An Introduction to Bayesian Networks: Concepts and Learning from Data

Kyu-Baek Hwang and Byoung-Tak Zhang
Biointelligence Lab
School of Computer Science and Engineering
Seoul National University
kbhwang@bi.snu.ac.kr btzhang@cse.snu.ac.kr

- Introduction
- Basic Concepts of Bayesian Networks
- Learning Bayesian Networks
  - Parameter Learning
  - Structural Learning
- Applications
  - DNA microarray
  - Classification
  - Dependency Analysis
- Summary

Bayes Rule for Probabilistic Inference (1/3)

- Medical diagnosis based on knowledge base (from [Mitchell 97])
  - Prior probability of *cancer*
    - $P(cancer) = 0.008$, $P(\text{-}cancer) = 0.992$
  - A test for *cancer*
    - $P(+ \mid cancer) = 0.98$, $P(- \mid cancer) = 0.02$
    - $P(+ \mid \text{-}cancer) = 0.03$, $P(- \mid \text{-}cancer) = 0.97$

Bayes Rule for Probabilistic Inference (2/3)

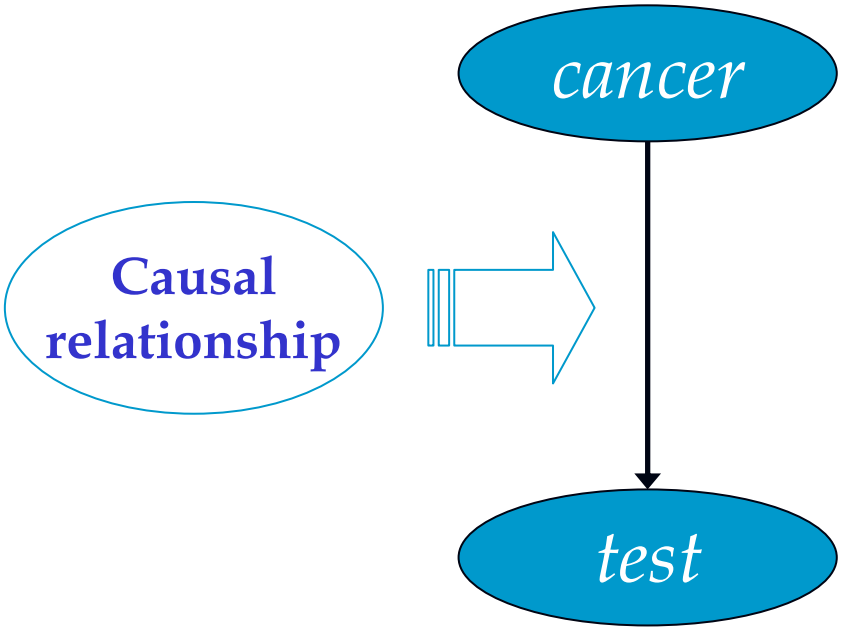
- If the result of the test is positive, how probable is the case of cancer?
  - We should calculate $P(cancer \mid +)$.
- Probabilistic inference from the given probabilities (knowledge).
  - $P(cancer \mid +) = P(+ \mid cancer) \cdot P(cancer) \; / \; P(+)$

$$= 0.98 \cdot 0.008 \; / \; P(+)$$

Bayes Rule for Probabilistic Inference (3/3)

- Marginalization
  - How to calculate $P(+)$?
  - $P(cancer \mid +) + P(\text{-}cancer \mid +)$ should be one.
    - $P(cancer \mid +) = 0.98 \cdot 0.008 \;/\; P(+) = 0.00784 \;/\; P(+)$
    - $P(\text{-}cancer \mid +) = P(+ \mid \text{-}cancer) \cdot P(\text{-}cancer) \;/\; P(+)$
      $= 0.03 \cdot 0.992 \;/\; P(+) = 0.02976 \;/\; P(+)$
    - $0.00784 \;/\; P(+) + 0.02976 \;/\; P(+) = 1$
    - $P(+) = 0.00784 + 0.02976 = 0.0376$
- $P(cancer \mid +) = 0.00784 \;/ 0.0376 = 0.21$

A Bayesian Network for Medical Diagnosis



$P(cancer) = 0.008$
$P(\text{-}cancer) = 0.992$

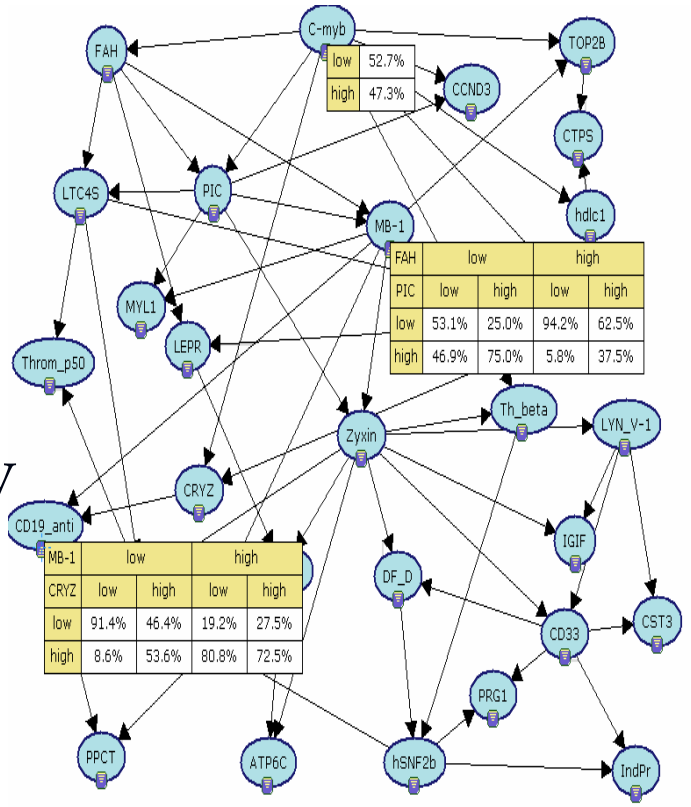
$P(+ \mid cancer) = 0.98$
$P(- \mid cancer) = 0.02$
$P(+ \mid \text{-}cancer) = 0.03$
$P(- \mid \text{-}cancer) = 0.97$

In our example, the knowledge base corresponds to the joint probability distribution.

$P(cancer) \cdot P(test \mid cancer) = P(test, cancer)$

Bayesian Networks

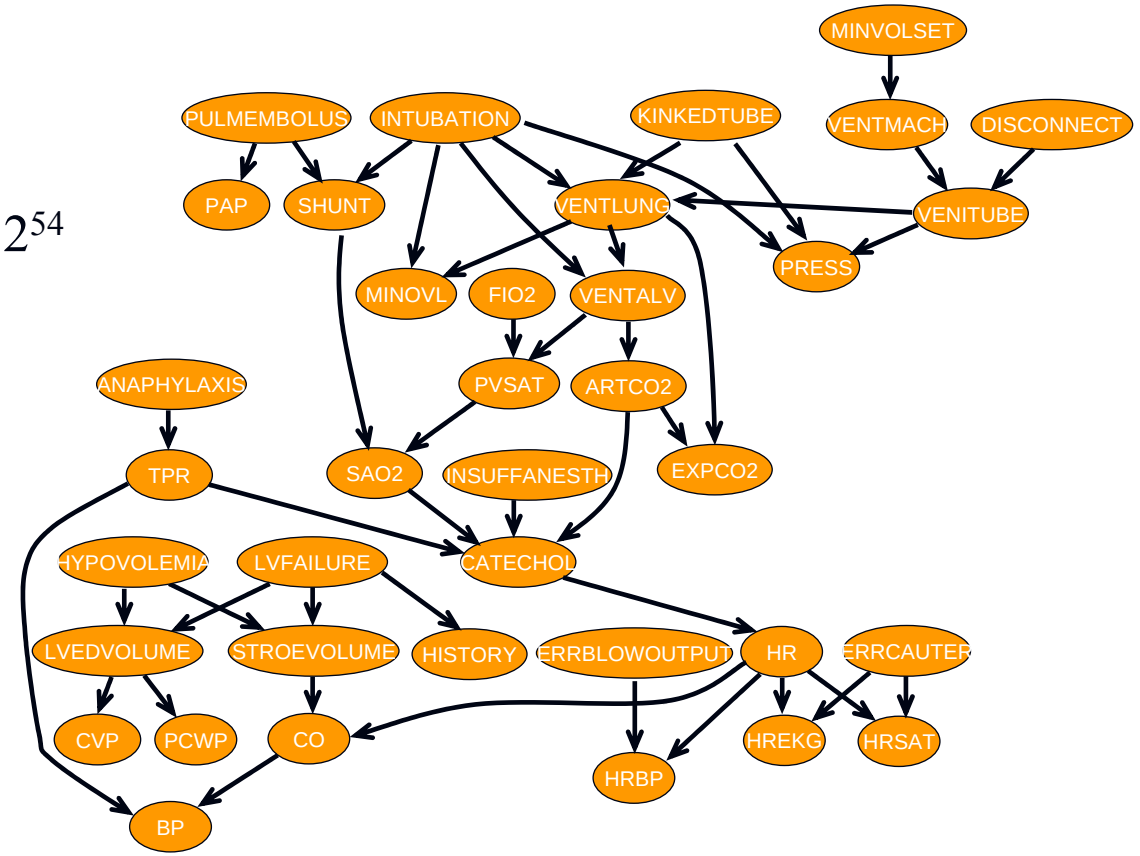
- A compact representation of knowledge base (probabilities)
- Qualitative part: graph theory
  - Directed acyclic graph (DAG)
  - Vertices: variables
  - Edges: dependency or influence of a variable on another.
- Quantitative part: probability theory
  - Set of (conditional) probabilities for all variables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



- Joint probability as a product of conditional probabilities
  - Can dramatically reduce the parameters for data modeling in Bayesian networks.

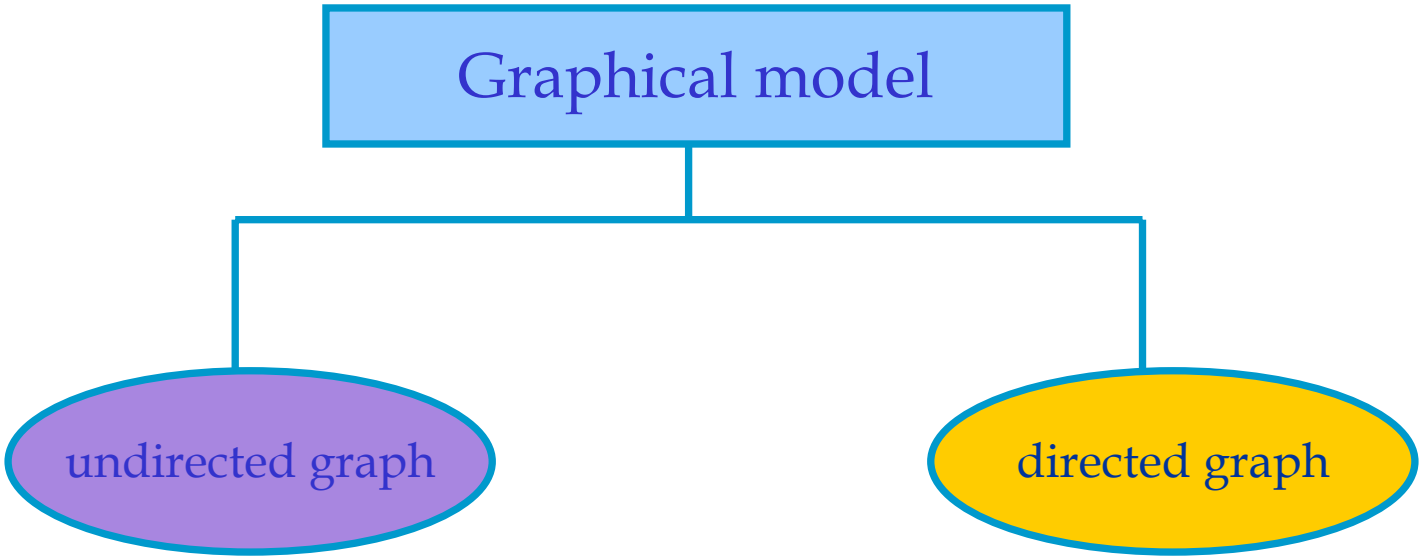
37 variables in total

509 parameters ←→ 2^{54}

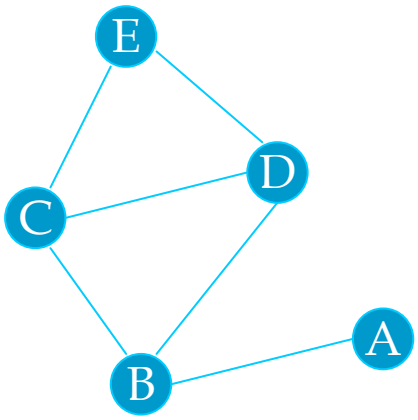


(From NIPS'01 tutorial by Friedman, N. and Daphne K.)

