hence, the two black nodes (variables) are *d*-separated and independent from each other.

Definition: Bayesian Networks

- The Bayesian network consists of the following.
  - A set of n variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ and a set of directed edges between the variables (vertices).
  - The variables with the directed edges form a directed acyclic graph (DAG) structure.
    - Directed cycles are not modeled.
  - To each variable X_i and its parents $\mathbf{Pa}(X_i)$, there is attached a conditional probability table for $P(X_i|\mathbf{Pa}(X_i))$.
    - Modeling for continuous variables is also possible.

$\mathbf{X}=\{X_1, X_2, \ldots, X_{10}\}$



Pa(X_5): the parents of X_5

Ch(X_5): the children of X_5

De(X_5): the descendents of X_5

Topological sort of $X_i \in \mathbf{X}$

→ $X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}$

Chain rule in a reverse order

$$P(\mathbf{X}\setminus\{X_{10}\}, X_{10}) = P(\mathbf{X}\setminus\{X_{10}\})P(X_{10} \mid \mathbf{X}\setminus\{X_{10}\})$$
$$= P(\mathbf{X}\setminus\{X_{10}\})\boxed{P(X_{10} \mid X_8)}$$

$$P(\mathbf{X}'\setminus\{X_9\}, X_9) = P(\mathbf{X}'\setminus\{X_9\})P(X_9 \mid \mathbf{X}'\setminus\{X_9\})$$
$$= P(\mathbf{X}'\setminus\{X_9\})\boxed{P(X_9 \mid X_7)}$$

$$P(\mathbf{X}''\setminus\{X_8\}, X_8) = P(\mathbf{X}''\setminus\{X_8\})P(X_8 \mid \mathbf{X}''\setminus\{X_8\})$$
$$= P(\mathbf{X}''\setminus\{X_8\})\boxed{P(X_8 \mid X_5, X_6)}$$

→ $$P(X) = P(X_1, \ldots X_{10}) = \prod_i P(X_i \mid \mathrm{Pa}(X_i))$$

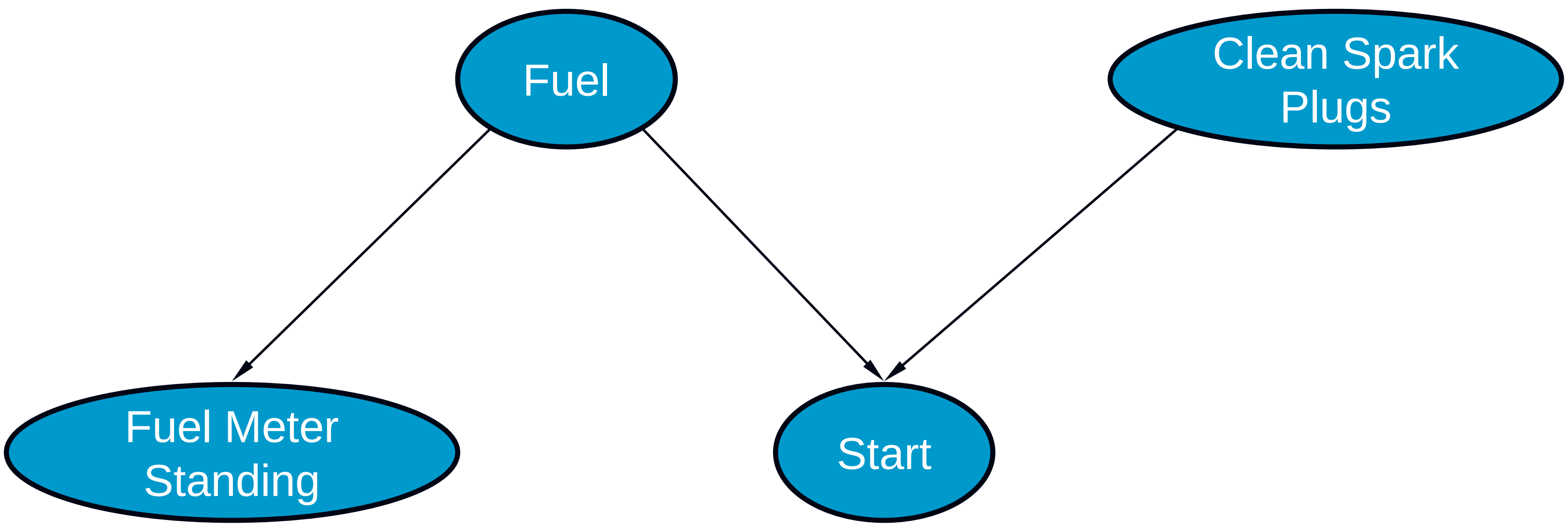
Bayesian Networks Represent Joint Probability Distribution

- By the *d*-separation property, the Bayesian network over n variables $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ represents $P(\mathbf{X})$ as follows:

$$\boxed{P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P(X_i \mid \mathbf{Pa}(X_i)).}$$

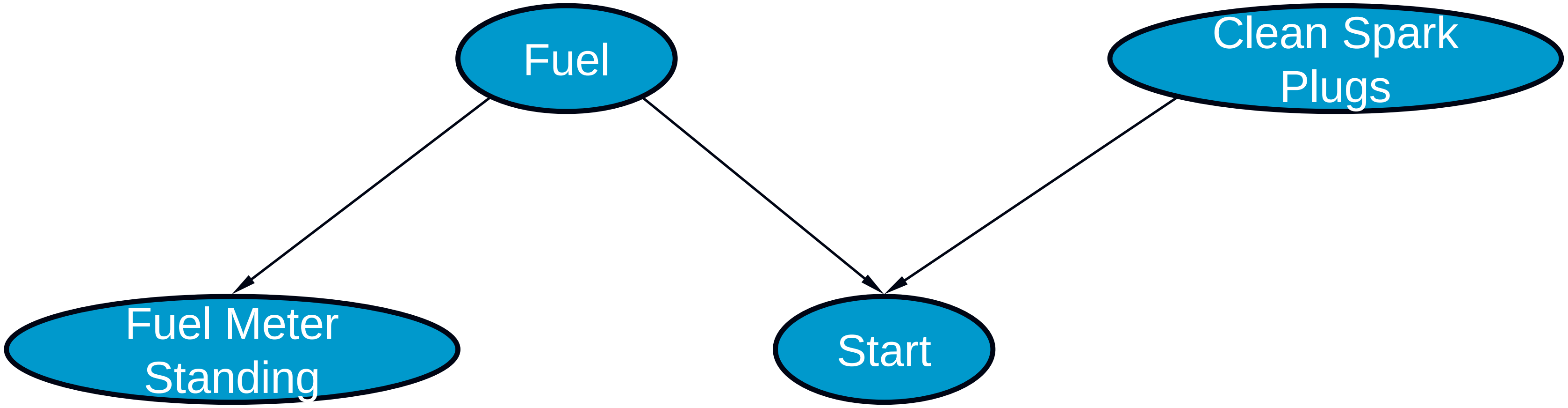
Causal Networks

- Node: event
- Arc: causal relationship between the two nodes
  - *A* → *B*: *A* causes *B*.
- Causal network for the car start problem (Jensen and Nielson, 2007)



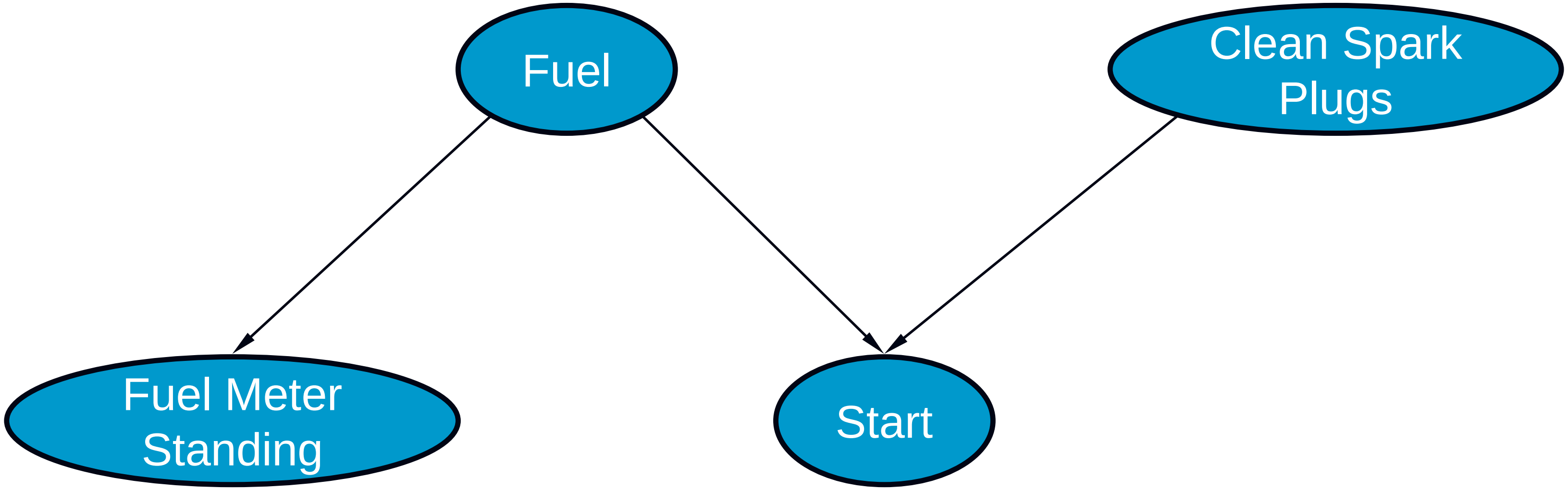
d-separation: Car Start Problem

1. ‘Start’ and ‘Fuel’ are dependent on each other.
2. ‘Start’ and ‘Clean Spark Plugs’ are dependent on each other.
3. ‘Fuel’ and ‘Fuel Meter Standing’ are dependent on each other.
4. ‘Fuel’ and ‘Clean Spark Plugs’ are conditionally dependent on each other given the value of ‘Start’.
5. ‘Fuel Meter Standing’ and ‘Start’ are conditionally independent given the value of ‘Fuel’.

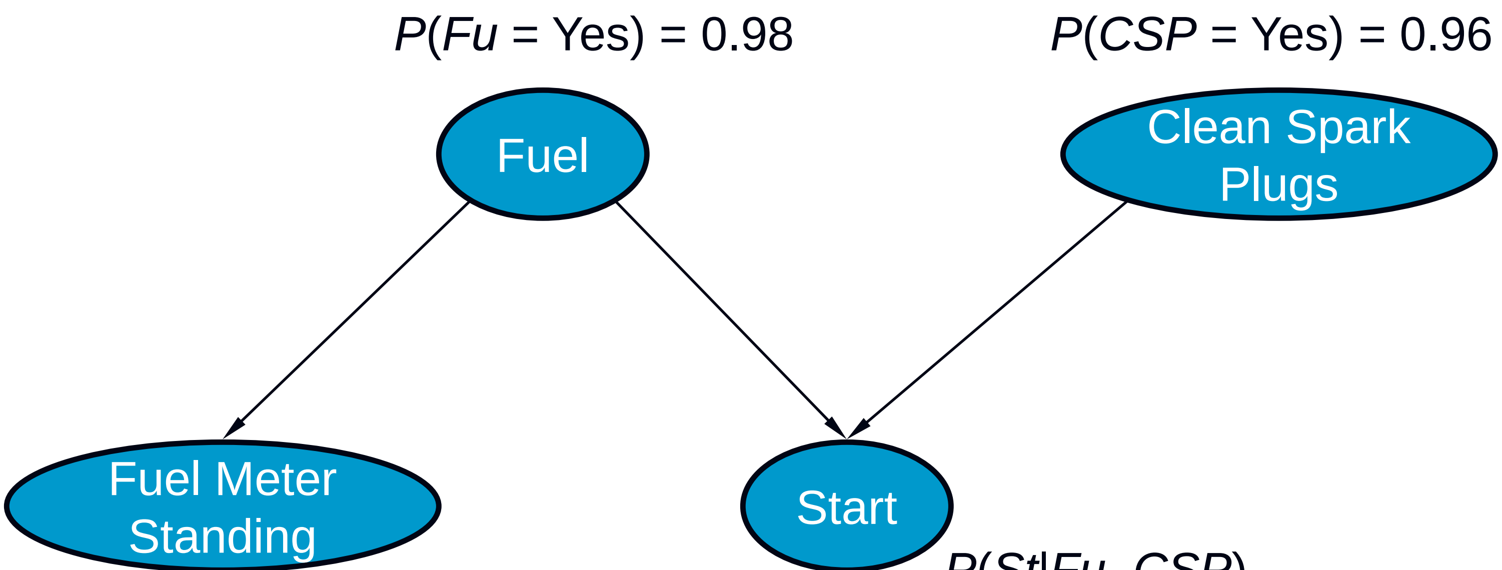


Reasoning with Causal Networks

- My car does not start. → Increases the certainty of no fuel and dirty spark plugs. → Increases the certainty of fuel meter's standing for the empty.
- Fuel meter stands for the half. → Decreases the certainty of no fuel → Increases the certainty of dirty spark plugs.



Bayesian Network for the Car Start Problem [Jensen and Nielson, 2007]



$P(FMS|Fu)$

	FMS = Full	FMS = Half	FMS = Empty
Fu = Yes	0.39	0.60	0.01
Fu = No	0.001	0.001	0.998

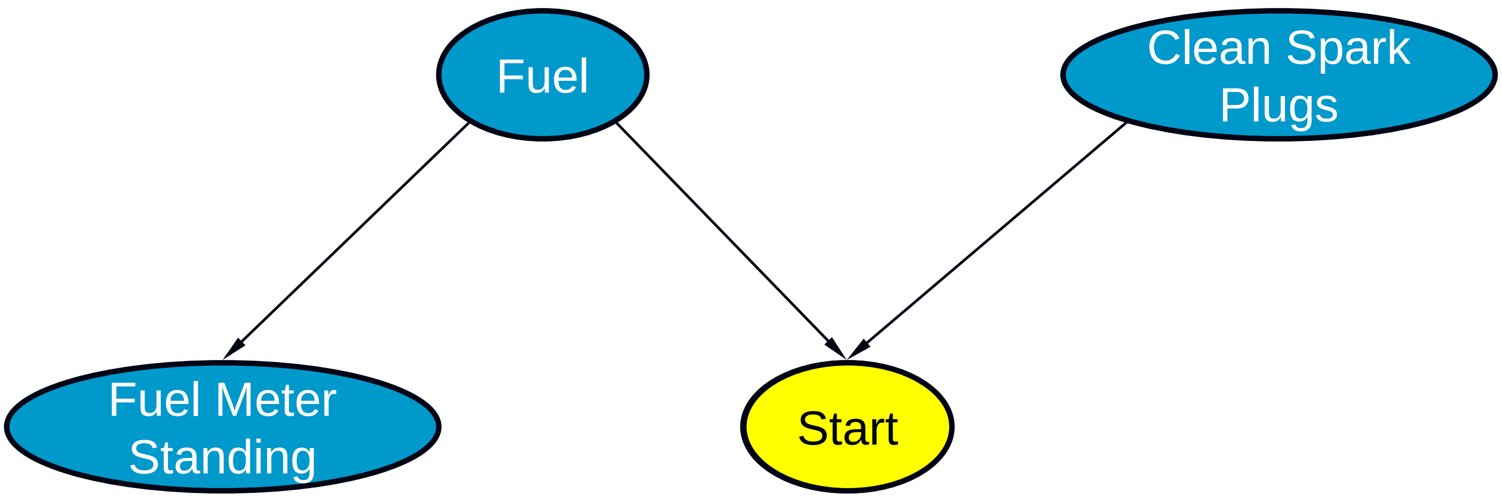
$P(St|Fu, CSP)$

(Fu, CSP)	$Start$=YES	$Start$=No
(Yes, Yes)	0.99	0.01
(Yes, No)	0.01	0.99
(No, Yes)	0	1
(No, No)	0	1

- Basic Concepts of Bayesian Networks
- Inference in Bayesian Networks
- Learning Bayesian Networks
  - Parametric Learning
  - Structural Learning
- Conclusion

Inference in Bayesian Networks

- Infer the probability of an event given some observations.