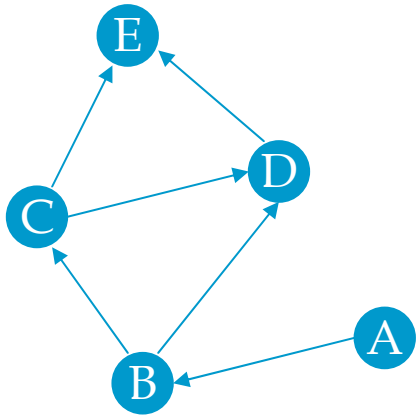
Probabilistic Graphical Models



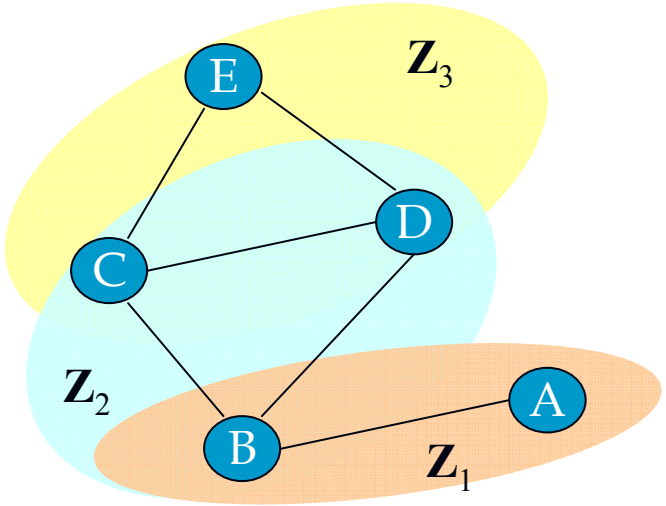
Markov Random Field



Bayesian Networks



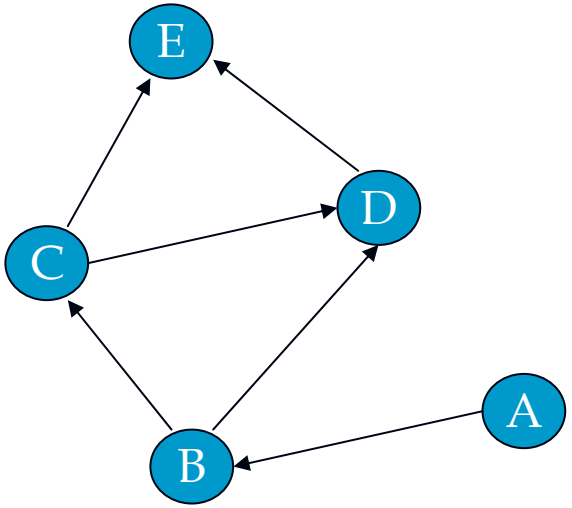
Representation of Joint Probability



$$P(A,B,C,D,E) = \frac{1}{\alpha}\prod_i \varphi_i(\mathbf{Z}_i)$$

$$= \frac{1}{\alpha}\varphi_1(A,B)\varphi_2(B,C,D)\varphi_3(C,D,E)$$

normalization constant



$$P(A,B,C,D,E) = P(A\,|\,\mathbf{Pa}(A))P(B\,|\,\mathbf{Pa}(B))P(C\,|\,\mathbf{Pa}(C))$$
$$P(D\,|\,\mathbf{Pa}(D))P(E\,|\,\mathbf{Pa}(E))$$

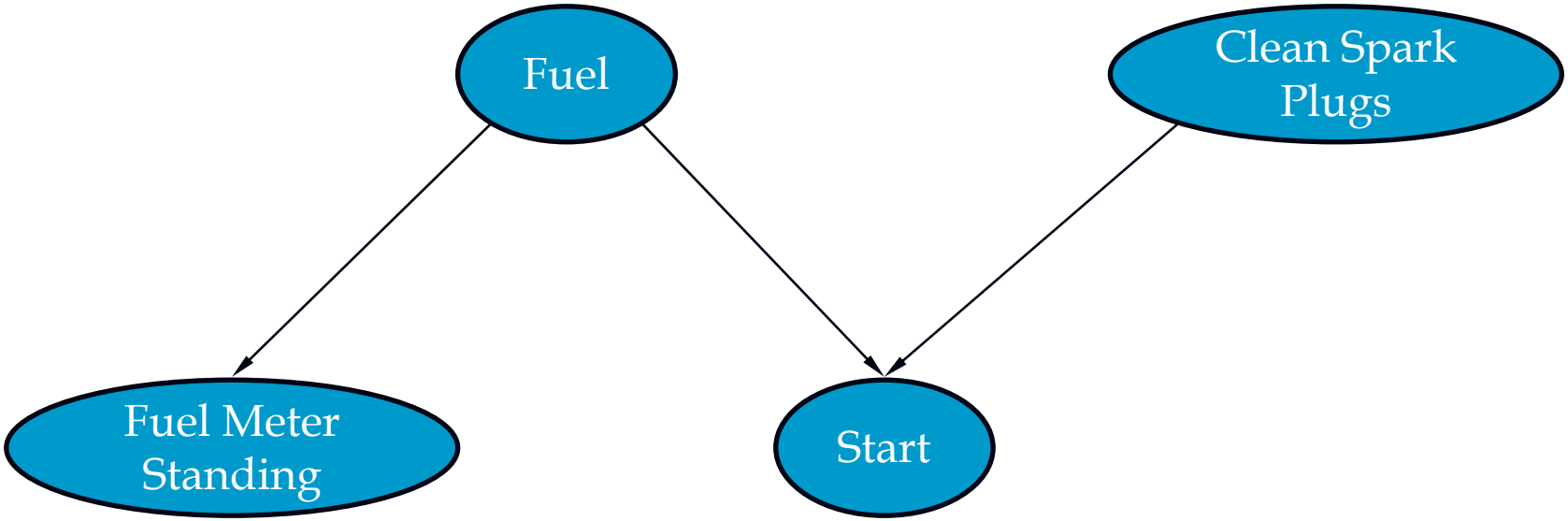
Real World Applications of BN

- Intelligent agents
  - Microsoft Office assistant: Bayesian user modeling
- Medical diagnosis
  - PATHFINDER (Heckerman, 1992): diagnosis of lymph node disease ➔ commercialized as INTELLIPATH (http://www.intellipath.com/)
- Control decision support system
- Speech recognition (HMMs)
- Genome data analysis
  - gene expression, DNA sequence, a combined analysis of heterogeneous data.
- Turbocodes (channel coding)

- Introduction
- Basic Concepts of Bayesian Networks
- Learning Bayesian Networks
  - Parameter Learning
  - Structural Learning
- Applications
  - DNA microarray
  - Classification
  - Dependency Analysis
- Summary

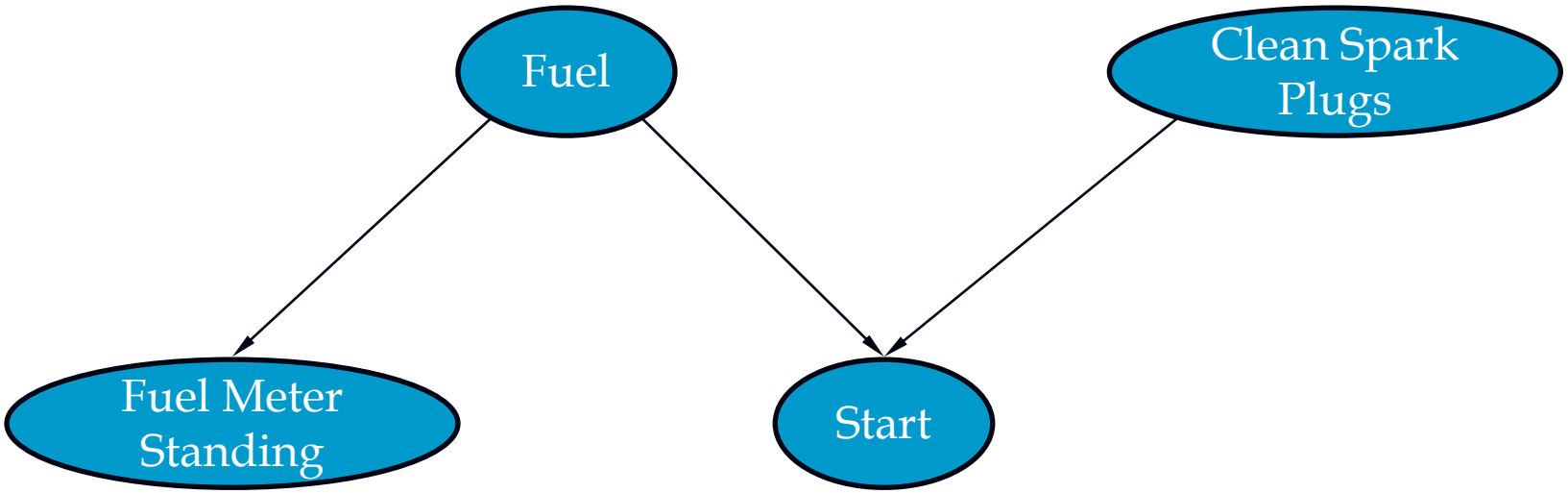
Causal Networks

- Node: event
- Arc: causal relationship between the two nodes
  - $A \rightarrow B$: A causes B.
- Causal network for the car start problem [Jensen 01]



Reasoning with Causal Networks

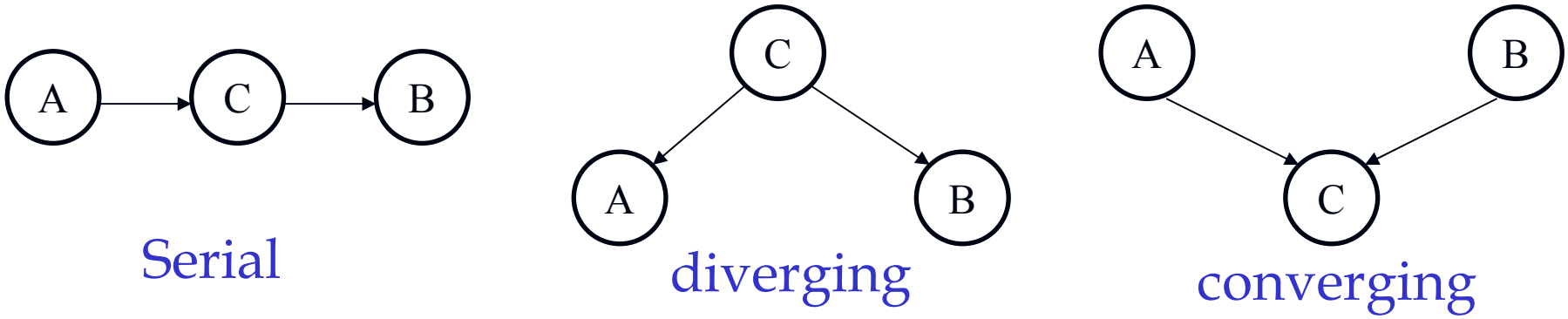
- My car does not start. → increases the certainty of no fuel and dirty spark plugs. → increases the certainty of fuel meter's standing for the empty.
- Fuel meter stands for the half. → decreases the certainty of no fuel → increases the certainty of dirty spark plugs.



d-Separation

A rule describing the influences between the nodes.