How probable is that the spark plugs are dirty?

In other words, what's the probability $P(CSP = No | St = No)$?

Inference Example

$$X_1 \rightarrow X_2 \rightarrow X_3$$

$P(X_1) = (0.6, 0.4)$

$P(X_2|X_1) =$

$X_1 == 0: (0.2, 0.8)$

$X_1 == 1: (0.5, 0.5)$

$P(X_3|X_2) =$

$X_2 == 0: (0.3, 0.7)$

$X_2 == 1: (0.7, 0.3)$

Initial State

$$X_1 \longrightarrow X_2 \longrightarrow X_3$$

$P(X_2) = \sum_{X1,\, X3} P(X_1, X_2, X_3)$

$= \sum_{X1,\, X3} P(X_1)P(X_2|X_1)P(X_3|X_2)$

$= \sum_{X1} P(X_1)P(X_2|X_1) \sum_{X3} P(X_3|X_2)$

$= \sum_{X1} P(X_1)P(X_2|X_1)$

$= 0.6 * (0.2, 0.8) + 0.4 * (0.5, 0.5)$

$= (0.12 + 0.2, 0.48 + 0.2) = (0.32, 0.68)$

$P(X_1) = (0.6, 0.4)$

$P(X_2|X_1) =$

$X_1 == 0: (0.2, 0.8)$

$X_1 == 1: (0.5, 0.5)$

$P(X_3|X_2) =$

$X_2 == 0: (0.3, 0.7)$

$X_2 == 1: (0.7, 0.3)$

Given that $X_3 == 1$

$$X_1 \rightarrow X_2 \rightarrow X_3$$

$P(X_1|X_3 = 1) = \beta\, P(X_1, X_3 = 1)$

$= \beta \sum_{X2} P(X_1, X_2, X_3 = 1)$

$= \beta \sum_{X2} P(X_1)P(X_2|X_1)P(X_3 = 1|X_2)$

$= \beta\, P(X_1) \sum_{X2} P(X_2|X_1)P(X_3 = 1|X_2)$

$= \beta\, P(X_1)\ (0.2 * 0.7 + 0.8 * 0.3,\ 0.5 * 0.7 + 0.5 * 0.3)$

$= \beta\ (0.6, 0.4) * (0.38, 0.5)$

$= \beta\ (0.228, 0.2) = (0.53, 0.47)$

$P(X_1) = (0.6, 0.4)$

$P(X_2|X_1) =$

$X_1 == 0: (0.2, 0.8)$

$X_1 == 1: (0.5, 0.5)$

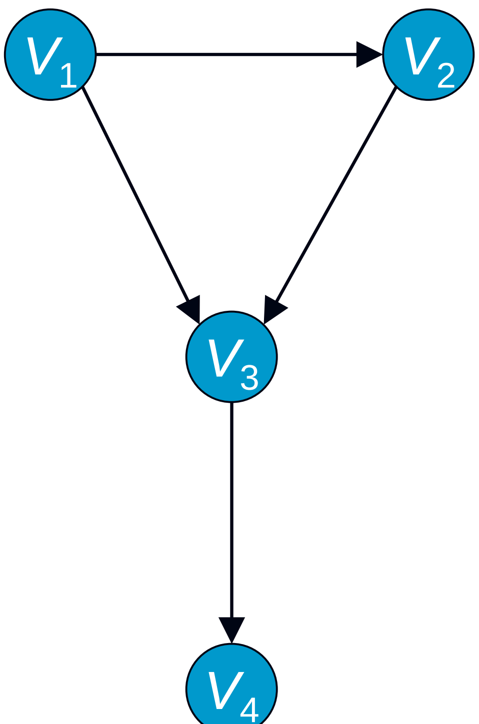
$P(X_3|X_2) =$

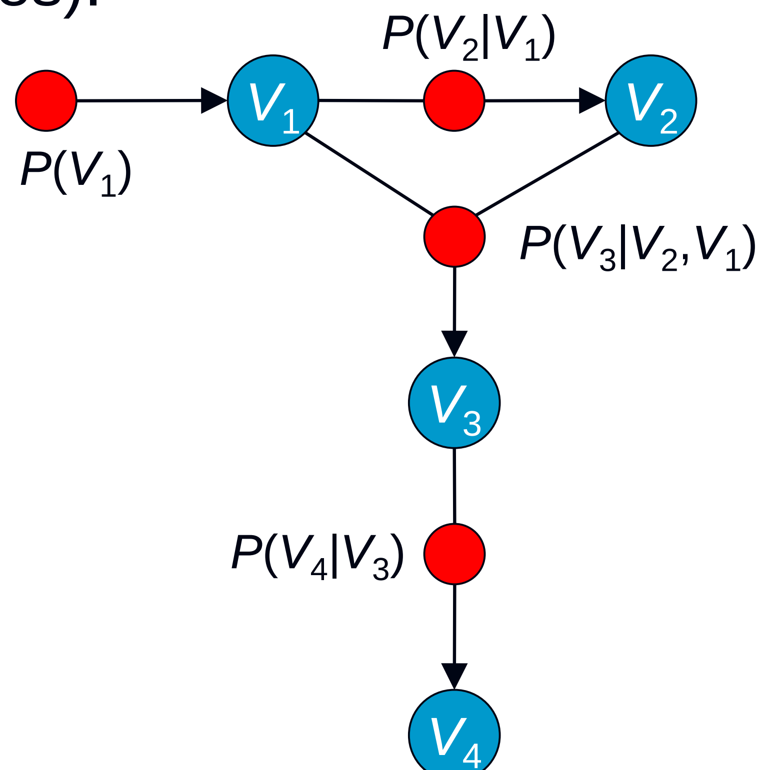
$X_2 == 0: (0.3, 0.7)$

$X_2 == 1: (0.7, 0.3)$

Factor Graph

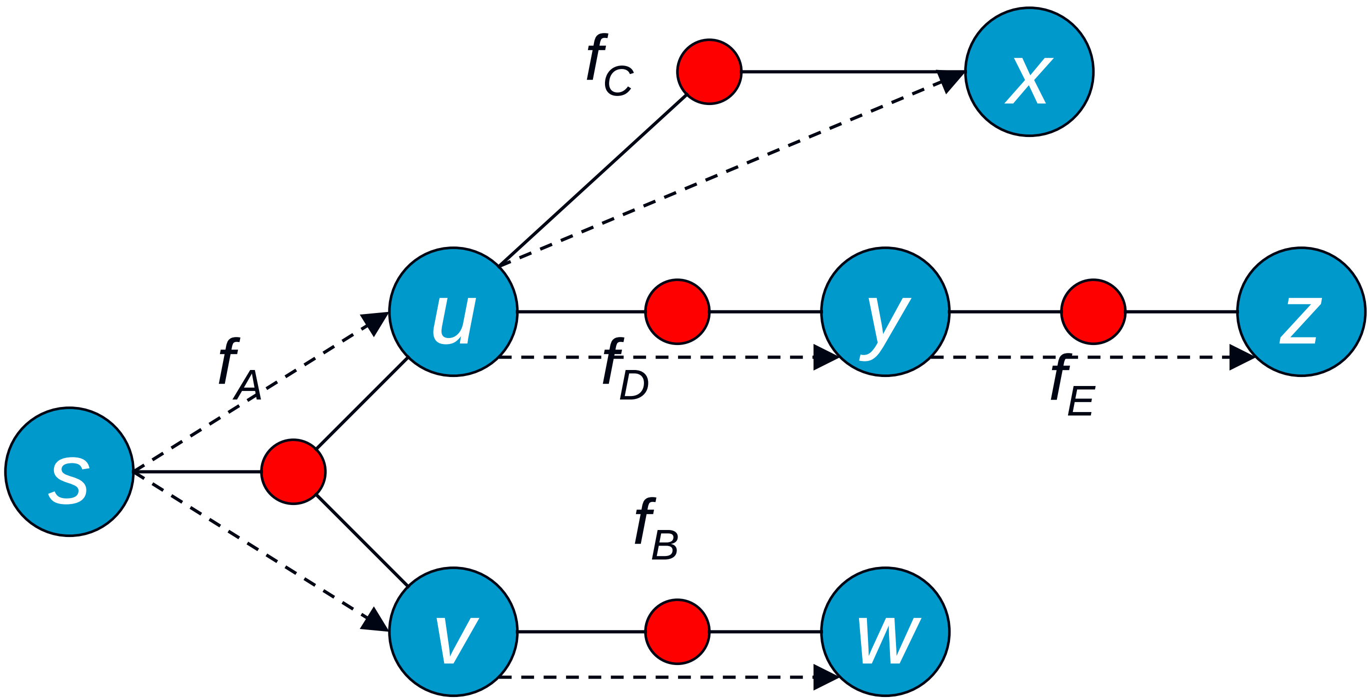
- A bipartite graph with one set of vertices corresponding to the variables in the Bayesian network and another set of vertices corresponding to the local functions (i.e., conditional probability tables).





Singly-Connected Networks

- A singly-connected network has only a single path (ignoring edge directions) connecting any two vertices.



Factorization of Global Distribution and Inference

- Example network represents the joint probability distribution as follows:

$$P(s,u,v,w,x,y,z) = f_A(s,u,v)\, f_B(v,w)\, f_C(u,x)\, f_D(u,y)\, f_E(y,z).$$

- The probability of *s* given the value of *z* is calculated as

$$P(s \mid z = z') = P(s, z = z') / \sum_s P(s, z = z'),$$

$$P(s, z = z') = \sum_{u,v,w,x,y} P(s,u,v,w,x,y,z = z'),$$

$$P(s, z = z')$$

$$= \sum_{u,v} f_A(s,u,v) \{\sum_w f_B(v,w)\} \{[\sum_x f_C(u,x)][\sum_y f_D(u,y)\, f_E(y, z = z')]\}.$$

Cost of Marginalization

- # of states of the variables:
  - n_u, n_v, n_w, n_x, n_y, n_z

$$P(s, z = z') = \sum_{u,v,w,x,y} P(s, u, v, w, x, y, z = z')$$

$$\boxed{n_u \times n_v \times n_w \times n_x \times n_y}$$

$$P(s, z = z')$$

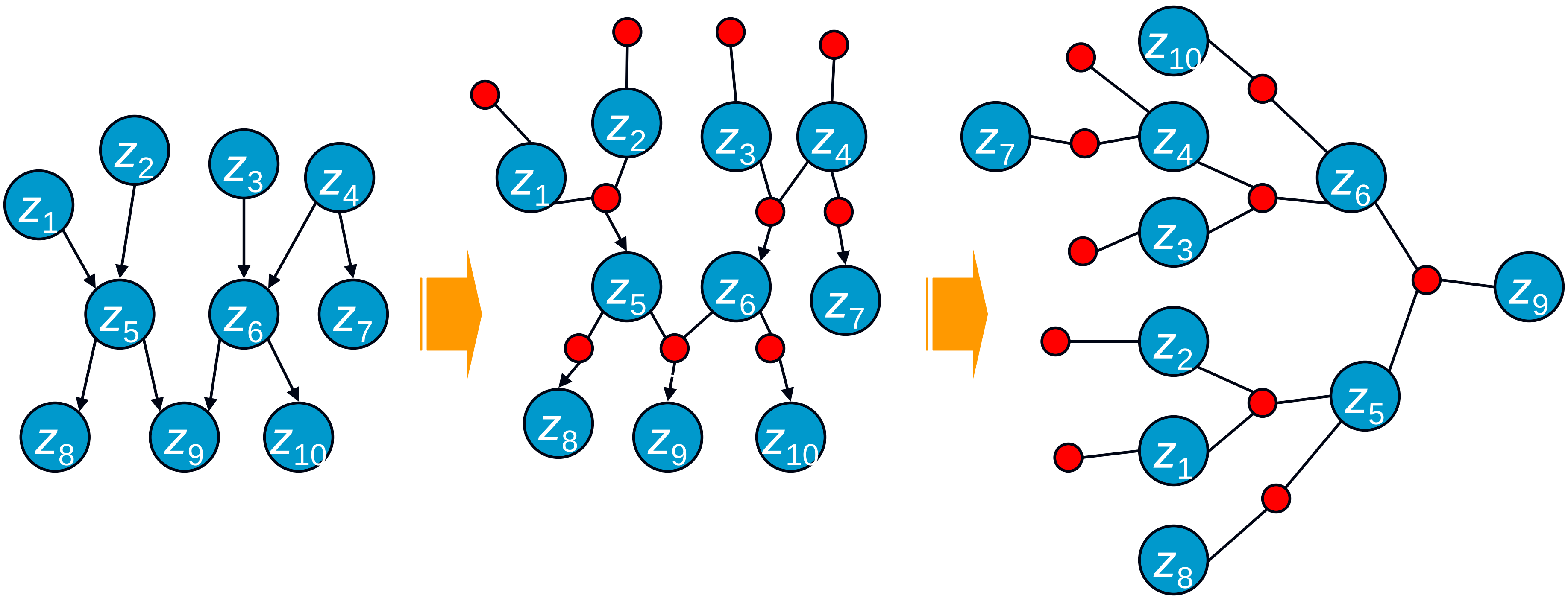
$$= \sum_{u,v} f_A(s,u,v)\{\sum_w f_B(v,w)\}\{[\sum_x f_C(u,x)][\sum_y f_D(u,y) f_E(y, z = z')]\}$$

$$\boxed{n_u \times n_v + n_w + n_x + n_y}$$

The Generalized Forward-Backward Algorithm

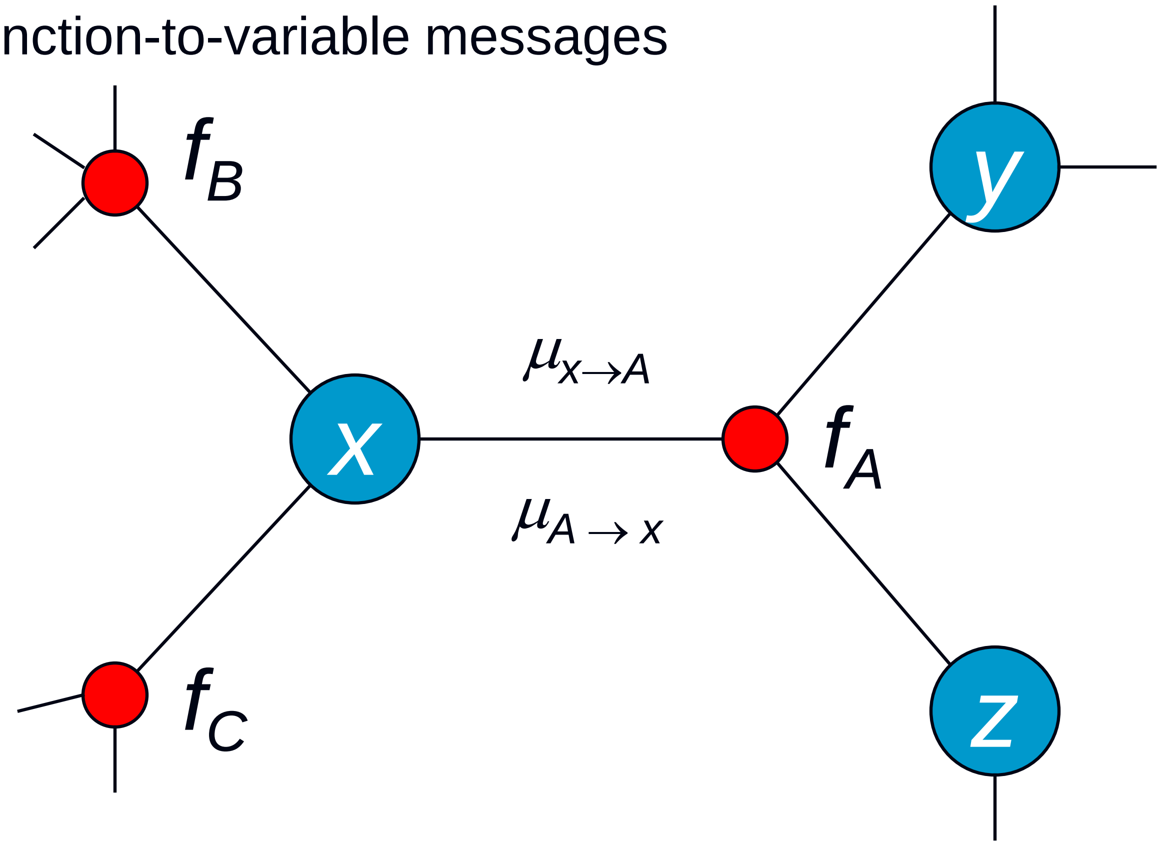
- The generalized forward-backward algorithm is one flavor of the probability propagation.
- The generalized forward-backward algorithm:
  1. Convert a Bayesian network into the factor graph.
  2. The factor graph is arranged as a horizontal tree with an arbitrary chosen “root” vertex.
  3. Beginning at the left-most level, messages are passed level by level forward to the root.
  4. Messages are passed level by level backward from root to the leaves.
- Messages represent the propagated probability through edges of the graphical model.

Convert a Bayesian Network into the Factor Graph



Message Passing in Graphical Models

- Two types of messages:
  - Variable-to-function messages
  - Function-to-variable messages



Calculation of Messages

- Variable-to-function message:
  - If *x* is unobserved, then

$$\mu_{x\to A}(x) = \mu_{B\to x}(x)\mu_{C\to x}(x).$$

  - If *x* is observed as *x*', then

$$\mu_{x\to A}(x') = 1, \quad \mu_{x\to A}(x) = 0 \quad \text{(for other values).}$$

- Function-to-variable message:

$$\mu_{A\to x}(x) = \sum_{y}\sum_{z} f_A(x,y,z)\mu_{y\to A}(y)\mu_{z\to A}(z).$$

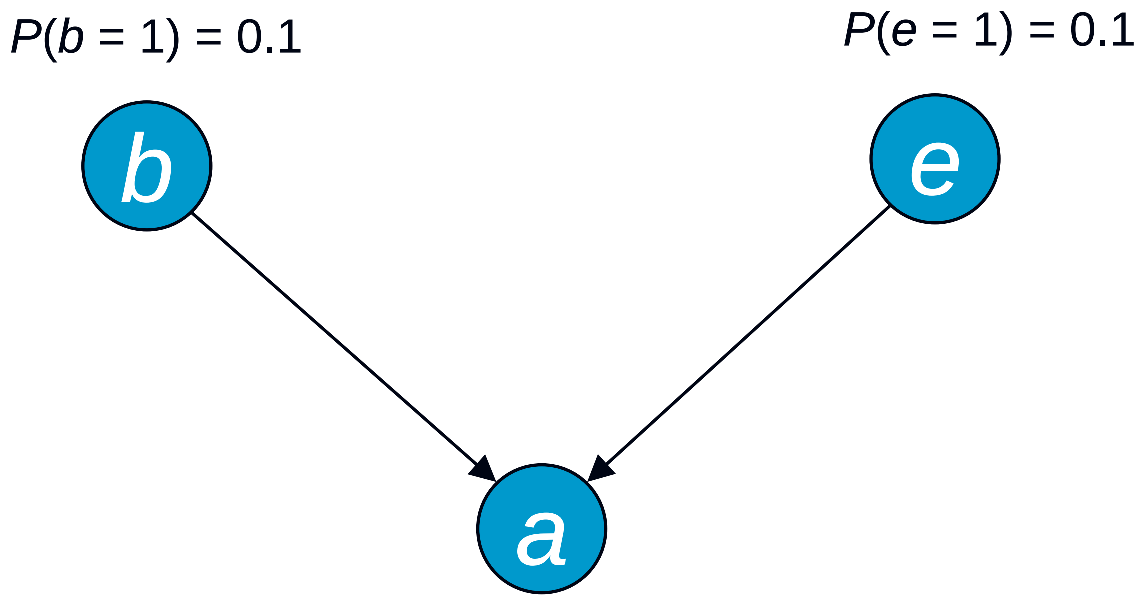
Computation of Conditional Probability

- After the generalized forward-backward algorithm ends, each edge in the factor graph has its calculated message values.
- The probability of *x* given the observations **v** is as follows:

$$P(x \mid \mathbf{v}) = \beta \mu_{A \to x}(x) \mu_{B \to x}(x) \mu_{C \to x}(x),$$

  - where β is a normalizing constant.

The Burglar Alarm Problem



(b, e)	$P(a = 0 \mid b, e)$	$P(a = 1 \mid b, e)$
(0, 0)	0.999	0.001
(0, 1)	0.865	0.135
(1, 0)	0.632	0.368
(1, 1)	0.393	0.607

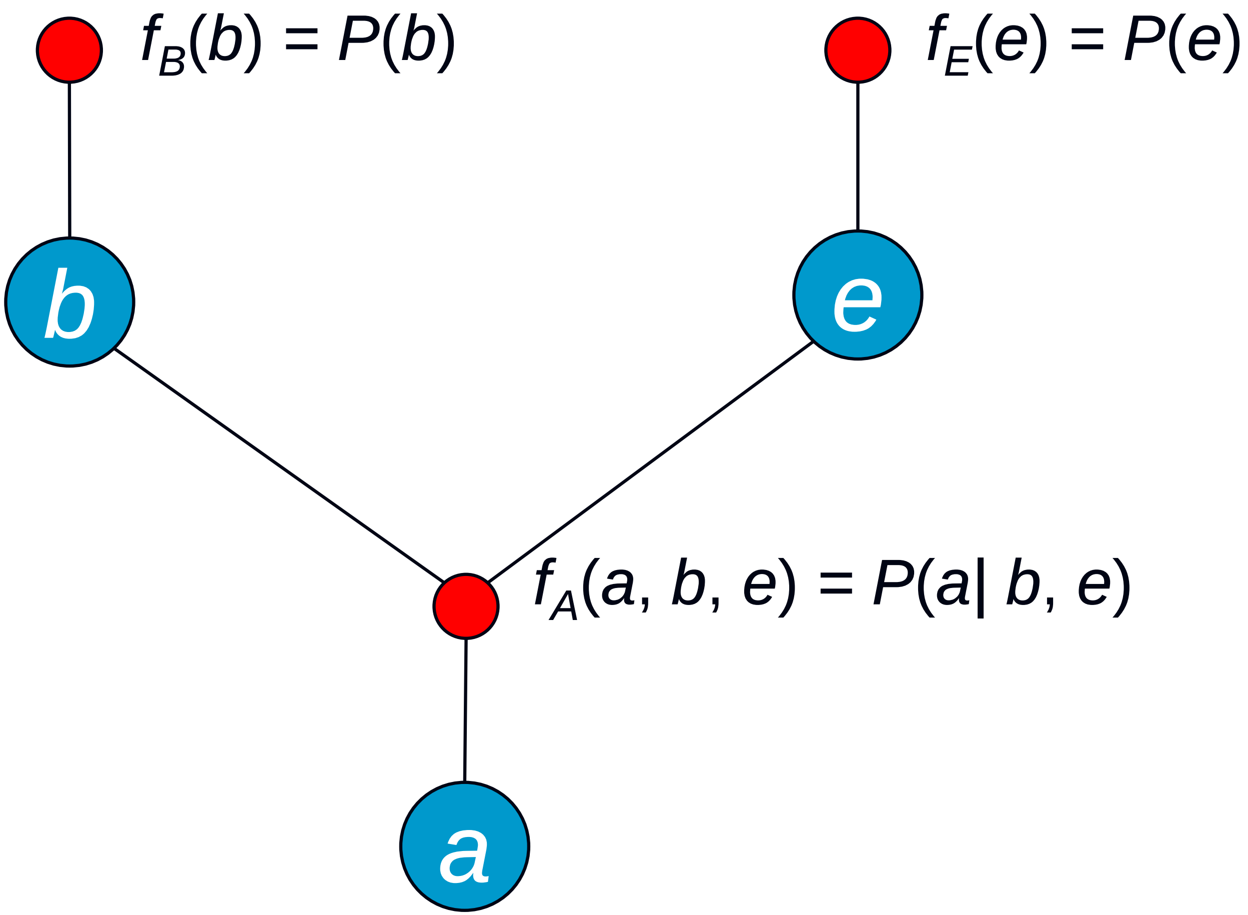
Alarm Alert

- Calculate $P(b, e|a = 1)$
- Because the network structure is simple,

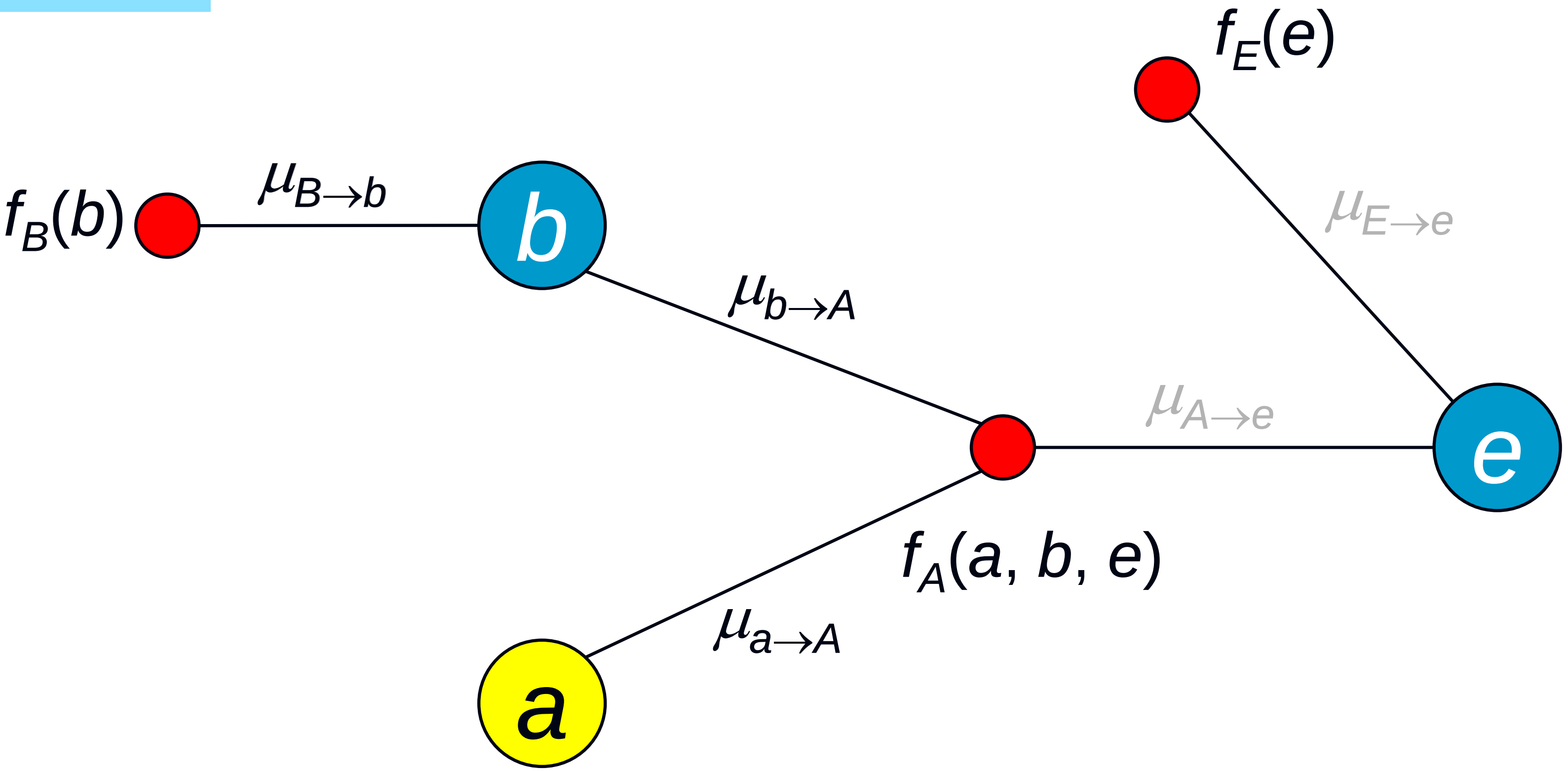
$$P(b,e\,|\,a)=\frac{P(b,e,a)}{P(a)}=\frac{P(b)P(e)P(a\,|\,b,e)}{\sum_{b',e'}P(b')P(e')P(a\,|\,b',e')}.$$

$P(b{=}0, e{=}0\,|\,a{=}1) = 0.016$
$P(b{=}0, e{=}1\,|\,a{=}1) = 0.233$
$P(b{=}1, e{=}0\,|\,a{=}1) = 0.635$
$P(b{=}1, e{=}1\,|\,a{=}1) = 0.116$