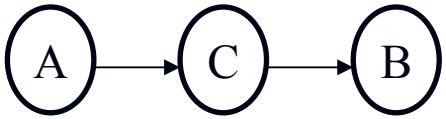
Connections in causal networks

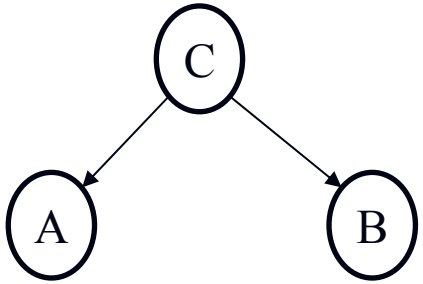


Definition [Jensen 01]:

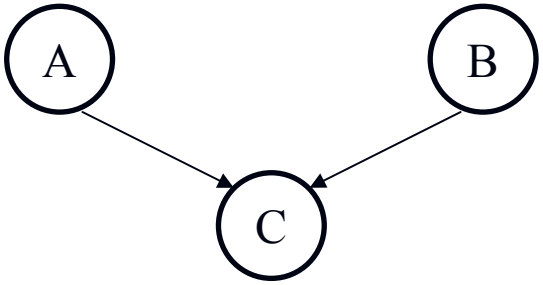
- Two nodes in a causal network are *d-separated* if for all paths between them there is an intermediate node *V* such that
  - the connection is *serial* or *diverging* and the state of *V* is known or
  - the connection is *converging* and neither *V* nor any of *V*'s descendants have received evidence.

d-Separation Example 1

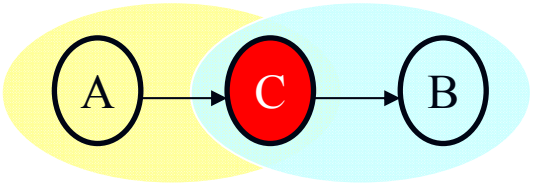


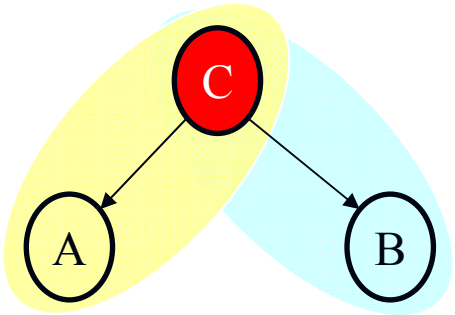


A and B is *marginally* dependent

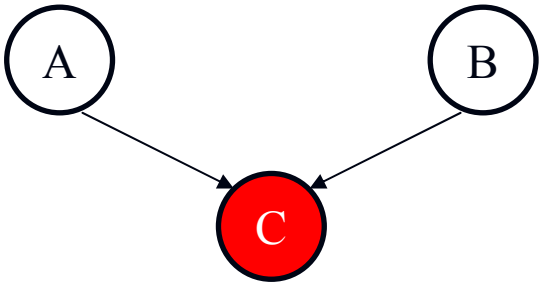


A and B is *marginally* independent



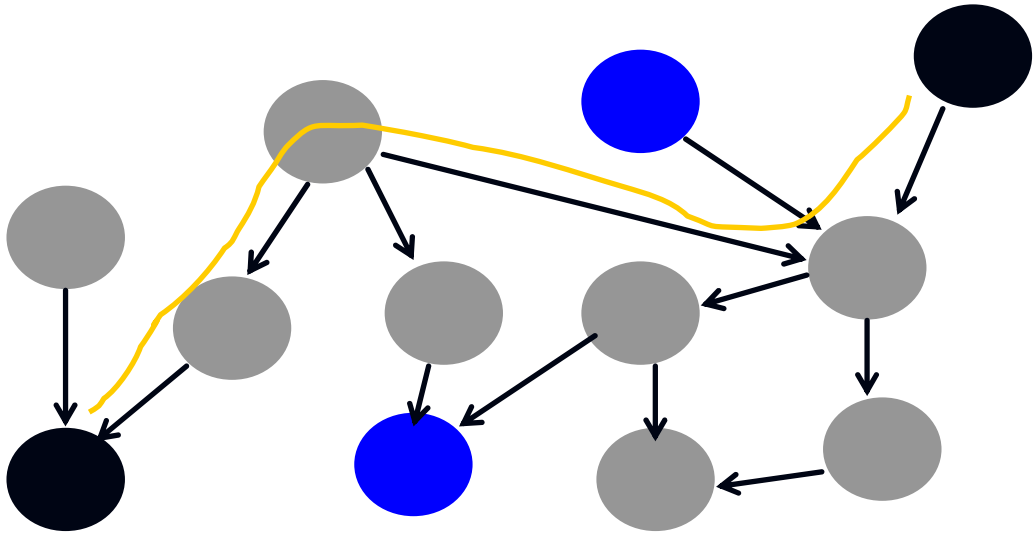


A and B is *conditionally* independent



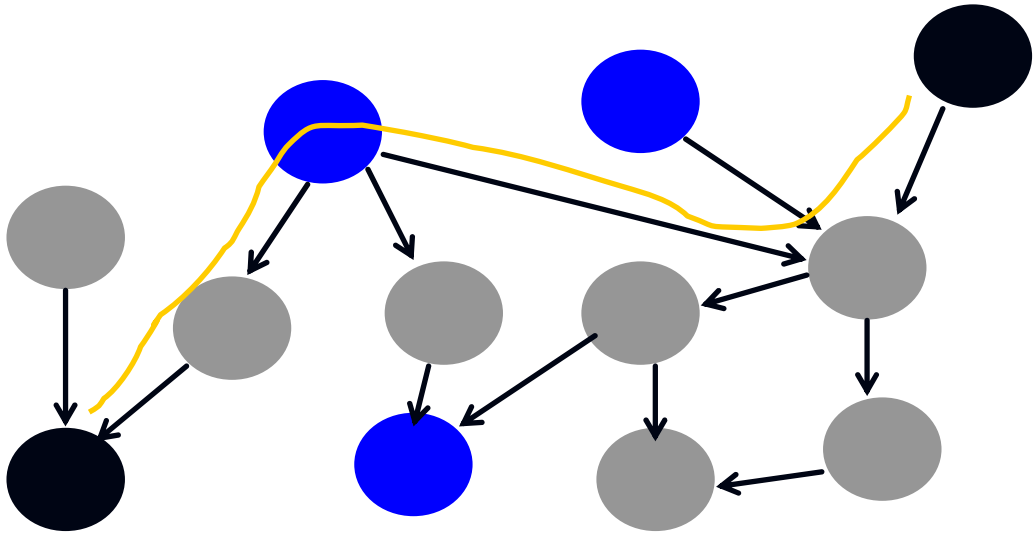
A and B is *conditionally* dependent

d-Separation Example 2



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

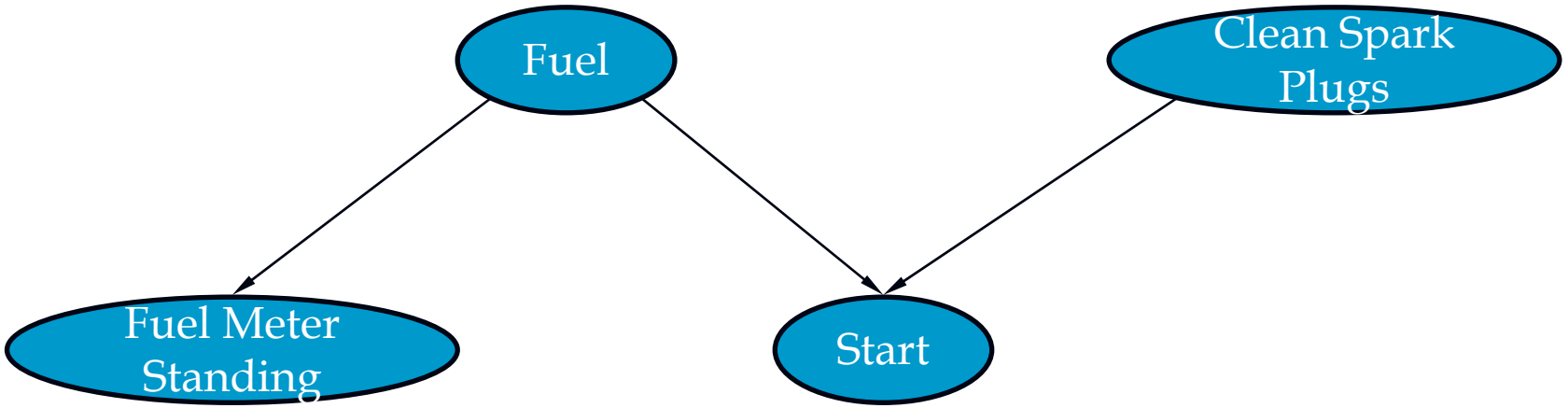
d-Separation Example 2



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

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’.



Probability for Quantifying Certainty in Causal Networks

- Basic axioms
  - $P(A) = 1$ iff A is certain.
    - $\Sigma_A P(A) = 1$ (summation is taken over all possible values of A.)
  - $P(A \cup B) = P(A) + P(B)$ iff A and B are mutually exclusive.
- *d*-separation in probability calculus
  - Event in the causal network → a random variable
  - If A and B are *d*-separated, then $P(A \mid B) = P(A)$.
    - A and B are probabilistically independent.