Applying the Generalized Forward-Backward Algorithm



Forward Pass I

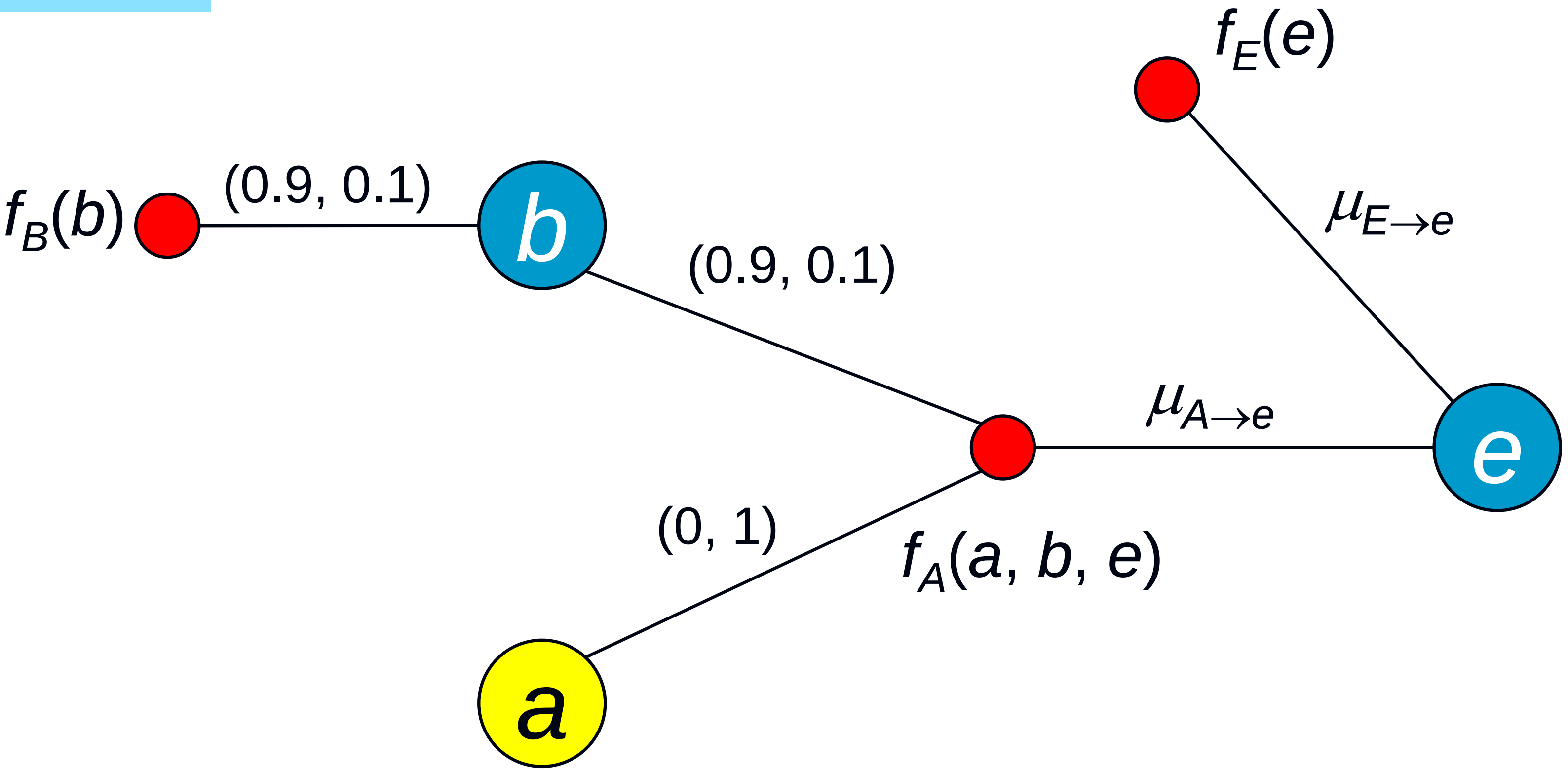


$\mu_{B\to b} = f_B(b) = (0.9, 0.1)$

$\mu_{b\to A} = \mu_{B\to b} = (0.9, 0.1)$

$\mu_{a\to A} = (0, 1)$

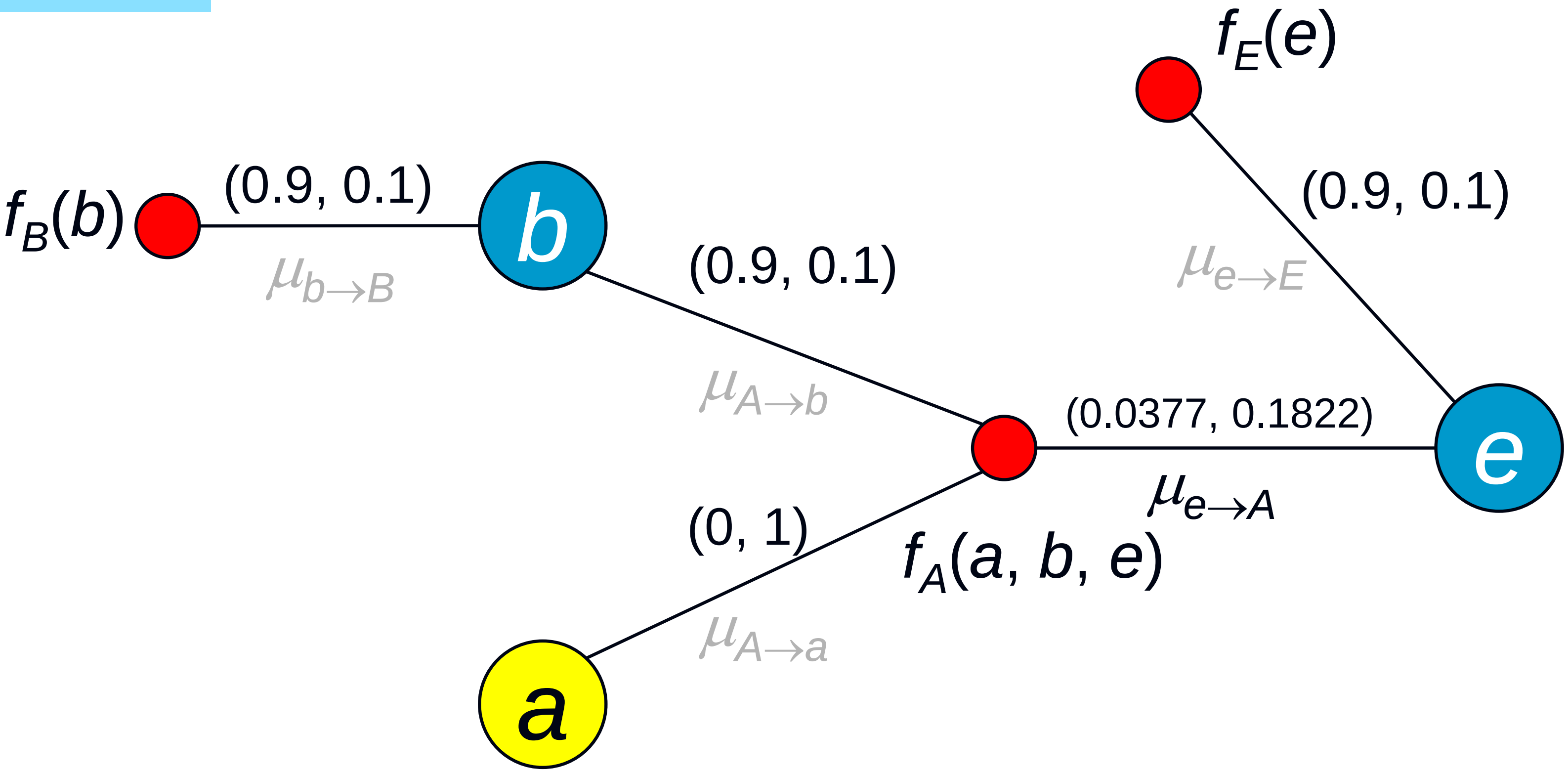
Forward Pass II



$$
\begin{aligned}
\mu_{A\to e}(e) &= \sum_{b,a} f_A(a,b,e)\mu_{b\to A}(b)\mu_{a\to A}(a) \\
&= \sum_{b} P(a=1\,|\,b,e)\mu_{b\to A}(b) \\
&= (0.001\times 0.9+0.368\times 0.1, 0.135\times 0.9+0.607\times 0.1) = (0.0377, 0.1822)
\end{aligned}
$$

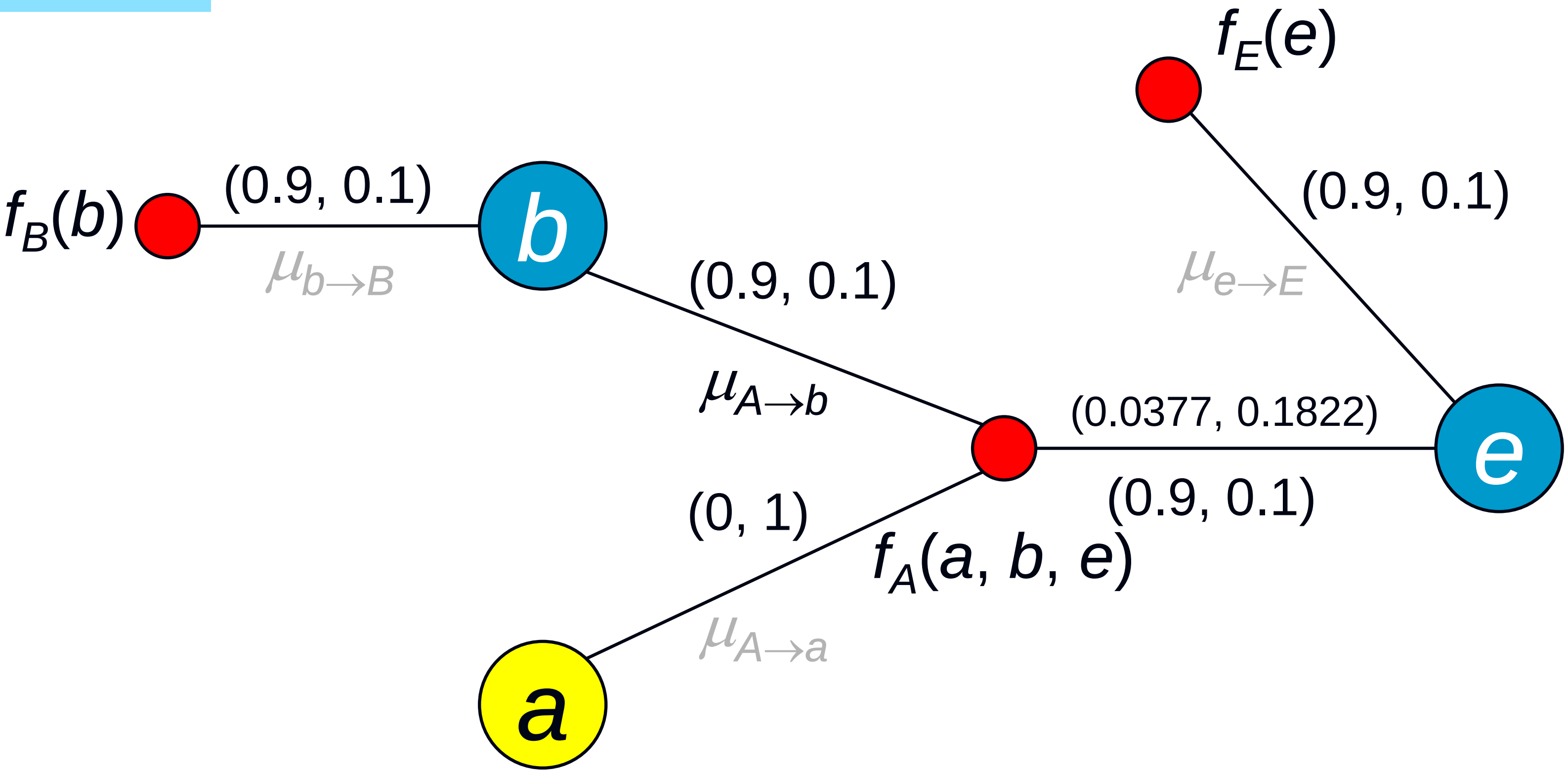
$\mu_{E\to e} = f_E(e) = (0.9, 0.1)$

Backward Pass I



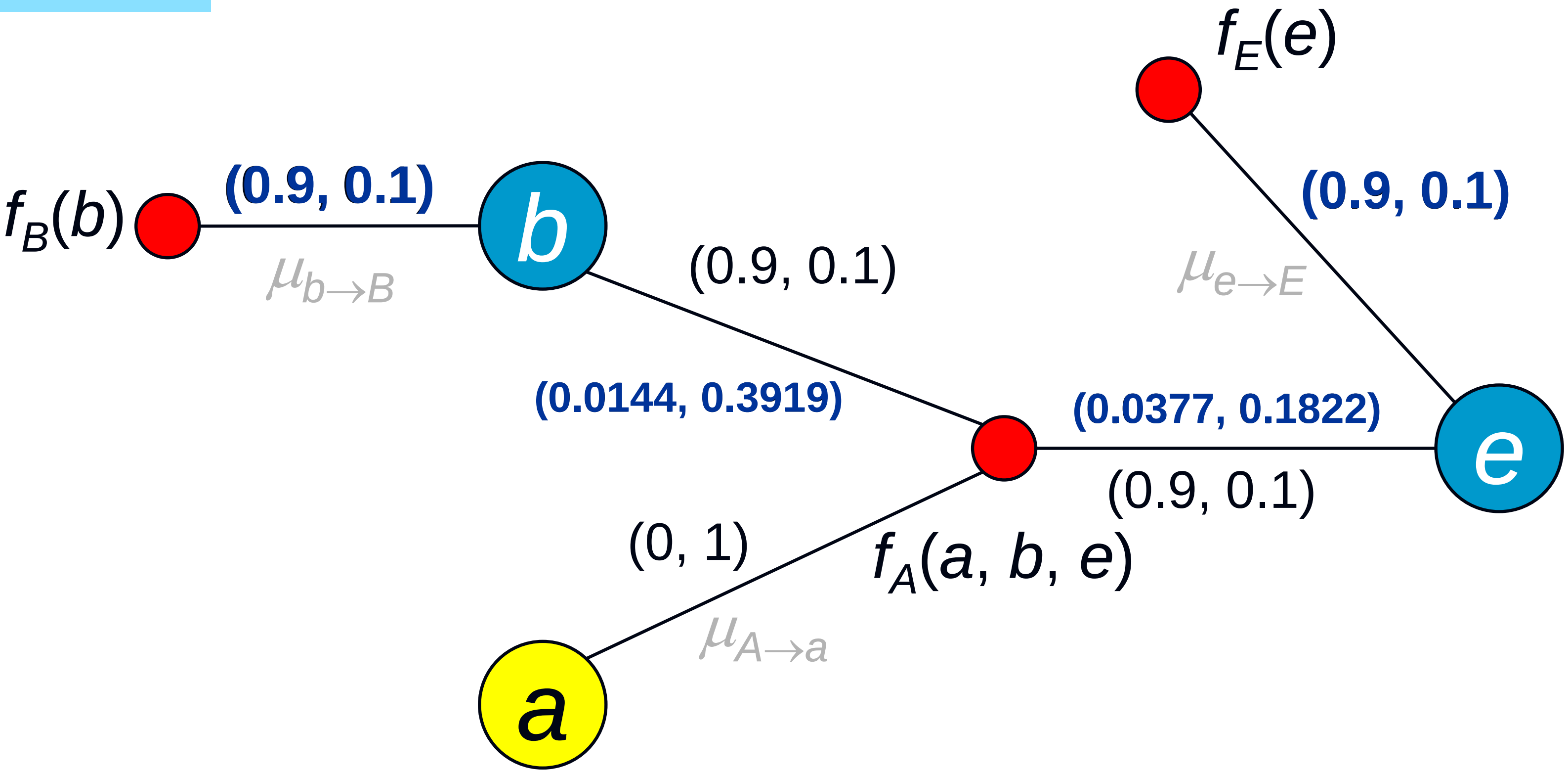
$$\mu_{e\to A} = \mu_{E\to e} = (0.9, 0.1)$$

Backward Pass II



$$\mu_{A\to b}(b) = \sum_{e,a} f_A(a,b,e)\mu_{e\to A}(b)\mu_{a\to A}(a)$$
$$= \sum_{e} P(a=1\,|\,b,e)\mu_{e\to A}(e)$$
$$= (0.001\times 0.9 + 0.135\times 0.1, 0.368\times 0.9 + 0.607\times 0.1) = (0.0144, 0.3919)$$

Finale



$$((P(b=0\,|\,a=1), P(b=1\,|\,a=1)) = \beta(\mu_{B\to b}(0)\mu_{A\to b}(0), \mu_{B\to b}(1)\mu_{A\to b}(1)) = (0.249, 0.751)$$

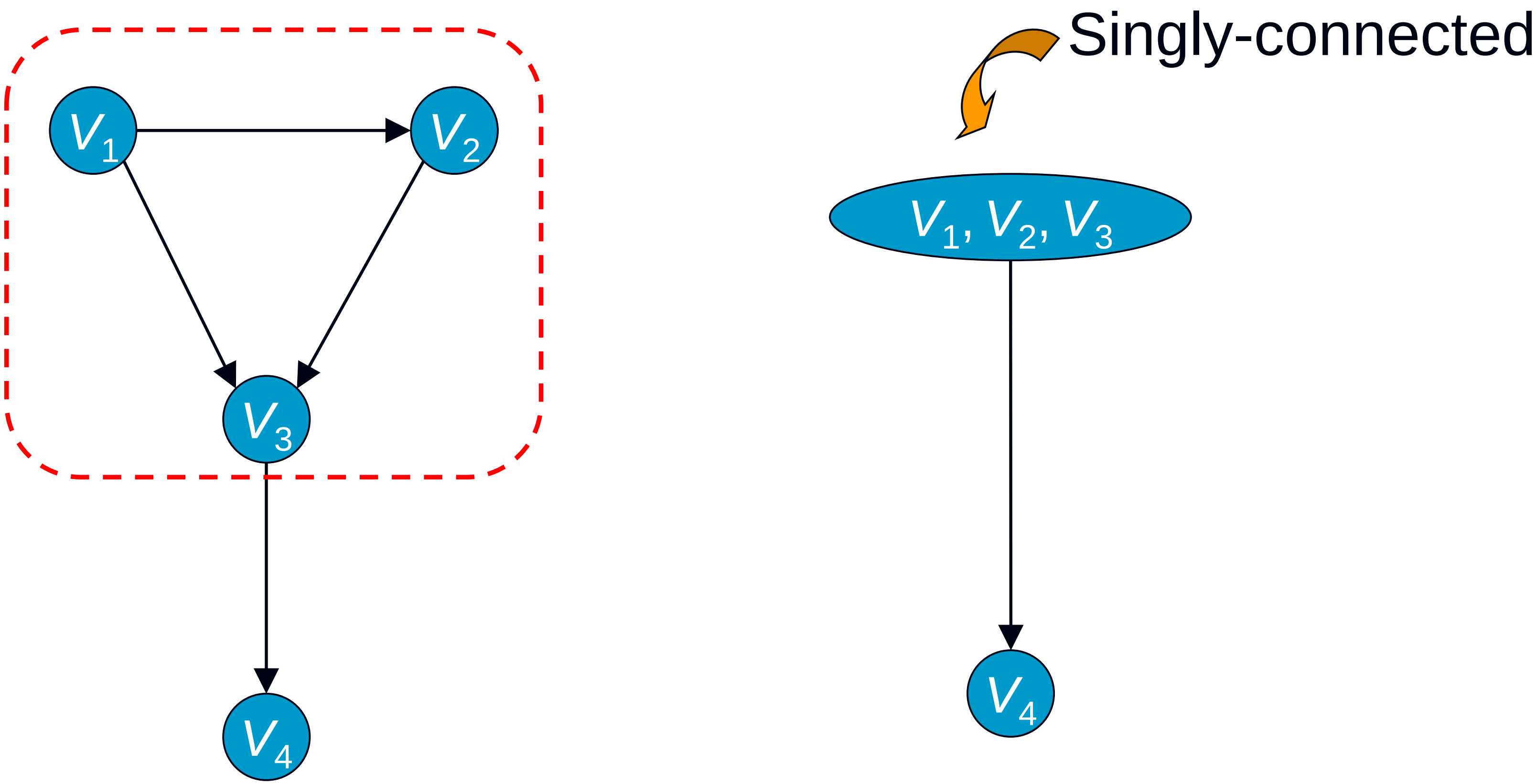
$$((P(e=0\,|\,a=1), P(e=1\,|\,a=1)) = \beta(\mu_{E\to e}(0)\mu_{A\to e}(0), \mu_{E\to e}(1)\mu_{A\to e}(1)) = (0.651, 0.349)$$

Inference in Multiply-Connected Networks

- Probabilistic inference in Bayesian networks (also in Markov random fields and factor graphs) in general is very hard. (NP-hardness's been proved.)

- Approximate inference
  - Use probability propagation in multiply-connected networks. → Loopy belief propagation.
  - Monte Carlo methods → Sampling
  - Variational inference
  - Helmholtz machines

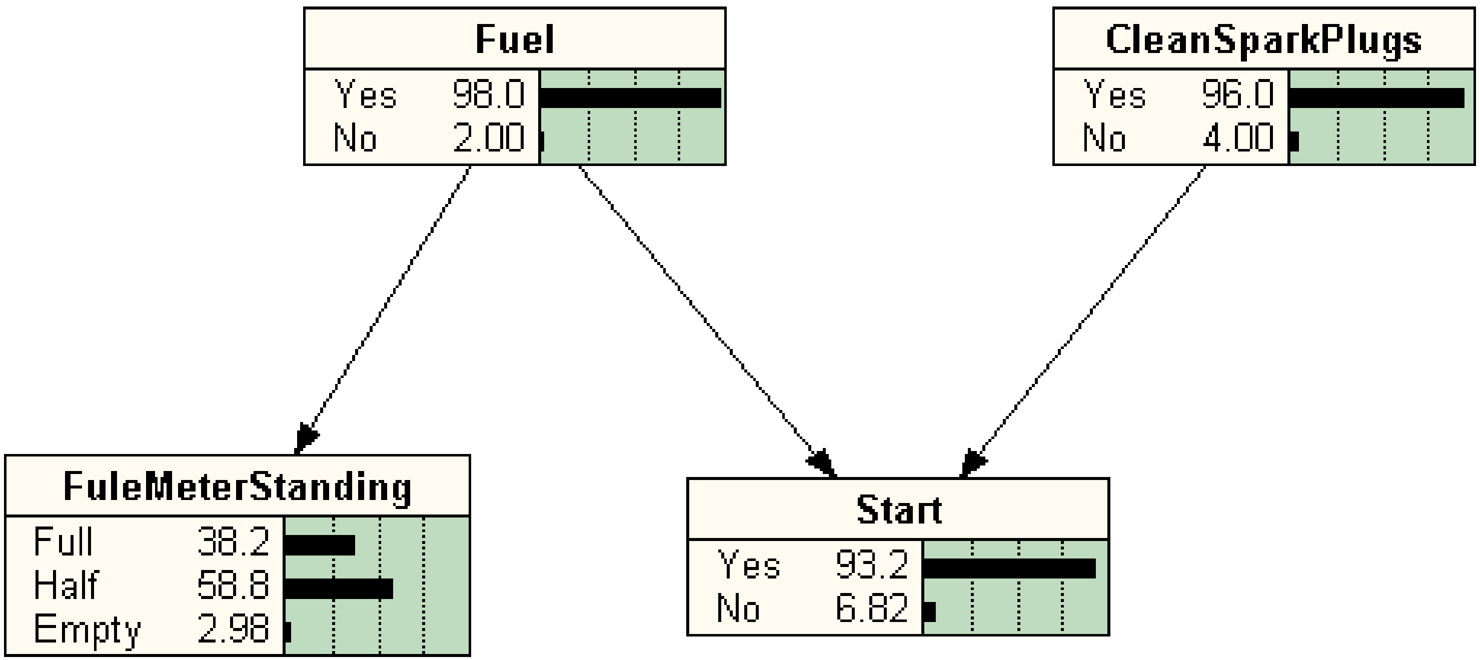
Grouping and Duplicating Variables



The new variable $\{V_1, V_2, V_3\}$ has values exponential to the number of included variables.

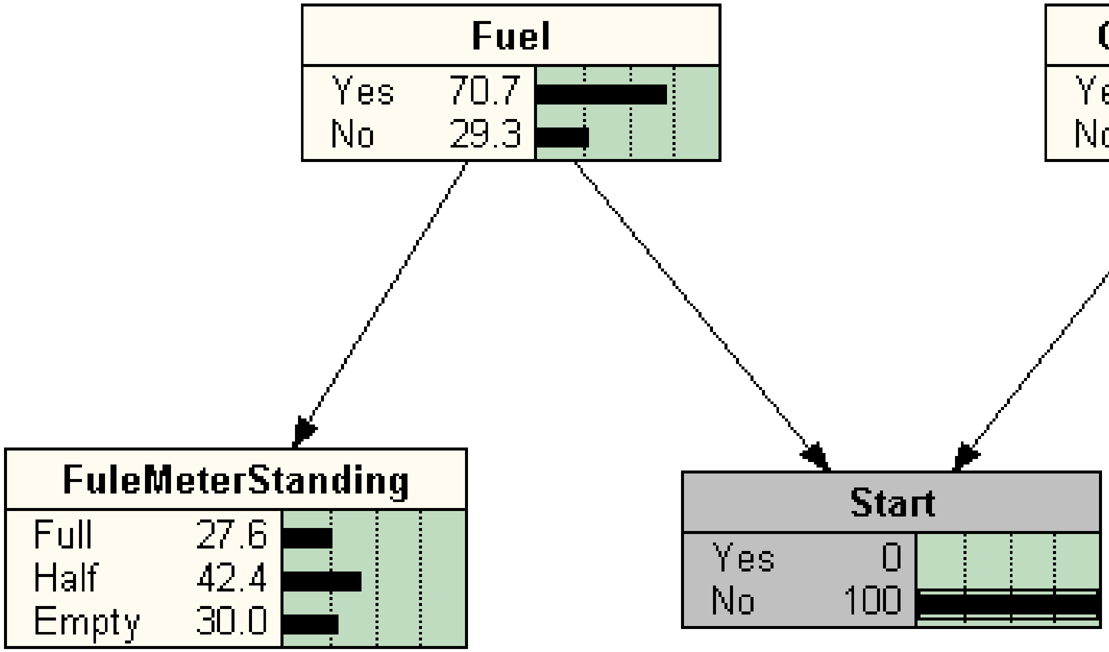
Initial State

P(*Fu*), *P*(*CSP*), *P*(*St*), and *P*(*FMS*)



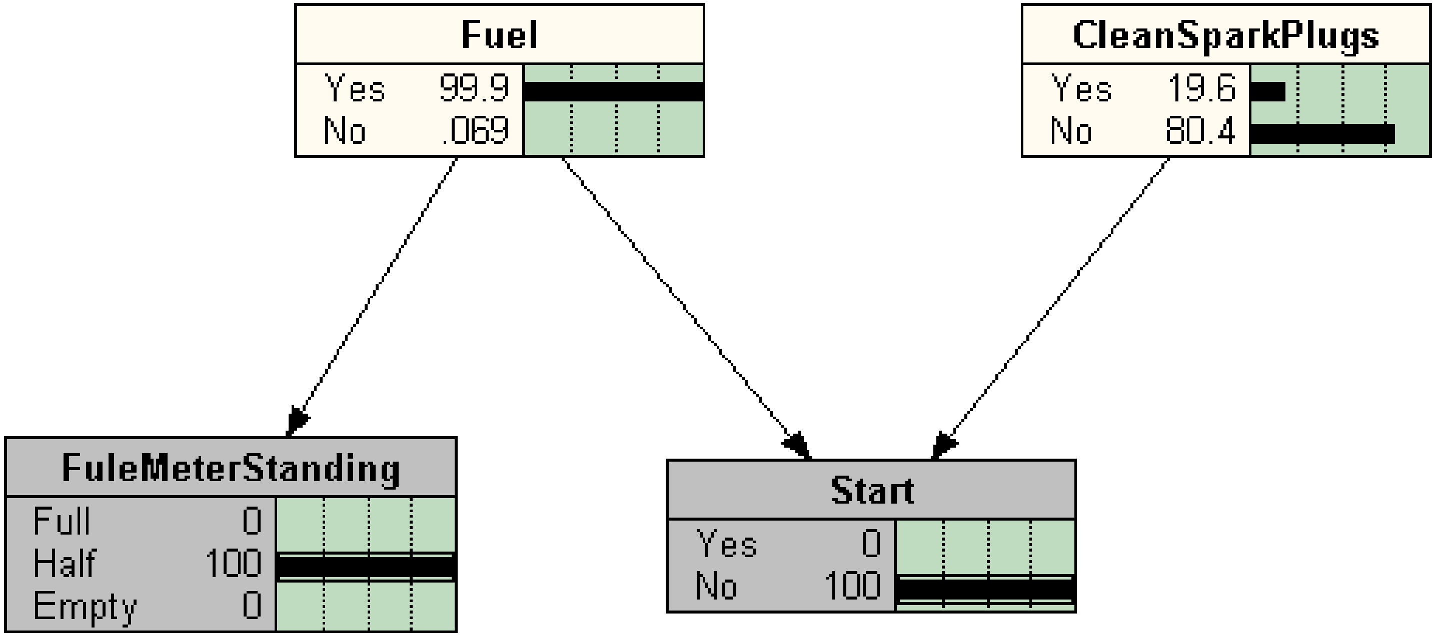
No Start

$P(Fu|St = \text{No})$, $P(CSP|St = \text{No})$, and $P(FMS|St = \text{No})$



Fuel Meter Stands for Half

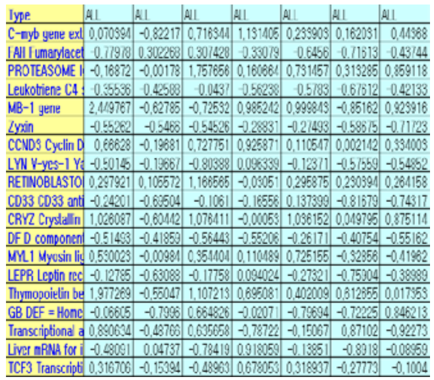
$P(Fu|St = \text{No}, FMS = \text{Half})$ and $P(CSP|St = \text{No}, FMS = \text{Half})$



- Basic Concepts of Bayesian Networks
- Inference in Bayesian Networks
- Learning Bayesian Networks
  - Parametric Learning
  - Structural Learning
- Conclusion

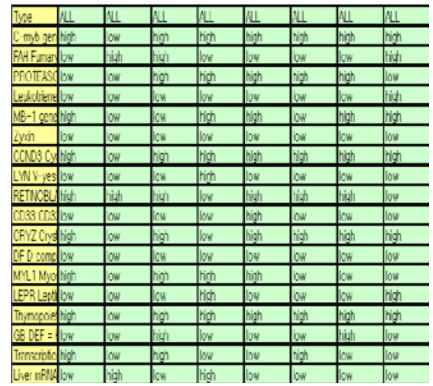
Learning Bayesian Networks

Data Acquisition

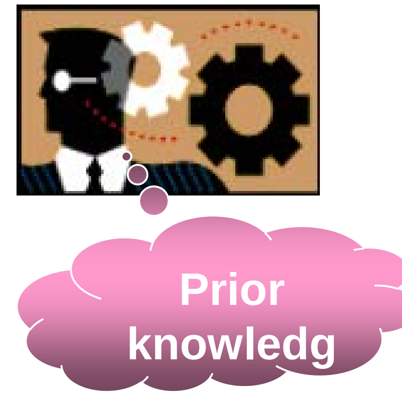


Type	ALL	ALL	ALL	ALL	ALL	ALL	ALL
C-myb gene ext	0,070394	-0,82217	0,716344	1,131405	0,233903	0,162031	0,44368
FAH Fumarylacet	-0.77978	0.302268	0.307428	-0.33079	-0.6456	-0.71613	-0.43744
PROTEASOME I	-0,16872	-0,00178	1,757656	0,160664	0,731457	0,313285	0,859118
Leukotriene C4	-0.35536	-0.42588	-0.0437	-0.56238	-0.5783	-0.67612	-0.42133
MB-1 gene	2,449767	-0,62785	-0,72532	0,985242	0,999843	-0,85162	0,923916
Zyxin	-0.55262	-0.5466	-0.54526	-0.28831	-0.27493	-0.58675	-0.71723
CCND3 Cyclin D	0,66628	-0,19681	0,727751	0,925871	0,110547	0,002142	0,334003
LYN V-yes-1 Ya	-0.50145	-0.19667	-0.80388	0.096339	-0.12371	-0.57559	-0.54852
RETINOBLASTO	0,297921	0,105572	1,166565	-0,03051	0,295875	0,230394	0,264158
CD33 CD33 anti	-0.24201	-0.63504	-0.1061	-0.16556	0.137399	-0.81679	-0.74317
CRYZ Crystallin	1,026087	-0,60442	1,076411	-0,00053	1,036152	0,049795	0,875114
DF D component	-0.51493	-0.41859	-0.56443	-0.55206	-0.26171	-0.40754	-0.55162
MYL1 Myosin li	0,530023	-0,00984	0,354404	0,110489	0,725155	-0,32856	-0,41962
LEPR Leptin rec	-0.12785	-0.63088	-0.17758	0.094024	-0.27321	-0.75904	-0.38989
Thymopoietin be	1,977269	-0,55047	1,107213	0,695081	0,402009	0,612655	0,017353
GB DEF = Homc	-0.06605	-0.7996	0.664826	-0.02071	-0.79694	-0.72225	0.846213
Transcriptional a	0,890634	-0,48766	0,635658	-0,78722	-0,15067	0,87102	-0,92273
Liver mRNA for i	-0.48091	0.04737	-0.78419	0.918059	-0.13851	-0.8918	-0.08959
TCF3 Transcripti	0,316706	-0,15394	-0,48963	0,678053	0,318937	-0,27773	-0,1004