Quantitative Specification by Probability Calculus

- Fundamentals
  - Conditional Probability

$$P(B \mid A) = \frac{P(A,B)}{P(A)}$$

  - Product Rule

$$P(A,B) = P(B \mid A)P(A) = P(A \mid B)P(B)$$

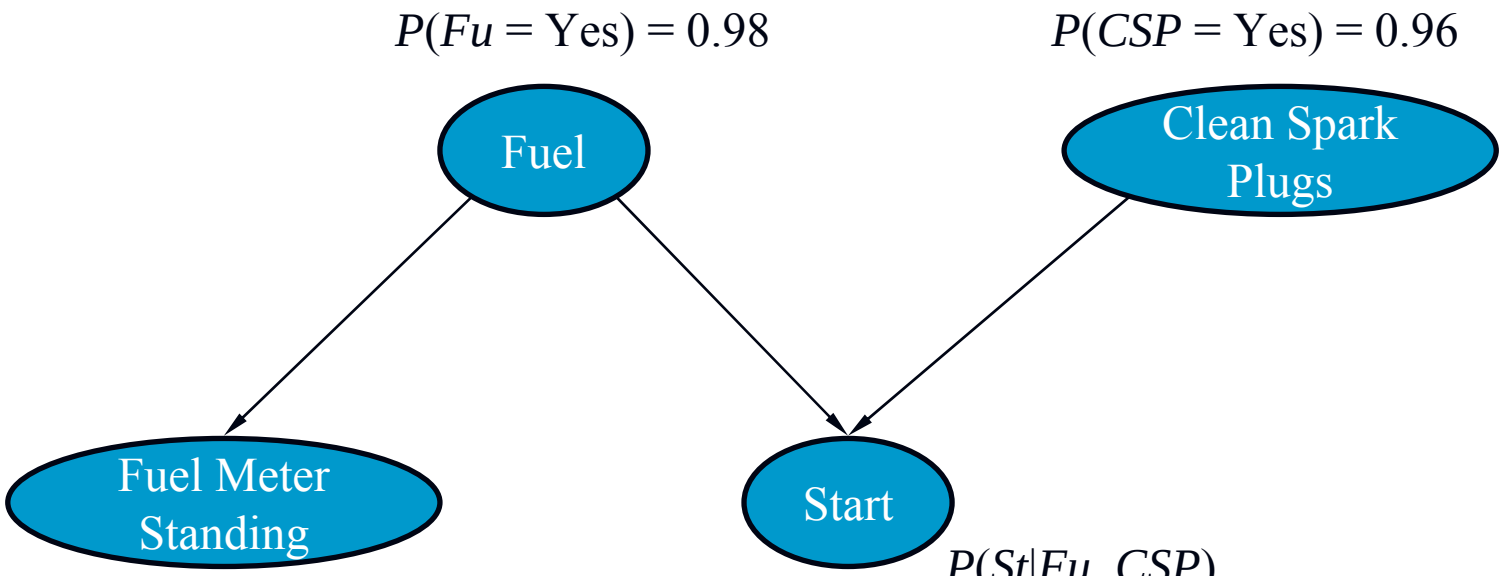
  - Chain Rule: a successive application of the product rule.

$$P(X_1, X_2, \ldots, X_n)$$
$$= P(X_1, X_2, \ldots, X_{n-1})P(X_n \mid X_1, X_2, \ldots, X_{n-1})$$
$$= P(X_1, X_2, \ldots, X_{n-2})P(X_{n-1} \mid X_1, X_2, \ldots, X_{n-2})P(X_n \mid X_1, X_2, \ldots, X_{n-1})$$
$$= \cdots$$
$$= P(X_1)\prod_{i=2}^{n} P(X_i \mid X_1, \ldots, X_{i-1})$$

Definition: Bayesian Networks

- A 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 variables.
  - The variables (vertices) 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 $P(X_i \mid \mathbf{Pa}(X_i))$.
    - Modeling for continuous variables is also possible.

Bayesian Network for the Car Start Problem



$P(FMS|Fu)$

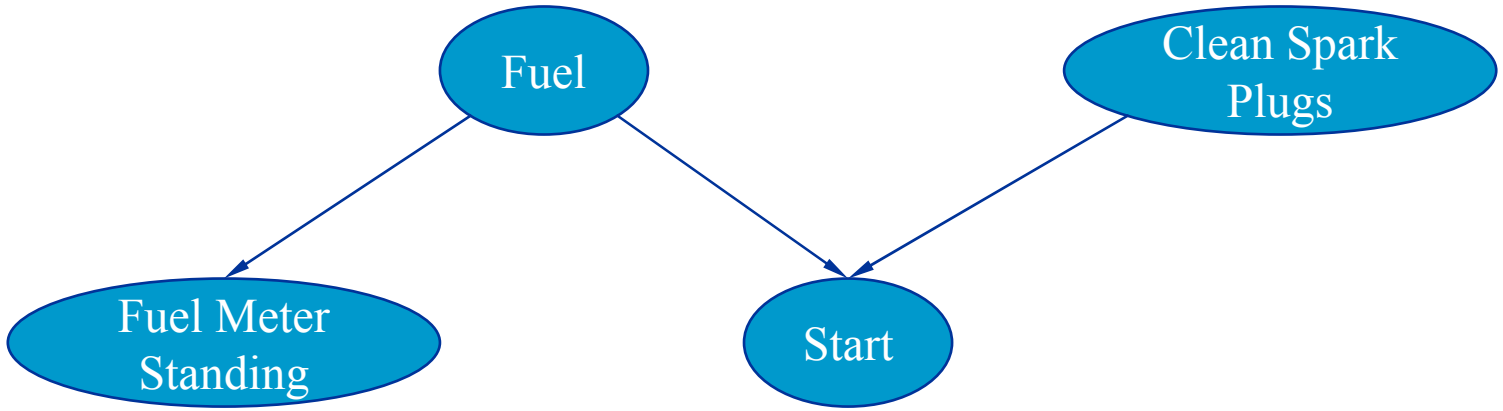
	FMS = Full	FMS = Half	FMS = Empty
Fu = Yes	0.39	0.60	0.01
Fu = No	0.001	0.001	0.998

$P(St|Fu, CSP)$

(*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

The Car Start Problem Revisited

1. No start → $P(St = \text{No}) = 1$ (evidence 1)
   - Update the conditional probabilities $P(Fu \mid St = \text{No})$, $P(CSP \mid St = \text{No})$, and $P(FMS \mid St = \text{No})$
2. Fuel meter stands for the half → $P(FMS = \text{Half}) = 1$ (evidence 2)
   - Update the conditional probabilities $P(Fu \mid St = \text{No}, FMS = \text{Half})$ and $P(CSP \mid St = \text{No}, FMS = \text{Half})$.



Calculation of Conditional Probabilities

- Calculation of $P(CSP \mid St = \text{No}, FMS = \text{Half})$ is as follows.

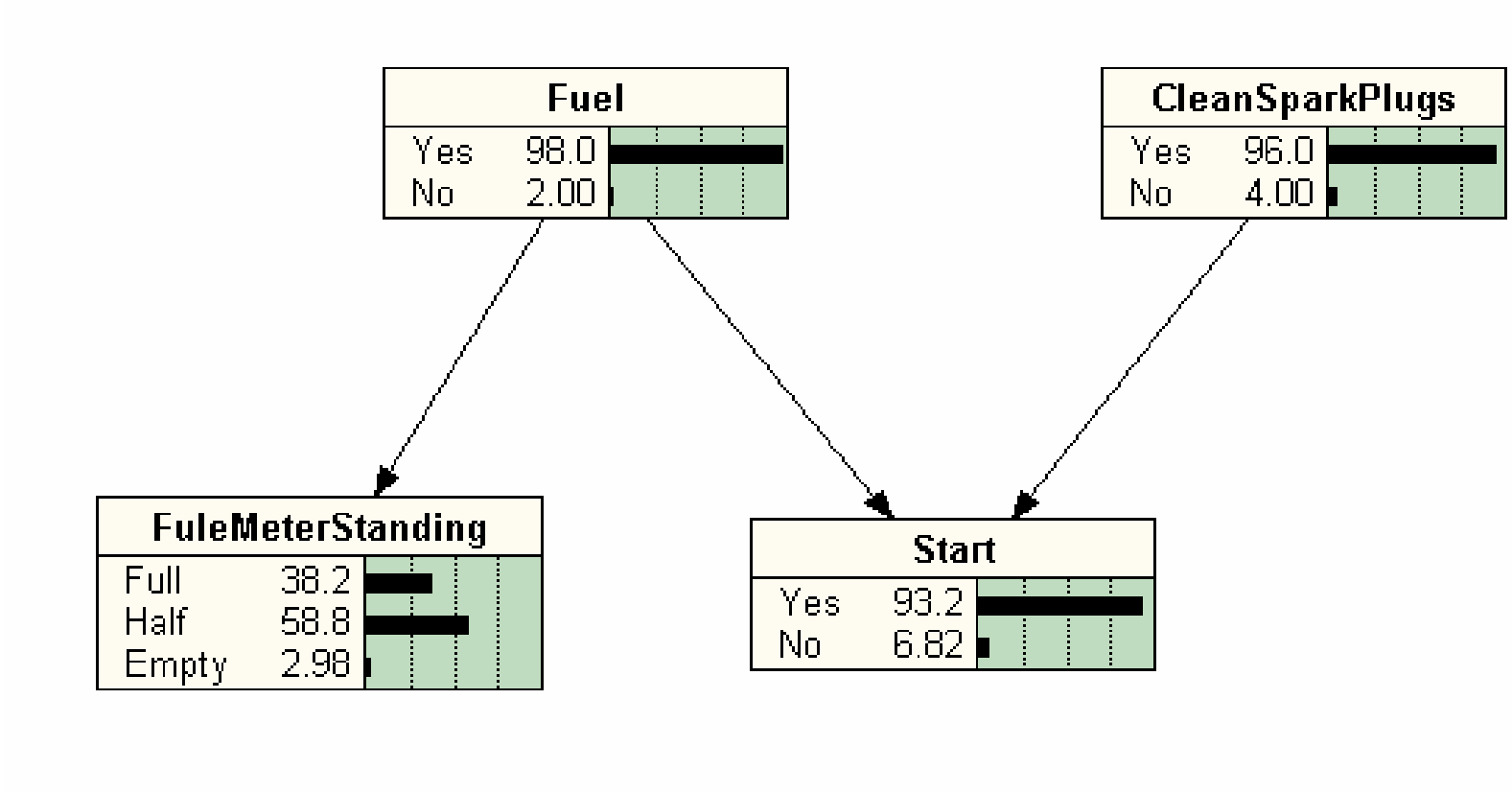
$$P(CSP \mid St, FMS) = \frac{P(CSP, St, FMS)}{P(St, FMS)}$$

$$= \frac{\sum_{Fu} P(Fu, CSP, St, FMS)}{\sum_{Fu,CSP} P(Fu, CSP, St, FMS)}$$

  - Summations in the above equation are taken over all possible values of the variables.
    - Calculation of the conditional probability by marginalization can be impossible.

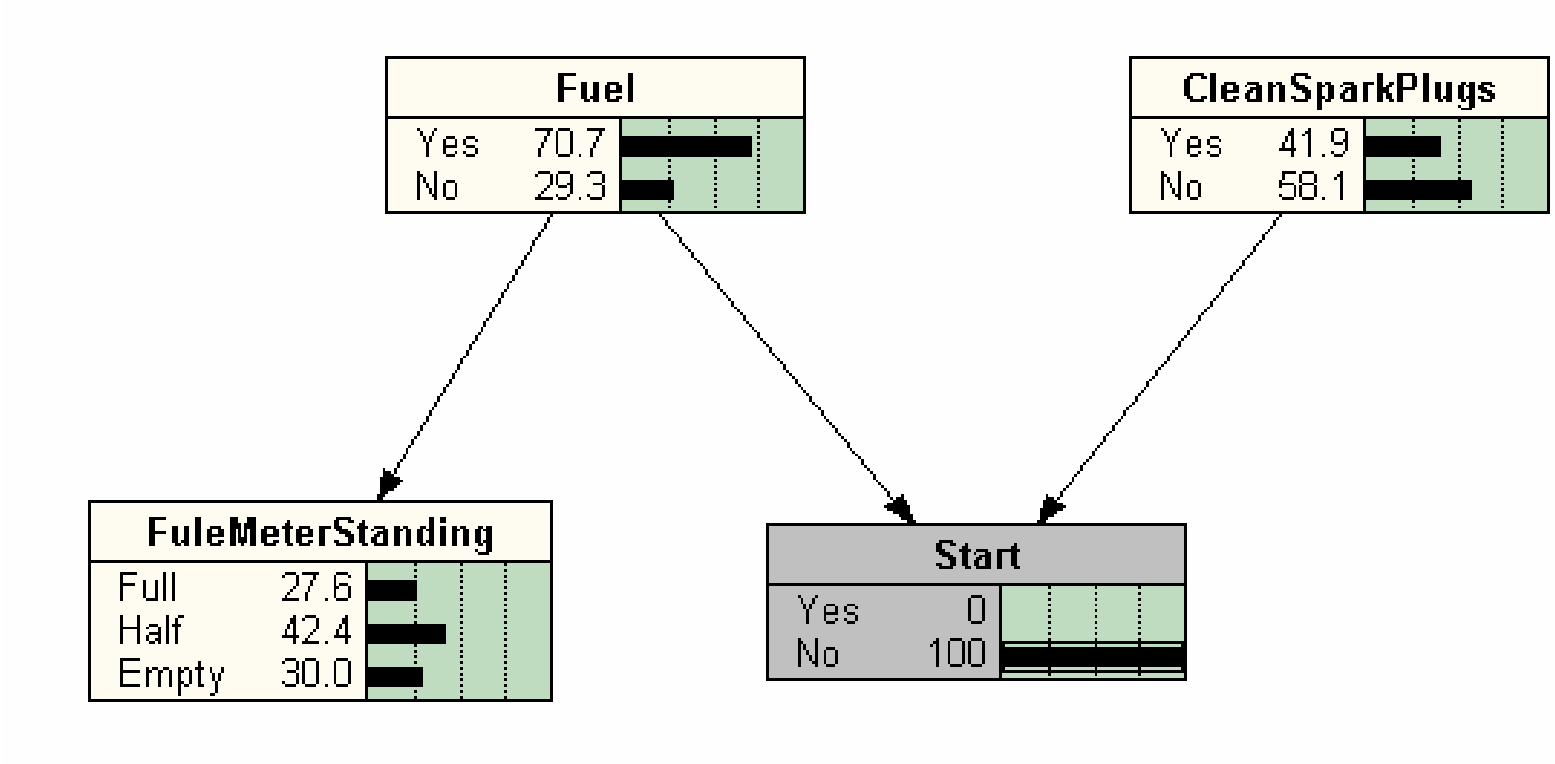
Initial State

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



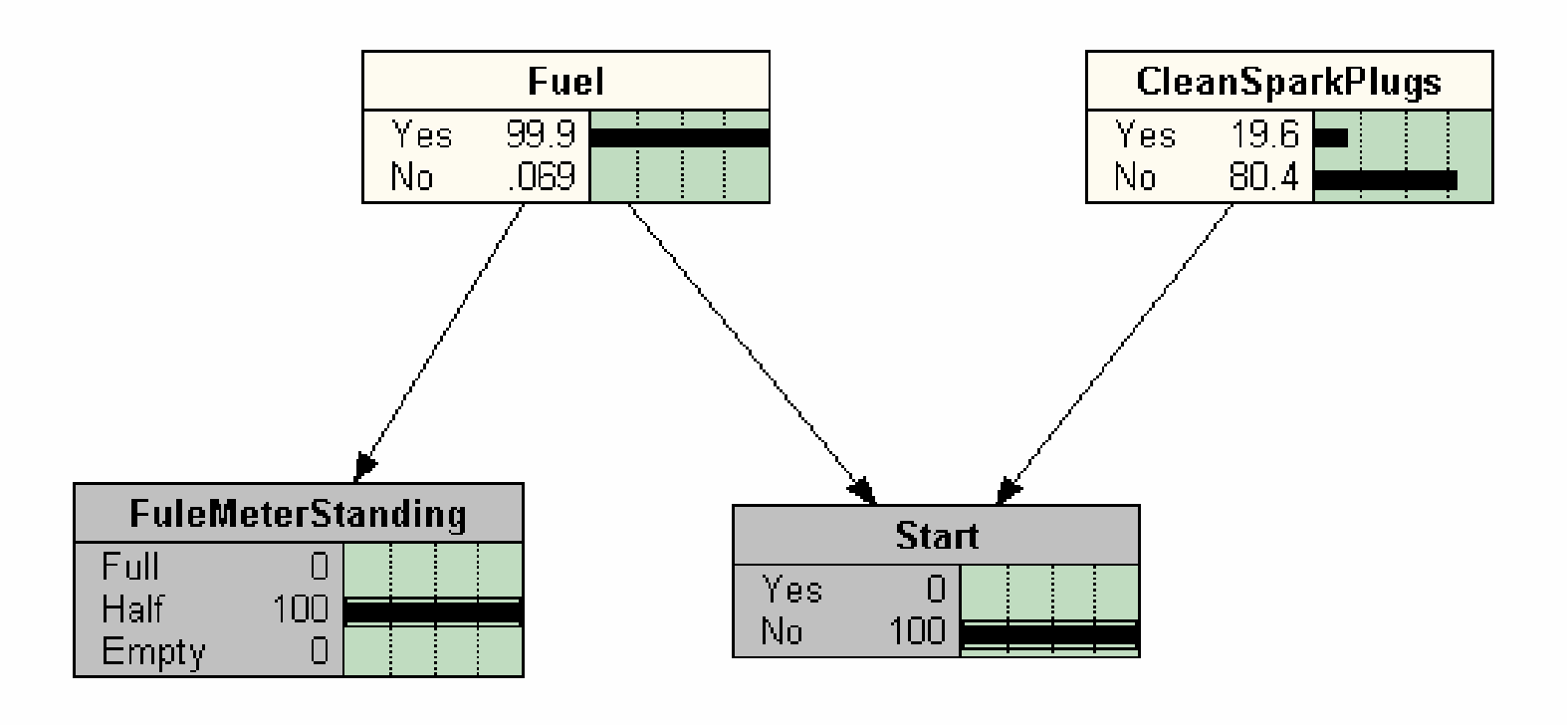
No Start

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



Fuel Meter Stands for Half

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



Bayesian Networks: Revisited

- Definition
  - A graphical model for the probabilistic relationships among a set of variables.
  - Compact representation of joint probability distributions on the basis of conditional probabilities.
  - Consists of the following.

Qualitative part

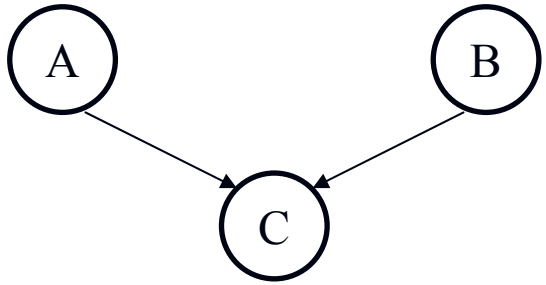
- A set of n variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ and a set of directed edges between variables.
- The variables (nodes) with the directed edges form a directed acyclic graph (DAG) structure.

Quantitative part

- To each variable X_i and its parents $\mathbf{Pa}(X_i)$, a conditional probability table for $P(X_i \mid \mathbf{Pa}(X_i))$.
  - Modeling for continuous variables is also possible.

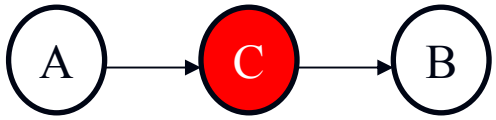
- Independence of two events
  - Marginal independence

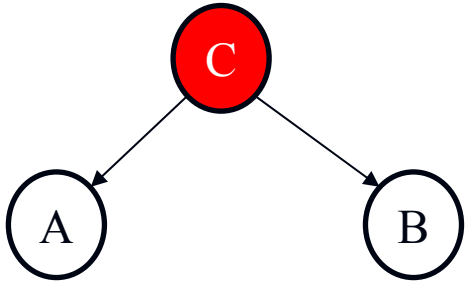
$$A \perp B \Leftrightarrow P(A \mid B) = P(A)$$
$$P(B \mid A) = P(B)$$



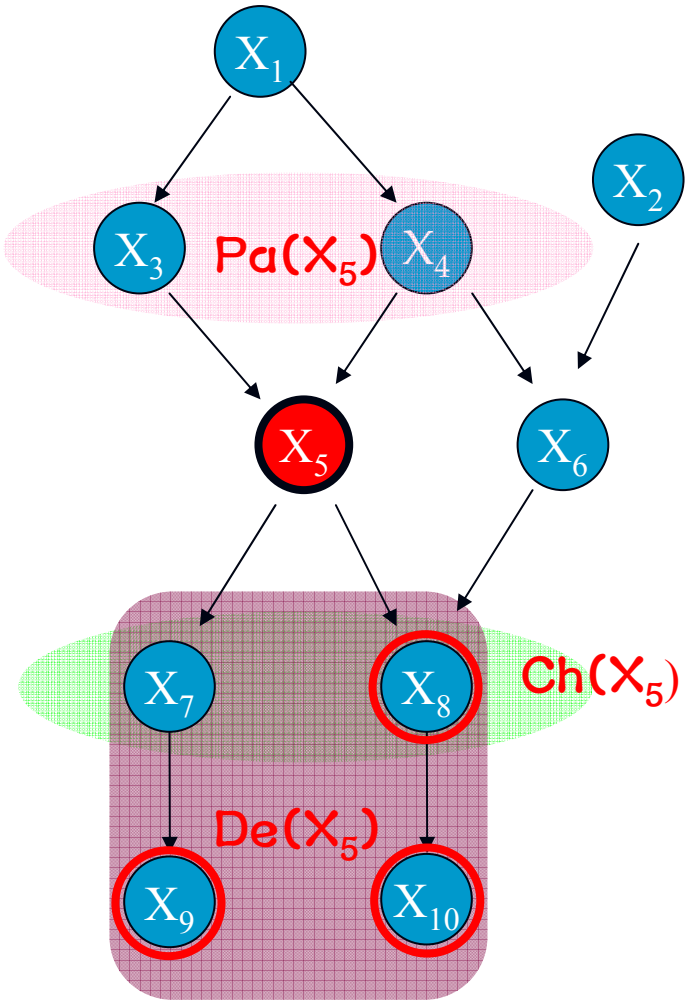
  - Conditional independence

$$A \perp B \mid C \Leftrightarrow P(A \mid B, C) = P(A \mid C),$$
$$P(B \mid A, C) = P(B \mid C)$$





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



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

$$P(\mathbf{X}\setminus\{X_{10}\}, X_{10}) = P(\mathbf{X}\setminus\{X_{10}\})P(X_{10} \mid \mathbf{X}\setminus\{X_{10}\})$$
$$= P(\mathbf{X}\setminus\{X_{10}\})P(X_{10} \mid X_8)$$
$$P(\mathbf{X}'\setminus\{X_9\}, X_9) = P(\mathbf{X}'\setminus\{X_9\})P(X_9 \mid \mathbf{X}'\setminus\{X_9\})$$
$$= P(\mathbf{X}'\setminus\{X_9\})P(X_9 \mid X_7)$$
$$P(\mathbf{X}''\setminus\{X_8\}, X_8) = P(\mathbf{X}''\setminus\{X_8\})P(X_8 \mid \mathbf{X}''\setminus\{X_8\})$$
$$= P(\mathbf{X}''\setminus\{X_8\})P(X_8 \mid X_5, X_6)$$