Preprocessing



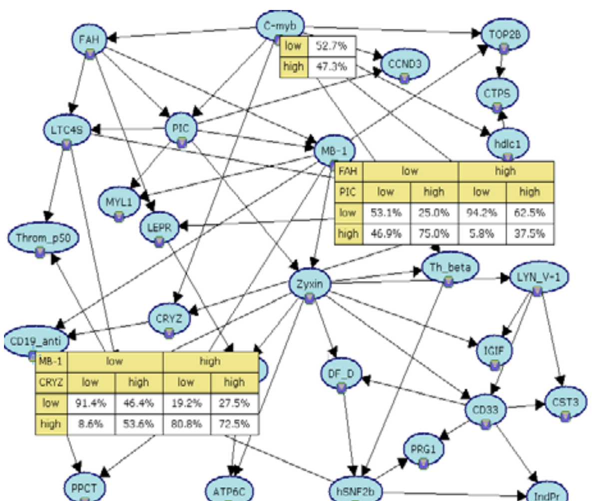
Type	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL
C-myb gen	high	low	high	high	high	high	high	high
FAH Fumar	low	high	high	low	low	low	low	high
PROTEASO	low	low	high	high	high	high	high	low
Leukotriene	low	low	low	low	low	low	low	high
MB-1 gene	high	low	low	high	high	low	high	high
Zyxin	low	low	low	low	low	low	low	low
CCND3 Cy	high	low	high	high	high	high	high	high
LYN V-yes	low	low	low	high	low	low	low	low
RETINOBLA	high	high	high	low	high	high	high	low
CD33 CD3	low	low	low	low	high	low	low	low
CRYZ Crys	high	low	high	low	high	high	high	high
DF D comp	low	low	low	low	low	low	low	low
MYL1 Myo	high	low	high	high	high	low	low	low
LEPR Lepti	low	low	low	high	low	low	low	high
Thymopoiet	high	low	high	high	high	high	high	high
GB DEF =	low	low	high	low	low	low	high	low
Transcriptio	high	low	high	low	low	high	low	low
Liver mRNA	low	high	low	high	low	low	low	low



Bayesian network learning

* Structure Search
* Score Metric
* Parameter Learning

BN = Structure
+ Local
probability
distribution



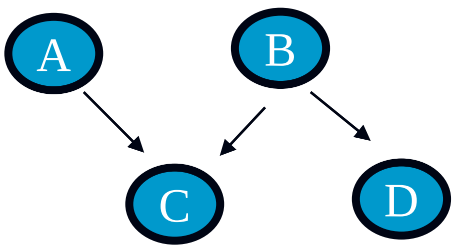
Learning Bayesian Networks (cont'd)

- Bayesian network learning consists of
  - Structure learning (DAG structure),
  - Parameter learning (for local probability distribution).

- Situations
  - Known structure and complete data.
  - Unknown structure and complete data.
  - Known structure and incomplete data.
  - Unknown structure and incomplete data.

Parameter Learning

- Task: Given a network structure, estimate the parameters of the model from data.



	A	B	C	D
S1	H	L	L	L
S2	H	H	H	H
	...	..	...	...
S_M	L	H	H	L

P(A)	
H	L
0.99	0.01

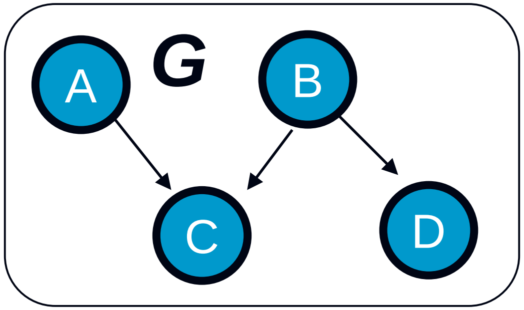
P(B)	
H	L
0.93	0.07

P(C|A, B)		
(A, B)	H	L
(H, H)	0.4	0.6
(H, L)	0.2	0.8
(L, H)	0.3	0.7
(L, L)	0.8	0.2

P(D|B)		
B	H	L
H	0.9	0.1
L	0.1	0.9

Key point: independence of parameter estimation

- D={$\mathbf{s}_1$, $\mathbf{s}_2$, …, $\mathbf{s}_M$}, where $\mathbf{s}_i$=(a_i, b_i, c_i, d_i) is an instance of a random vector variable **S**=(A, B, C, D).
- Assumption: samples $\mathbf{s}_i$ are *independent and identically distributed (i.i.d.)*.

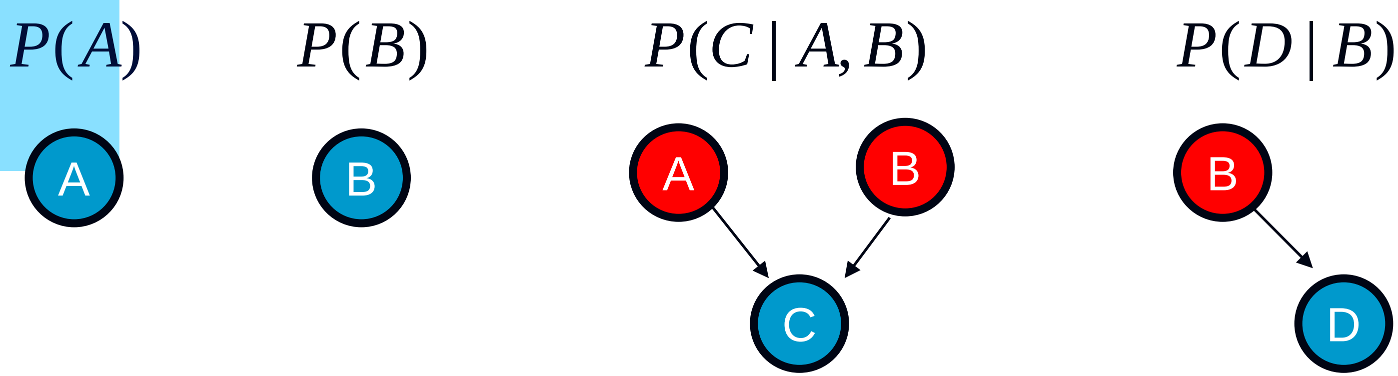


$$L_G(\Theta; D) = P_G(D \mid \Theta) = \prod_{i=1}^{M} P_G(\mathbf{s}_i \mid \Theta)$$

$$= \prod_{i=1}^{M} \left[P_G(a_i|\Theta) P_G(b_i|\Theta) P_G(c_i|a_i,b_i,\Theta) P_G(d_i|b_i,\Theta) \right]$$

$$= \left(\prod_{i=1}^{M} P_G(a_i|\Theta) \right) \left(\prod_{i=1}^{M} P_G(b_i|\Theta) \right) \left(\prod_{i=1}^{M} P_G(c_i|a_i,b_i,\Theta) \right) \left(\prod_{i=1}^{M} P_G(d_i|b_i,\Theta) \right)$$

Independent parameter estimation for each node (variable)



- One can estimate the parameters for $P(A)$, $P(B)$, $P(C|A, B)$, and $P(D|B)$ in an independent manner.
  - If A, B, C, and D are all binary-valued, the number of parameters are reduced from 15 (2^4-1) to 8 (1+1+4+2).

$$
\begin{array}{cc}
 & \begin{array}{cccc} \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \end{array} \\
\begin{array}{c} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \\ \mathbf{s}_M \end{array} & \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ \vdots & \vdots & \vdots & \vdots \\ a_M & b_M & c_M & d_M \end{bmatrix}
\end{array}
\quad P(A,B,C,D)=P(A)\mathrm{x}P(B)\mathrm{x}P(C|A,B)\mathrm{x}P(D|B)
$$

Methods for Parameter Estimation

- Maximum Likelihood Estimation
  - Choose the value of Θ which maximizes the likelihood for the observed data *D*.

$$\hat{\Theta} = \arg\max_{\Theta} L_G(\Theta; D) = \arg\max_{\Theta} P(D \mid \Theta)$$

- Bayesian Estimation
  - Represent uncertainty about parameters using a probability distribution over Θ.
  - Θ is also a random variable rather than a parameter value.

$$\underbrace{P(\Theta \mid D)}_{\text{posterior}} = \frac{P(\Theta)P(D \mid \Theta)}{P(D)} \propto \underbrace{P(\Theta)}_{\text{prior}} \underbrace{P(D \mid \Theta)}_{\text{likelihood}}$$

Bayes Rule, MAP and ML

- Bayes' rule

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

h: hypothesis (models or parameters)
D: data

- **ML** (maximum likelihood) estimation

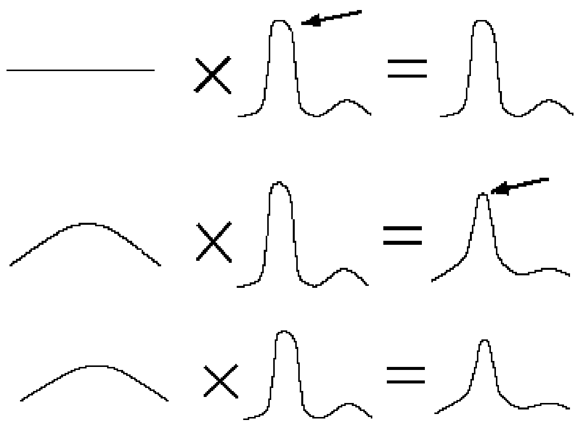
$$h^* = \arg\max_h P(D \mid h)$$

- **MAP** (maximum a posteriori) estimation

$$h^* = \arg\max_h P(h \mid D)$$

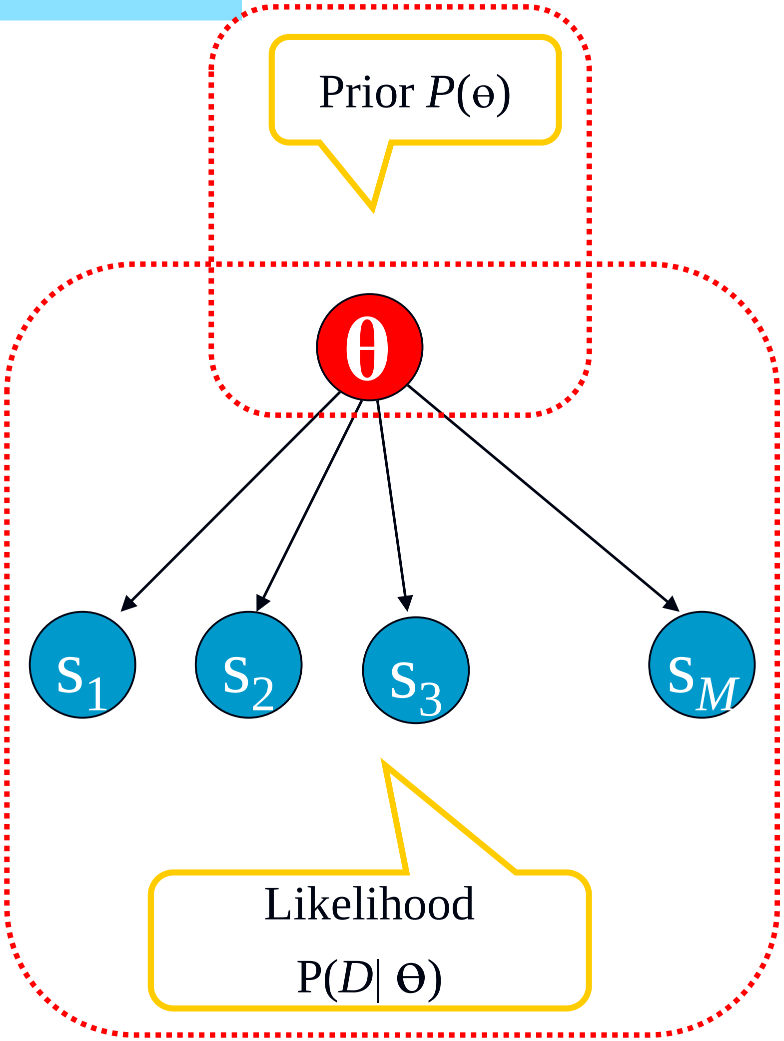
- **Bayesian Learning**
  - Not a point estimate, but the posterior distribution

$$P(h \mid D)$$



From NIPS'99 tutorial by Z. Ghahramani

Bayesian estimation (for multinomial distribution)



Prior knowledge or pseudo counts

$$\theta \sim Dir(\alpha_1, \alpha_2, \ldots, \alpha_K)$$

$$\Rightarrow P(\theta) \propto \prod_k \theta_k^{\alpha_k - 1}$$

Sufficient statistics

$$P(\theta \mid D) \propto P(\theta) P(D \mid \theta)$$

$$\propto \prod_k \theta_k^{\alpha_k + N_k - 1}$$

$$P(S_{M+1} = k \mid D) = \int P(k \mid \boldsymbol{\theta}) P(\boldsymbol{\theta} \mid D) d\boldsymbol{\theta}$$

$$= \int \theta_k P(\boldsymbol{\theta} \mid D) d\boldsymbol{\theta}$$

$$= E_{P(\boldsymbol{\theta}|D)}[\theta_k] = \frac{\alpha_k + N_k}{\sum_l (\alpha_l + N_l)}$$

Smoothed version of MLE

An Example: Coin toss

Maximum likelihood estimation

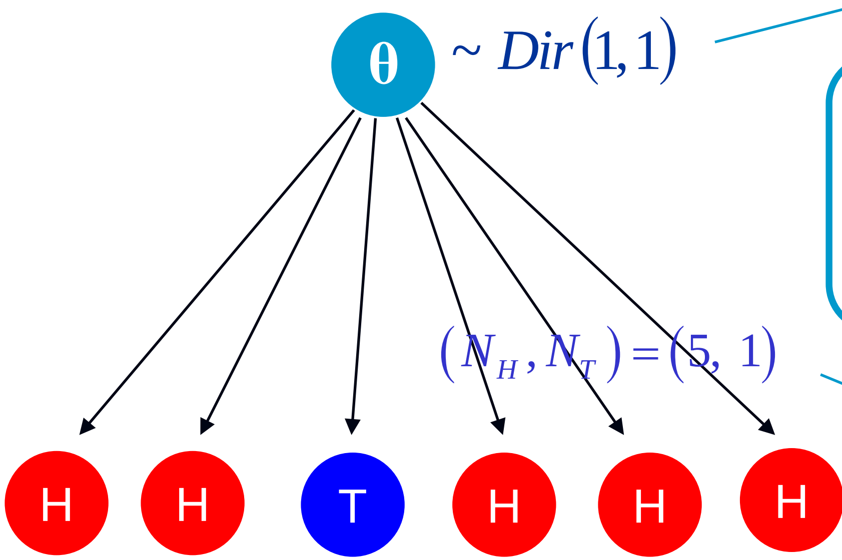
H H T H H H

$$P(D\,|\,\boldsymbol{\theta}) = \theta_H\theta_T\theta_H\theta_H\theta_H\theta_H = \theta_H^5\theta_T^1$$