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

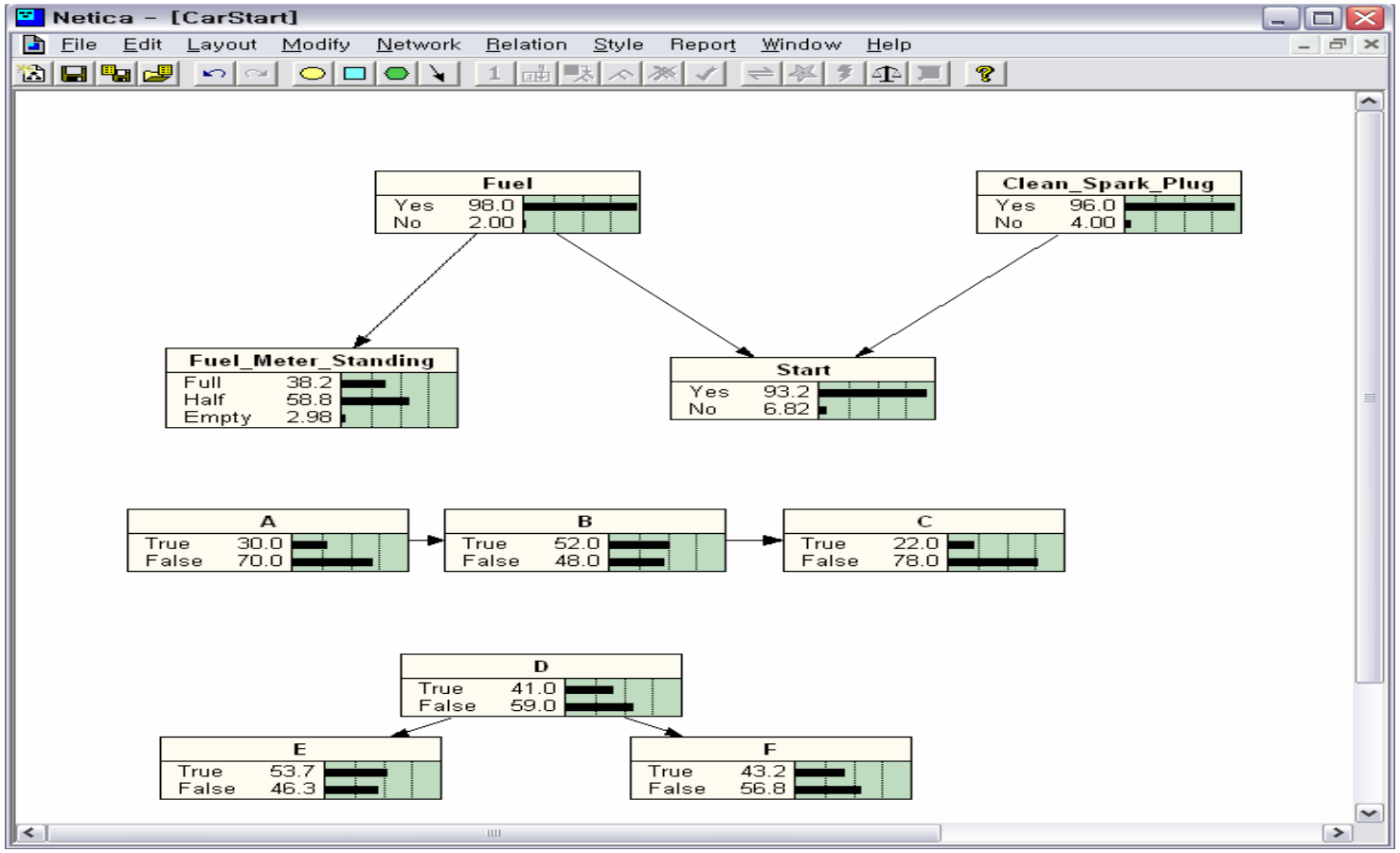
Bayesian Network Represents the Joint Probability Distribution

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

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

  - Given the joint probability distribution, any conditional probability can be calculated in principle.

An Illustration of Conditional Independence in BNs



Netica (http://www.norsys.com)

Inference Example

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

$P(X_2 \mid X_1) =$

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

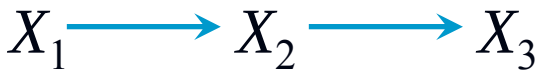
$X_1 == 1$: (0.5, 0.5)

$P(X_3 \mid X_2) =$

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

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

Initial State



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

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

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

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

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

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

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

$P(X_2 \mid X_1) =$

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

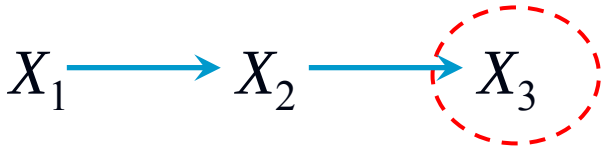
$X_1 == 1$: (0.5, 0.5)

$P(X_3 \mid X_2) =$

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

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

Given that $X_3 == 1$



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

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

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

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

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

$= \beta$ (0.6, 0.4) * (0.62, 0.5)

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

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

$P(X_2 \mid X_1)$ =

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

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

$P(X_3 \mid X_2)$ =

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

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

Learning Example

Data (22 examples):

X_1: 0 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0 0 1 0 1 1 0

X_2: 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 1 0 1 0 1 1 1

X_3: 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 0

10 20

Parameter Learning

$P(X_1) = (13/22, 9/22)$

$P(X_2 \mid X_1) =$

$X_1 == 0$: $(7/13, 6/13)$

$X_1 == 1$: $(3/9, 6/9)$

Data (22 examples):

X_1: 0 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0 0 1 0 1 1 0

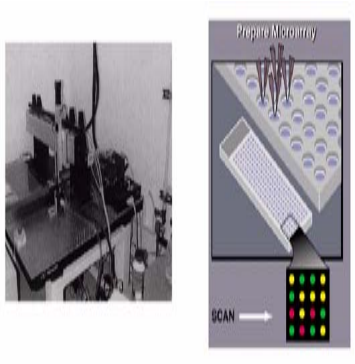
X_2: 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 1 0 1 0 1 1 1

X_3: 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 0 0

- Introduction
- Basic Concepts of Bayesian Networks
- Learning Bayesian Networks
  - Parameter Learning
  - Structural Learning
- Applications
  - DNA microarray
  - Classification
  - Dependency Analysis
- Summary

Learning Bayesian Networks

Data Acquisition



Preprocessing

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	-0.69504	-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 lig	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 = Home	-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

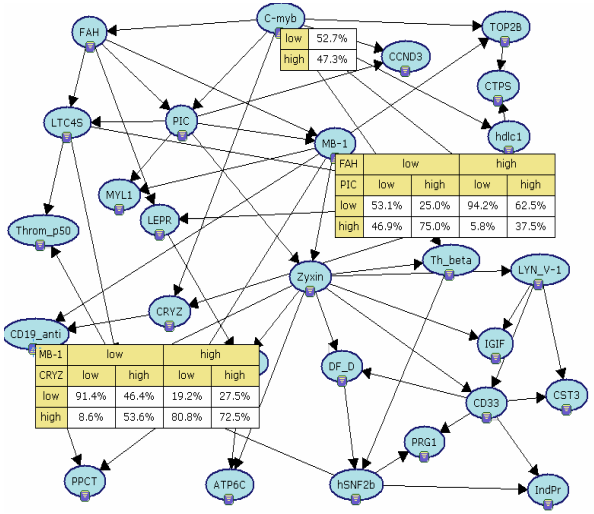
Type	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL
C-myb gen	high	low	high	high	high	high	high	high
FAH Fumar	low	high	high	low	low	low	low	high
PROTEASO	low	low	high	high	high	high	high	low
Leukotriene	low	low	low	low	low	low	low	high
MB-1 gene	high	low	low	high	high	low	high	high
Zyxin	low	low	low	low	low	low	low	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

Prior knowledge

Bayesian network learning

* Structure Search
* Score Metric
* Parameter Learning

BN = Structure + Local probability distribution



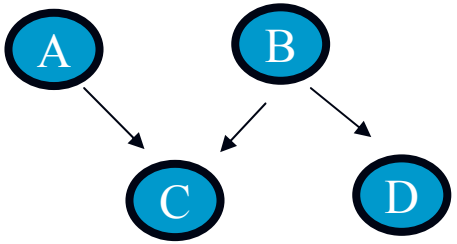
Learning Bayesian Networks (Cont'd)

- 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

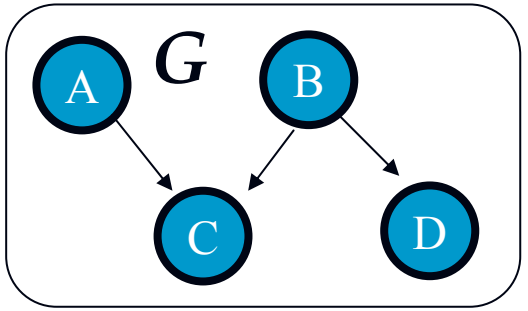
Key point: independence of parameter estimation

- 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)*.

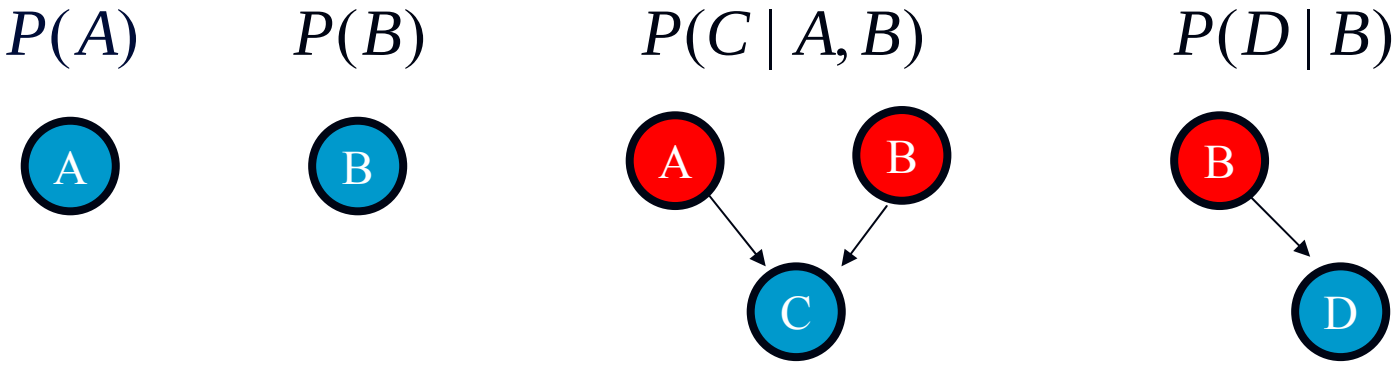
$$L_G(\Theta; D) = P_G(D \mid \Theta) = \prod_{i=1}^{M} P_G(\mathbf{s}_i \mid \Theta)$$

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

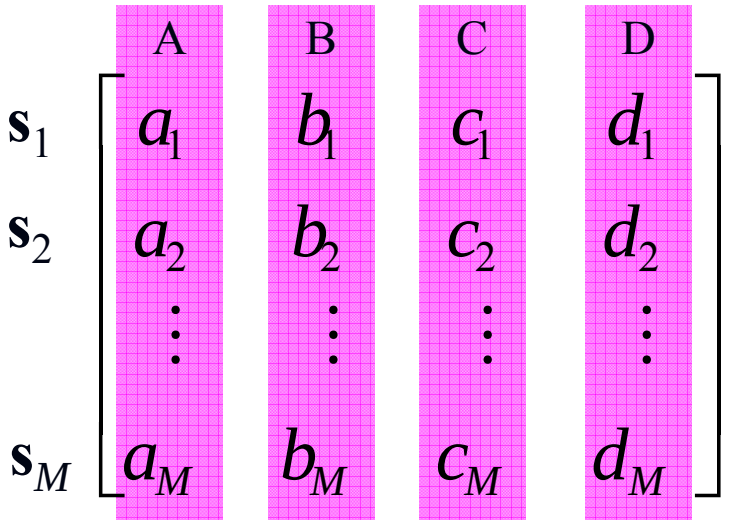
$$= \left(\prod_{i=1}^{M} P_G(a_i|\Theta) \right) \left(\prod_{i=1}^{M} P_G(b_i|\Theta) \right) \left(\prod_{i=1}^{M} P_G(c_i|a_i,b_i,\Theta) \right) \left(\prod_{i=1}^{M} P_G(d_i|b_i,\Theta) \right)$$



Independent parameter estimation for each node (variable)



- One can estimate the parameters for $P(A)$, $P(B)$, $P(C \mid A, B)$, and $P(D \mid B)$ 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).



$$P(A,B,C,D)=P(A)\times P(B)\times P(C|A,B)\times P(D|B)$$

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.

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

posterior

prior

likelihood

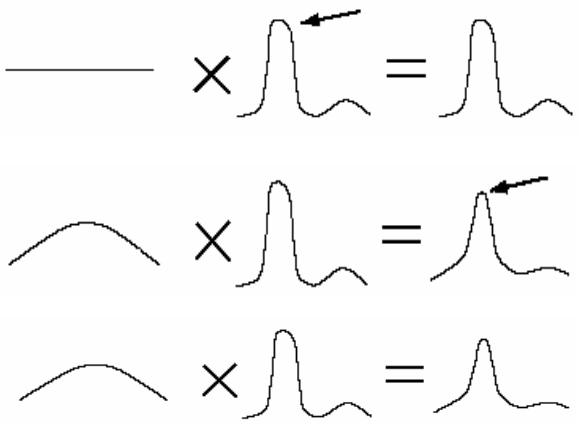
Bayes Rule, MAP and ML

- Bayes' rule

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

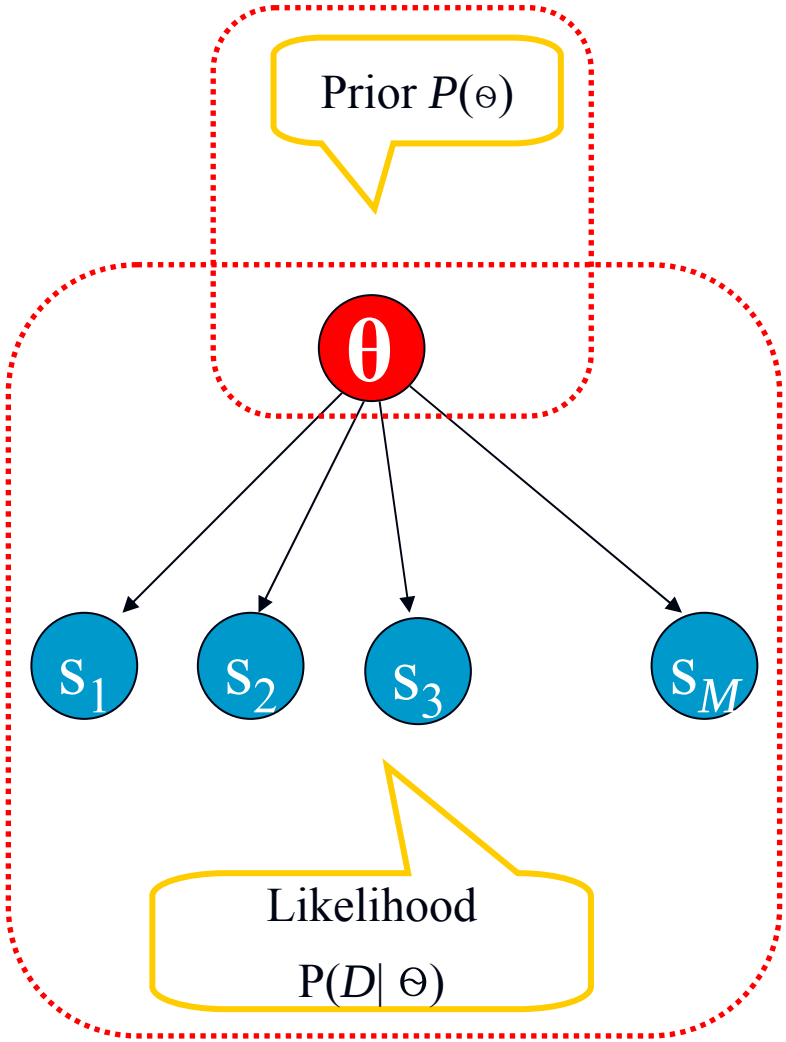
h: hypothesis (models or parameters)
D: data

  - **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 Ghahramani, Z.

Bayesian estimation (for multinomial distribution)



Prior knowledge or pseudo counts

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

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

Sufficient statistics

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

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

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

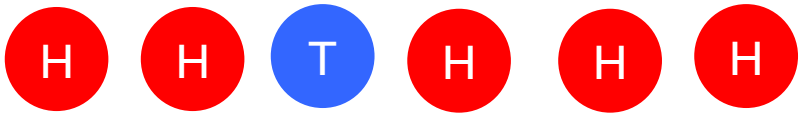
$$= \int \theta_k P(\boldsymbol{\theta} \mid D)d\boldsymbol{\theta}$$

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

Smoothed version of MLE

An Example: Coin toss

Maximum likelihood estimation

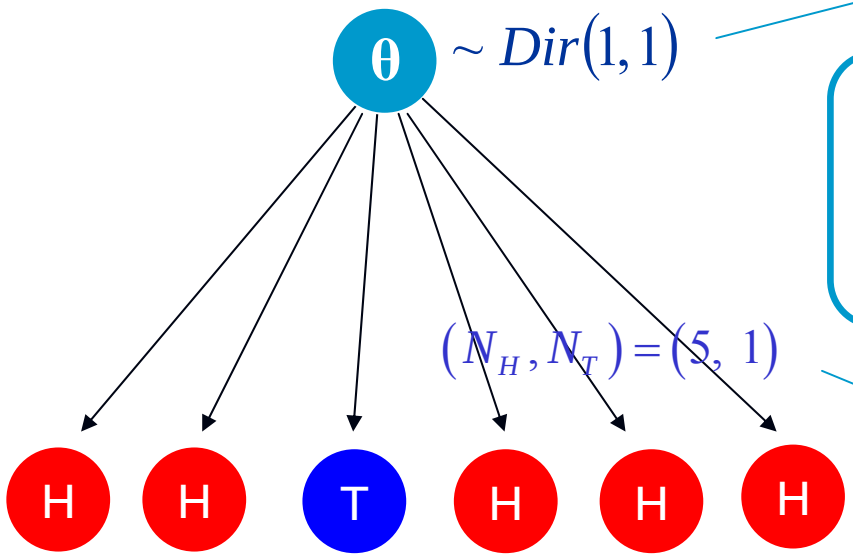


$$P(D \mid \boldsymbol{\theta}) = \theta_H \theta_T \theta_H \theta_H \theta_H \theta_H = \theta_H^5 \theta_T^1$$

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

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

Bayesian inference



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

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

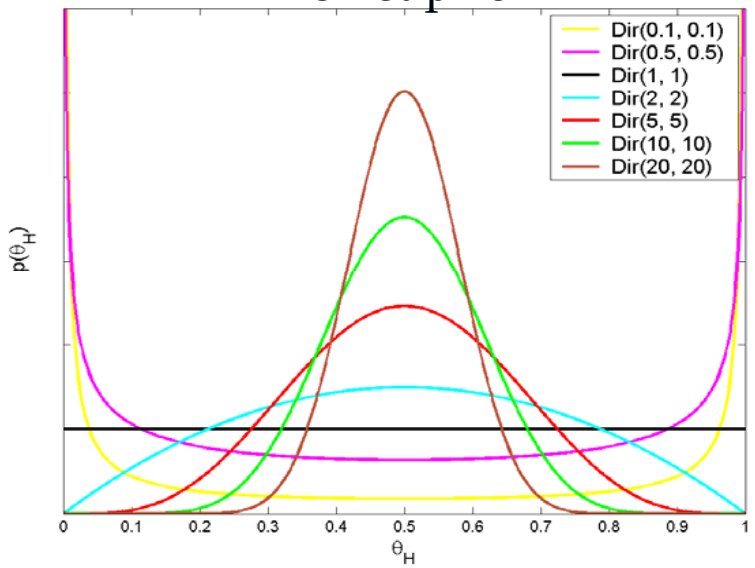
$$P(\mathrm{H} \mid D) = \frac{1+5}{(1+5)+(1+1)} = \frac{3}{4} = 0.75$$

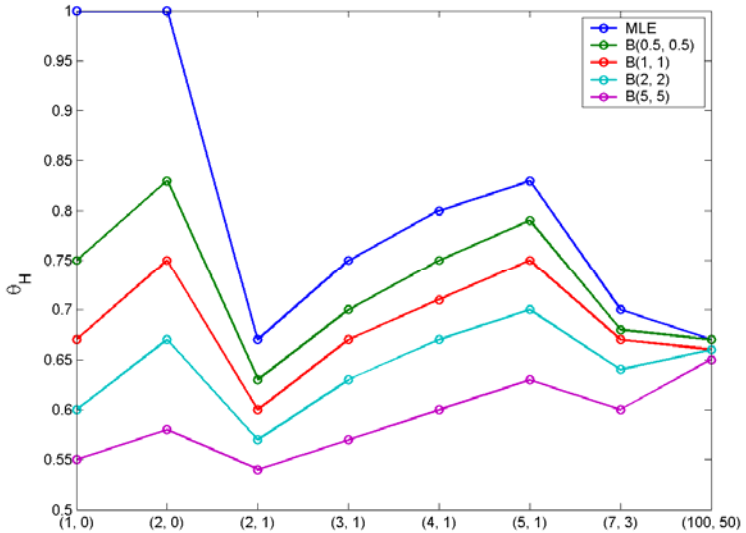
$$P(D \mid \boldsymbol{\theta}) = \theta_H \theta_T \theta_H \theta_H \theta_H \theta_H = \theta_H^5 \theta_T^1$$

P(H)