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} P(D\,|\,\boldsymbol{\theta}) = \frac{5}{5+1} = \frac{5}{6}$$

$$P(S=H) = \hat{\theta}_H \approx 0.833$$

Bayesian inference



$$P(\boldsymbol{\theta}) \propto \theta_H^{(1-1)}\theta_T^{(1-1)}$$

$$P(\theta\,|\,D) \propto P(\theta)\times P(D\,|\,\theta) = \theta_H^5\theta_T^1$$

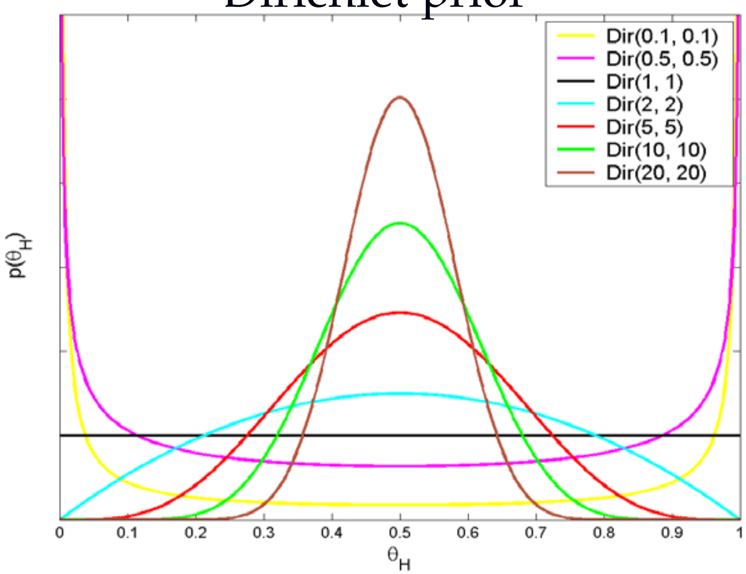
$$P(\mathrm{H}\,|\,D) = \frac{1+5}{(1+5)+(1+1)} = \frac{3}{4} = 0.75$$

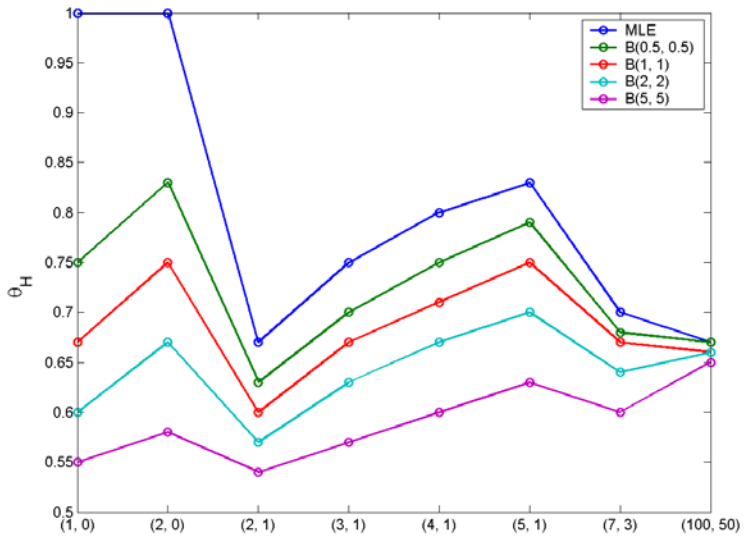
$$P(D\,|\,\boldsymbol{\theta}) = \theta_H\theta_T\theta_H\theta_H\theta_H\theta_H = \theta_H^5\theta_T^1$$

P(H)

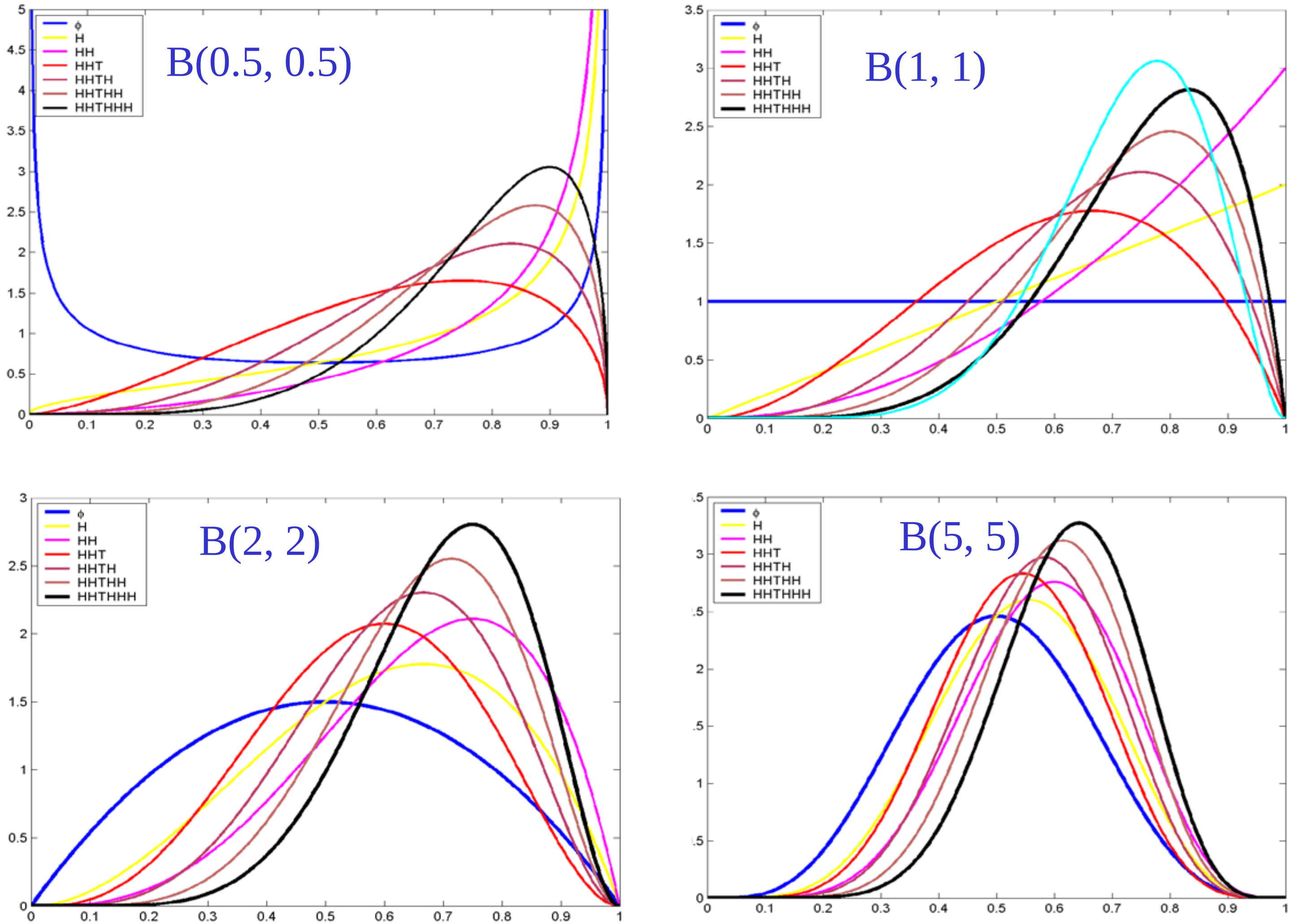
	H	HH	HHT	HHTH	HHTHH	HHTHH H	HHTHH HTTHH	(100, 50)
MLE	1.00	1.00	0.67	0.75	0.80	0.83	0.70	0.67
B(0.5, 0.5)	0.75	0.83	0.63	0.70	0.75	0.79	0.68	0.67
B(1, 1)	0.67	0.75	0.60	0.67	0.71	0.75	0.67	0.66
B(2, 2)	0.60	0.67	0.57	0.63	0.67	0.70	0.64	0.66
B(5, 5)	0.55	0.58	0.54	0.57	0.60	0.63	0.60	0.65

Dirichlet prior





Variation of posterior distribution for the parameter



Structure Learning

- Task: Given a data set, search a most plausible network structure underlying the generation of the data set.

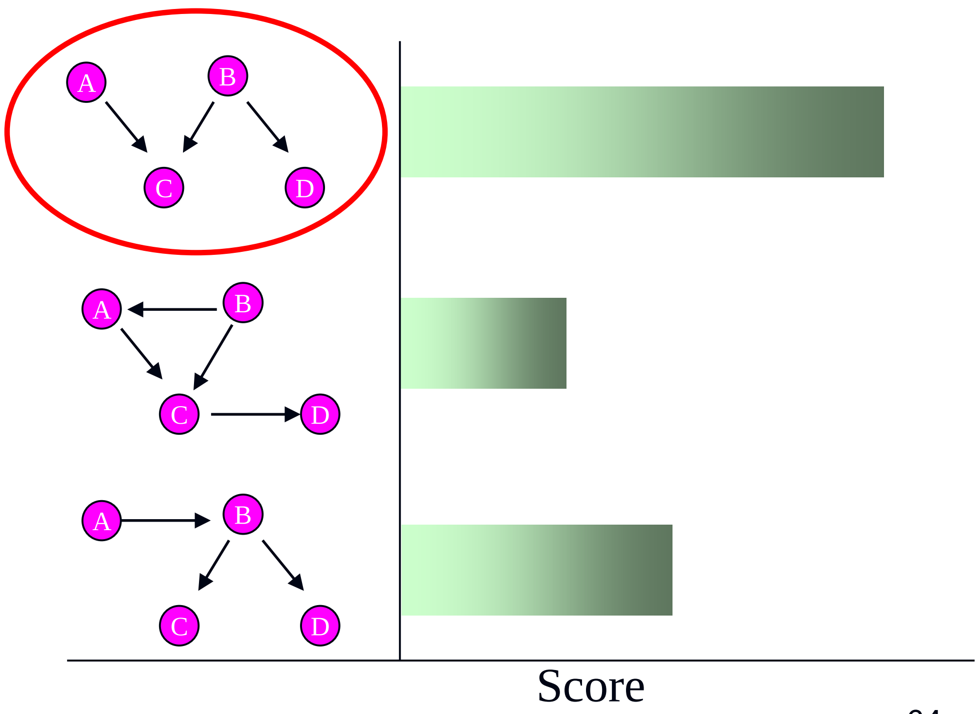
  Metric-based approach

  Use a scoring metric to measure how well a particular structure fits the observed set of cases.

	A	B	C	D
S1	H	L	L	L
S2	H	H	H	H
	...	...	...	...
S_M	L	H	H	L

Scoring metric

Search strategy



Scoring Metric

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

Prior for network structure

Marginal likelihood

- Likelihood Score

$$Score(G;D) = \log P\left(D \mid G, \Theta_{MLE}\right) \propto \sum_{i=1}^{N} I(X_i;\mathbf{Pa}_i) - \sum_{i=1}^{N} H(X_i)$$

  - Nodes of high mutual information (dependency) with their parents get higher score.
  - Since, I(X;Y) ≤ I(X; {Y, Z}), the fully connected network is obtained in an unrestricted case.
  - Prone to overfitting.

Likelihood Score in Relation with Information Theory

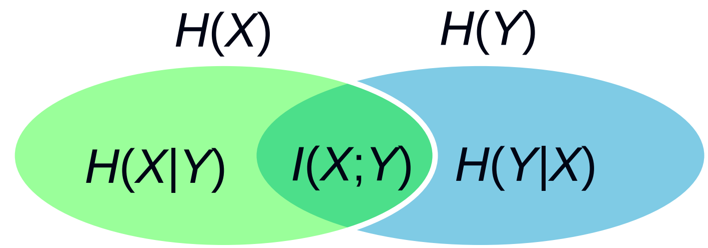
$$\log L(\hat{\Theta};\mathrm{D}) = \sum_{i=1}^{N}\sum_{j=1}^{|\mathbf{Pa}_i|}\sum_{k=1}^{|X_i|} N_{ijk}\log\frac{N_{ijk}}{N_{ij}}$$

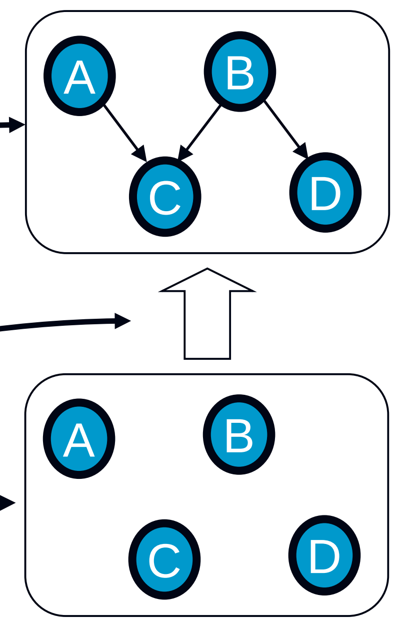
$$= M\sum_{i=1}^{N}\sum_{j=1}^{|\mathbf{Pa}_i|}\sum_{k=1}^{|X_i|}\frac{N_{ijk}}{M}\log\frac{N_{ijk}}{N_{ij}}$$

$$= -M\sum_{i=1}^{N} H\left(X_i \mid \mathbf{Pa}_i\right)$$

$$= M\left(\sum_{i=1}^{N}\left(H(X_i) - H(X_i \mid \mathbf{Pa}_i)\right) - \sum_{i=1}^{N} H(X_i)\right)$$

$$\propto \sum_{i=1}^{N} I(X_i;\mathbf{Pa}_i) \quad -\sum_{i=1}^{N} H(X_i)$$





Bayesian Score

- Consider the uncertainty in parameter estimation in Bayesian network

$$Score(G;D) = P(G)\int P(D \mid G, \Theta)P(\Theta \mid G)d\Theta$$

  - Assuming a complete data and parameter independence, the marginal likelihood can be rewritten as

$$P(D \mid G) = \prod_{i=1}^{N}\left[\int P\left(D\left(X_i;\mathbf{Pa}\left(X_i\right)\right)\middle| G, \boldsymbol{\theta}_i\right)P\left(\boldsymbol{\theta}_i \mid G\right)d\boldsymbol{\theta}_i\right]$$

Marginal likelihood for each pair of (X_i;**Pa**(X_i))

Bayesian Dirichlet Score

- For a multinomial case, if we assume a Dirichlet prior for each parameter (Heckerman, 1995),

$$\boxed{\int P\Big(D\big(X_i;\mathbf{Pa}(X_i)\big)\Big|\,G,\boldsymbol{\theta}_i\Big)P(\boldsymbol{\theta}_i\,|\,G)d\boldsymbol{\theta}_i=\prod_{j=1}^{|\mathbf{Pa}_i|}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})}\prod_{k=1}^{|X_i|}\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}}$$

$$\theta_{ij}\sim Dir\left(\alpha_{ij1},\alpha_{ij2},\ldots,\alpha_{ij|X_i|}\right)$$

$$\alpha_{ij}=\sum_{k=1}^{|X_i|}\alpha_{ijk}$$

$$N_{ijk}=\#\text{ of }\left(X_i=x_i^k,\mathbf{Pa}(X_i)=pa_i^j\right)$$

$$N_{ij}=\sum_{k=1}^{|X_i|}N_{ijk}$$

$$\Gamma(n+1)=n\Gamma(n)=\cdots=n!$$

$$\Gamma(1)=1$$

$$\Gamma(x)=\int_0^\infty t^{x-1}e^{-t}dt$$

Bayesian Dirichlet Score (cont'd)

H H T H

$$\theta \sim Dir(\alpha_H, \ \alpha_T) \qquad \alpha = \alpha_H + \alpha_T$$

$$P(H \mid \phi) = \frac{\alpha_H}{\alpha_H + \alpha_T}$$

$$P(H \mid H) \frac{(\alpha_H + 1)}{(\alpha_H + 1) + \alpha_T}$$

$$P(T \mid HH) = \frac{\alpha_T}{(\alpha_H + 2) + \alpha_T}$$

$$P(H \mid HHT) = \frac{(\alpha_H + 2)}{(\alpha_H + 2) + (\alpha_T + 1)}$$

$$\frac{\Gamma(\alpha_H + 3)}{\Gamma(\alpha_H)} \qquad \frac{\Gamma(\alpha_T + 1)}{\Gamma(\alpha_T)}$$

$$\frac{[\alpha_H \times (\alpha_H + 1) \times \cdots (\alpha_H + 2)] \times \alpha_T}{\alpha \times (\alpha + 1) \times \cdots (\alpha + 3)}$$

$$\frac{\Gamma(\alpha + 4)}{\Gamma(\alpha)}$$

$$\boxed{\frac{\Gamma(\alpha)}{\Gamma(\alpha + 4)} \left[\frac{\Gamma(\alpha_H + 3)}{\Gamma(\alpha_H)} \times \frac{\Gamma(\alpha_T + 1)}{\Gamma(\alpha_T)} \right]}$$

Bayesian Dirichlet Score (cont'd)

$$P(D \mid G) = \prod_{i=1}^{N} \left[\int P\big(D\big(X_i; \mathbf{Pa}(X_i)\big) \big| G, \boldsymbol{\theta}_i\big) P(\boldsymbol{\theta}_i \mid G) d\boldsymbol{\theta}_i \right]$$

$$= \prod_{i=1}^{N} \prod_{j=1}^{|\mathbf{Pa}_i|} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{|X_i|} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

- log *P*(*D*|*G*) in Bayesian score is asymptotically equivalent to BIC and minus the MDL criterion.

$$\log P(D \mid G) \approx BIC(G; D) = \log P(D \mid \hat{\Theta}, G) - \frac{\dim(G)}{2} \log M$$

$$\dim(G) = \# \text{ of parameters in } G$$

Structure Search

- Given a data set, a score metric, and a set of possible structures,
  - Find the network structure with maximal score.
  - Discrete optimization

- One can utilize the property of independent score for each pair of (X_i, **Pa**(X_i)).

$$P(D \mid G) = \prod_{i=1}^{N} P\left(D(X_i; \mathbf{Pa}(X_i)) \mid G\right)$$

$$\Rightarrow \quad Score(G; D) = \log P(D \mid G) = \sum_{i=1}^{N} Score(X_i; Pa(X_i))$$

Tree-Structured Networks

- Definition: Each node has *at most one parent*.
  - An effective search algorithm exists.

$$Score(G \mid D) = \sum_i Score(X_i \mid Pa_i) = \overbrace{\boxed{\sum_i \left[Score(X_i \mid Pa_i) - Score(X_i) \right]}}^{\text{Improvement over empty network}} + \overbrace{\boxed{\sum_i Score(X_i)}}^{\text{Score for empty network}}$$

Chow and Liu (1968)

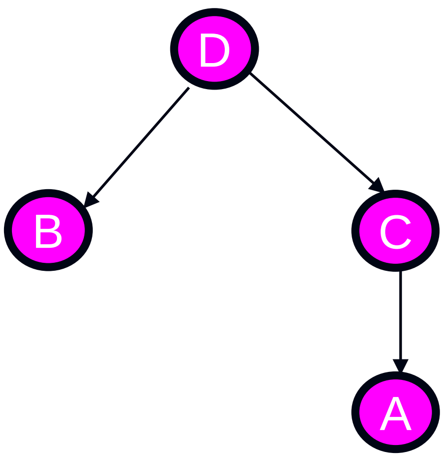
Construct the *undirected complete graph* with the weights of edge $E(X_i, X_j)$ being $I(X_i; X_j)$.

↓

Build a *maximum weighted spanning tree*.

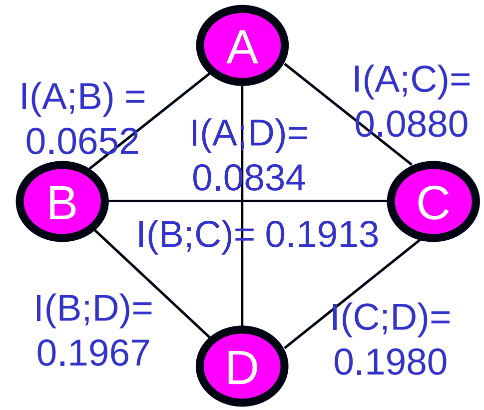
↓

Transform to *a directed tree* with an arbitrary root node.

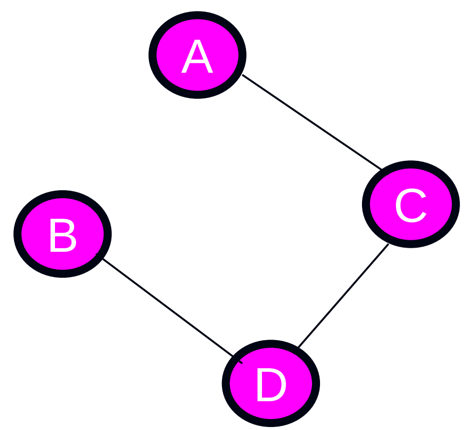


	A	B	C	D
S1	H	L	L	L
S2	H	H	H	H
	…	..	…	…
S_M	L	H	H	L

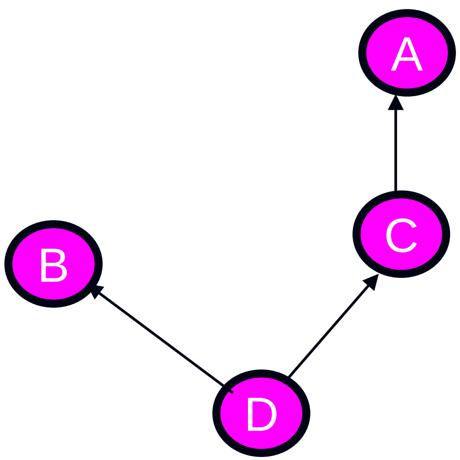
Complete undirected graph



Maximum spanning tree



Directed tree



Search Strategies for General Bayesian Networks

- With more than one parents per node ➔ NP-hard (Chickering *et al*., 1996)
  - Heuristic search methods are usually employed.
    - Greedy hill-climbing (local search)
    - Greedy hill-climbing with random restart
    - Simulated annealing
    - Tabu search
    - …

Greedy Local Search Algorithm

INPUT: Data set, Scoring Metric,
Possible Structures

Initialize the structure
- empty network, random network, tree-structured network, etc.

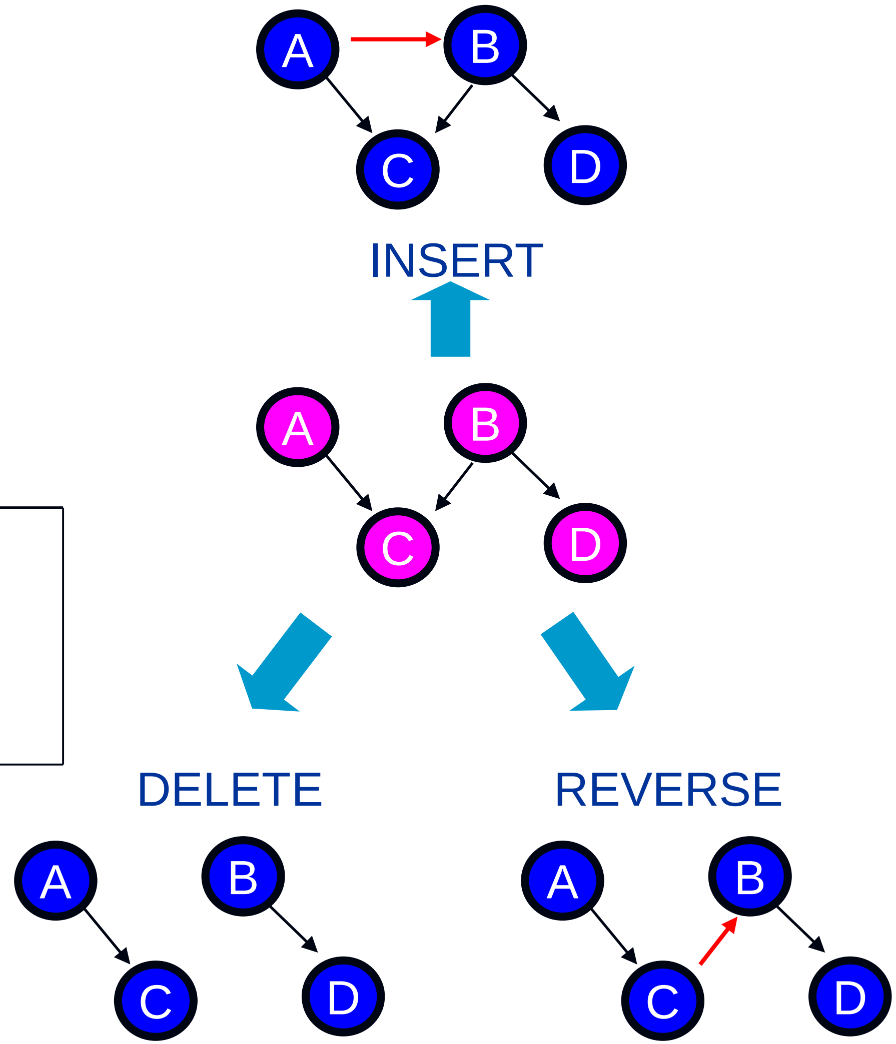
Do a local edge operation resulting in the largest improvement in score among all possible operations

$\text{Score}(G_{t+1};D) > \text{Score}(G_t;D)$

YES

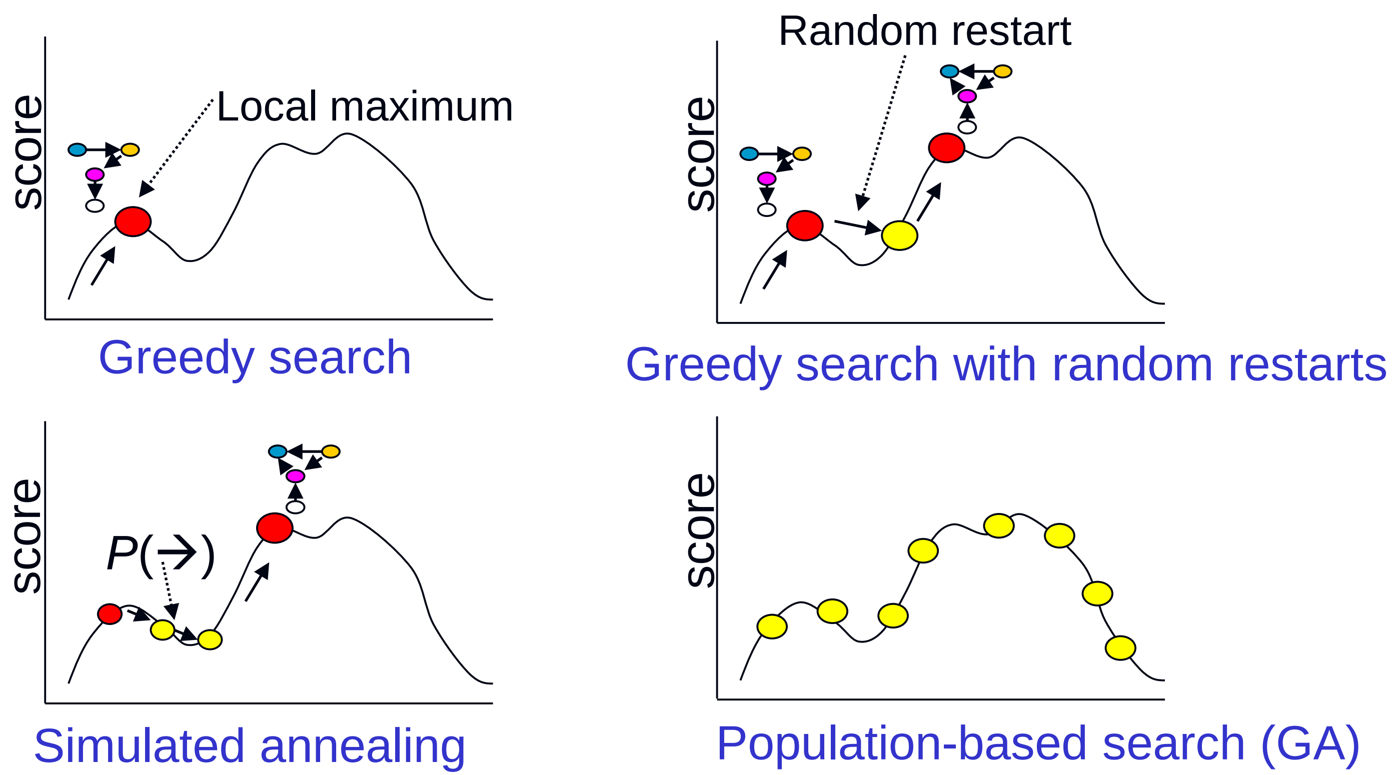
NO

$G_{\text{final}} = G_t$



Enhanced Search

- Greedy local search can get stuck in local maxima or plateaux.
- Standard heuristics to escape the two includes
  - Search with random restarts, simulated annealing, tabu search.
- Genetic algorithm: a population-based search.



- Basic Concepts of Bayesian Networks
- Inference in Bayesian Networks
- Learning Bayesian Networks
  - Parametric Learning
  - Structural Learning
- Conclusion

Conclusion

- Bayesian networks provide an efficient/effective framework for organizing the body of knowledge by encoding the probabilistic relationships among variables of interest.
  - Graph theory + probability theory: DAG + local probability distribution.
  - Conditional independence and conditional probability are keystones.
  - A compact way to express complex systems by simpler probabilistic modules and thus a natural framework for dealing with complexity and uncertainty.

- Two problems in the learning of Bayesian networks from data
  - Parameter estimation: MLE, MAP, Bayesian estimation
  - Structural learning: tree-structured network, heuristic search for general Bayesian network.

참고 문헌

- 베이즈망 소개
  - F. V. Jensen and T. D. Nielson, **Bayesian Networks and Decision Graphs**, 2nd Edition, Springer, 2007. (2장)
- 베이즈망 추론
  - B. J. Frey, **Graphical Models for Machine Learning and Digital Communication**, MIT Press, 1998. (2장)
- 베이즈망 학습
  - D. Heckerman, “A Tutorial on Learning with Bayesian Networks,” **Learning in Graphical Models**, M. I. Jordan (Ed.), pp. 301-354, Kluwer Academic Publishers, 1998.

Bayesian Network Software Packages

- Bayes Net Toolbox (by Kevin Murphy)
  - https://code.google.com/p/bnt/
  - A variety of algorithms for learning and inference in graphical models (written in MATLAB).
- WEKA
  - http://www.cs.waikato.ac.nz/~ml/weka/
  - Bayesian network learning and classification modules are included among a collection of machine learning algorithms (written in JAVA).
- A number of Bayesian network packages in R
  - http://www.bnlearn.com/
  - bnlearn, gRbase, and others.
- A detailed list and comparison are referred to
  - http://www.cs.ubc.ca/~murphyk/Bayes/bnsoft.html

Thank You

Source: Writing and reading a book with DNA, IEEE Spectrum (August 2012)