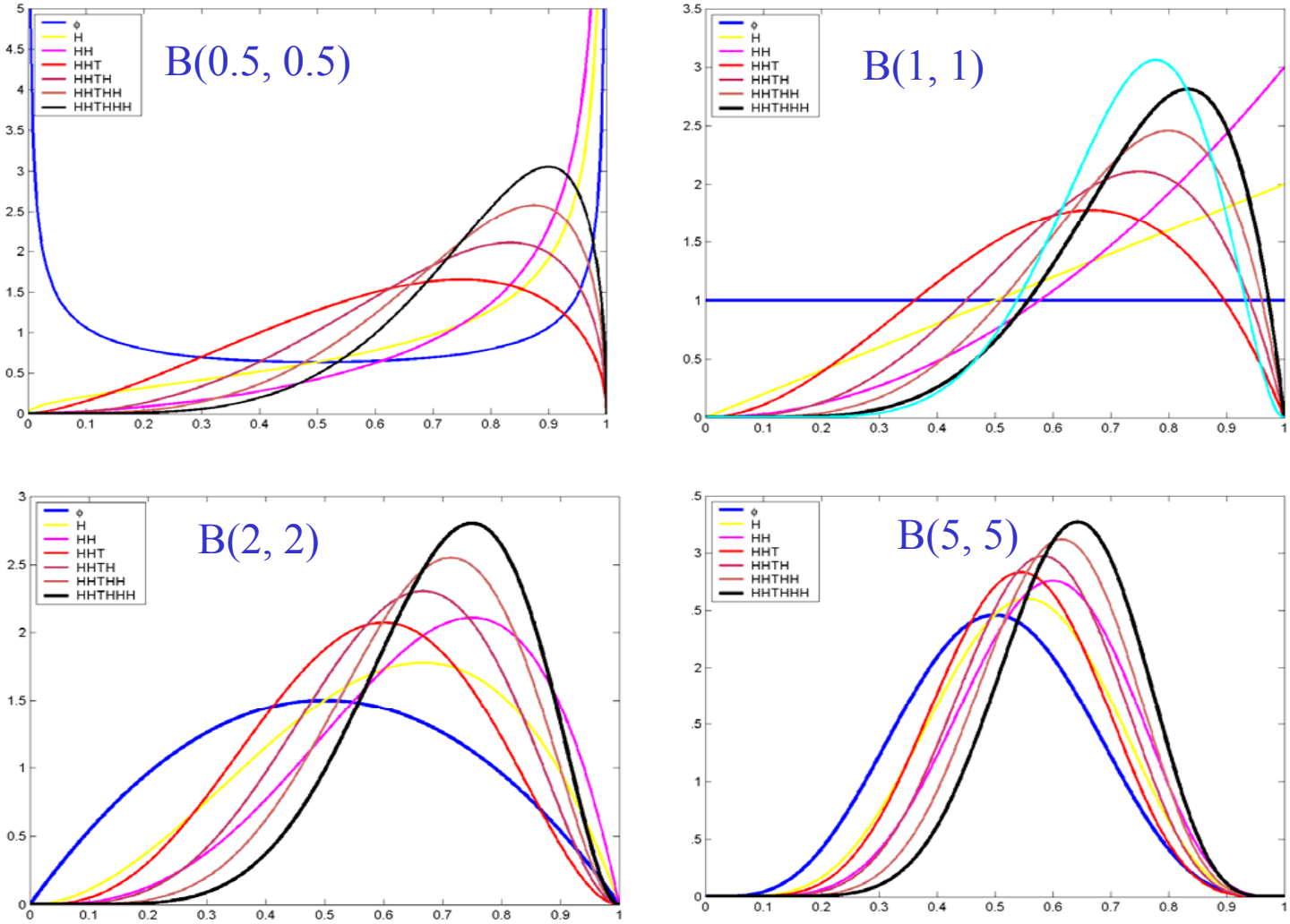
	H	HH	HHT	HHTH	HHTHH	HHTHH H	HHTHH HTTHH	(100, 50)
MLE	1.00	1.00	0.67	0.75	0.80	0.83	0.70	0.67
B(0.5, 0.5)	0.75	0.83	0.63	0.70	0.75	0.79	0.68	0.67
B(1, 1)	0.67	0.75	0.60	0.67	0.71	0.75	0.67	0.66
B(2, 2)	0.60	0.67	0.57	0.63	0.67	0.70	0.64	0.66
B(5, 5)	0.55	0.58	0.54	0.57	0.60	0.63	0.60	0.65

Dirichlet prior





Variation of posterior distribution for the parameter



- Introduction
- Basic Concepts of Bayesian Networks
- Learning Bayesian Networks
  - Parameter Learning
  - Structural Learning
    - Scoring Metric
    - Search Strategy
- Practical Applications
  - DNA microarray
  - Classification
  - Dependency Analysis
- Summary

Structural Learning

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

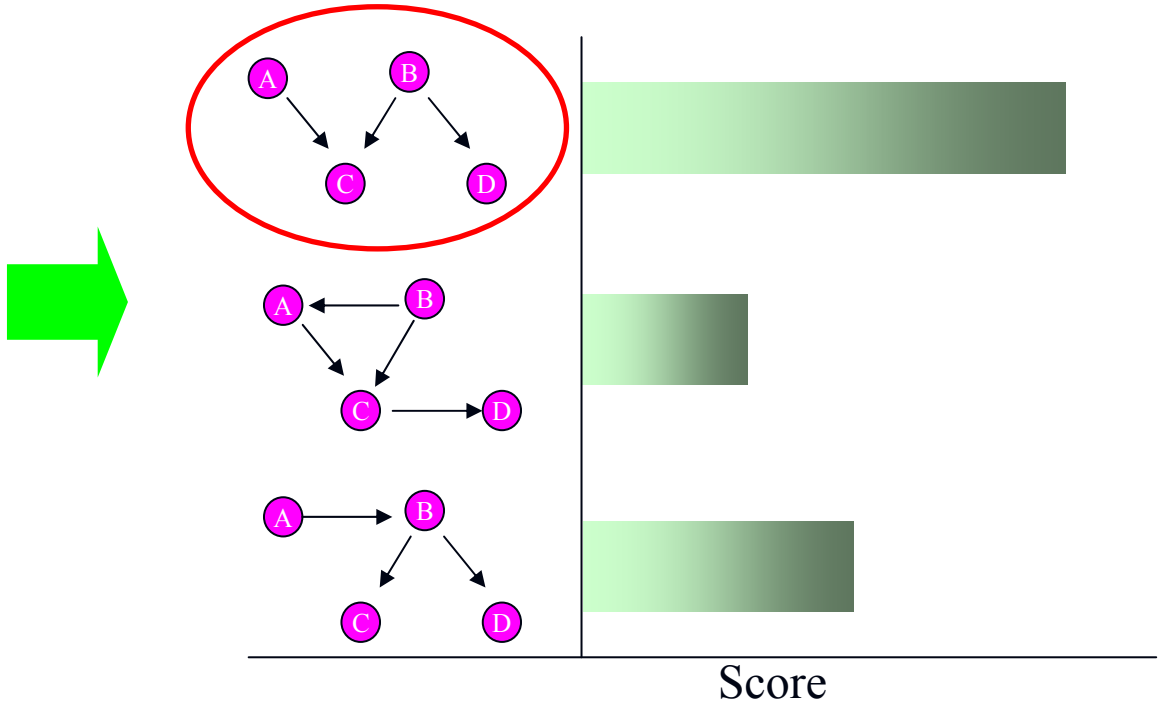
  Metric (score)-based approach

  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, $I(X;Y) \leq I(X;\{Y,Z\})$, the fully connected network is obtained in an unrestricted case.
  - Prone to overfitting.

Likelihood Score in Relation with Information Theory

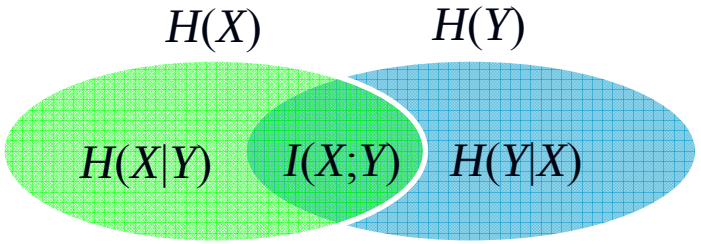
$$\log L(\hat{\Theta}; \mathrm{D}) = \sum_{i=1}^{N} \sum_{j=1}^{|\mathbf{Pa}_i|} \sum_{k=1}^{|X_i|} N_{ijk} \log \frac{N_{ijk}}{N_{ij}}$$

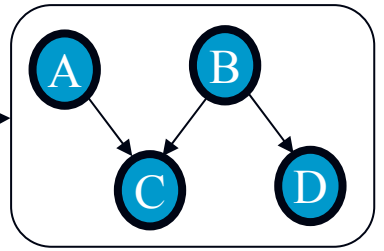
$$= M \sum_{i=1}^{N} \sum_{j=1}^{|\mathbf{Pa}_i|} \sum_{k=1}^{|X_i|} \frac{N_{ijk}}{M} \log \frac{N_{ijk}}{N_{ij}}$$

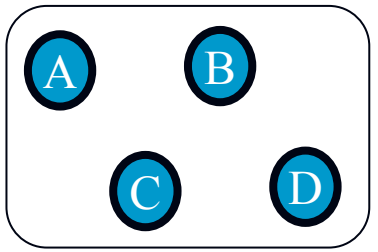
$$= -M \sum_{i=1}^{N} H(X_i \mid \mathbf{Pa}_i)$$

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

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







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 $(X_i;\mathbf{Pa}(X_i))$

Bayesian Dirichlet Score

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

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

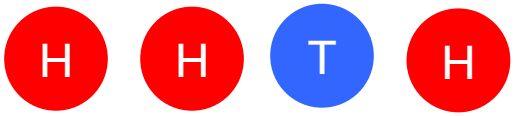
$$\theta_{ij} \sim Dir\left(\alpha_{ij1},\alpha_{ij2},\ldots,\alpha_{ij|X_i|}\right) \qquad \alpha_{ij} = \sum_{k=1}^{|X_i|}\alpha_{ijk}$$

$$N_{ijk} = \#\ \text{of}\ \left(X_i = x_i^k, \mathbf{Pa}(X_i) = pa_i^j\right) \qquad N_{ij} = \sum_{k=1}^{|X_i|}N_{ijk}$$

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

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

Bayesian Dirichlet Score (Cont'd)



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

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

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

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

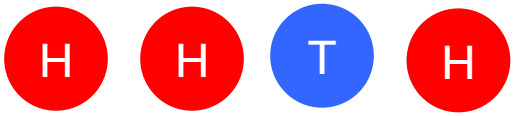
$$P(H \mid HHT) = \frac{(\alpha_H + 2)}{(\alpha_H + 2) + (\alpha_T + 1)}$$

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

$$\frac{[\alpha_H \times (\alpha_H + 1) \times \cdots (\alpha_H + 2)] \times \alpha_T}{\alpha \times (\alpha + 1) \times \cdots (\alpha + 3)} \qquad \frac{\Gamma(\alpha + 4)}{\Gamma(\alpha)}$$

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

Bayesian Dirichlet Score (Cont'd)



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

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

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

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

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

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

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

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

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

Bayesian Dirichlet Score (Cont'd)

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

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

- $\log P(D \mid G)$ in Bayesian score is asymptotically equivalent to BIC (Schwarz, 1978) and minus the MDL criterion (Rissanen, 1987).

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

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

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 $(X_i, \mathbf{Pa}(X_i))$.

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

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

Tree-Structured Network

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

Improvement over empty network

Score for empty network

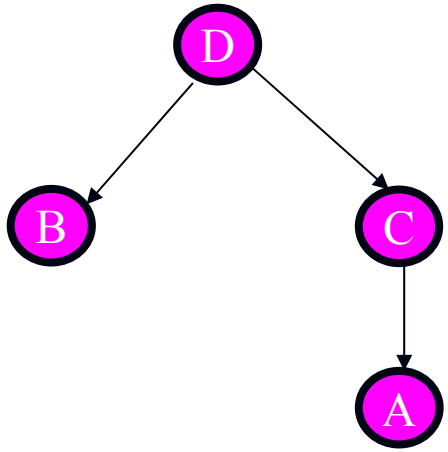
$$Score(G \mid D) = \sum_i Score(X_i \mid Pa_i) = \sum_i [Score(X_i \mid Pa_i) - Score(X_i)] + \sum_i Score(X_i)$$

Chow and Liu (1968)

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

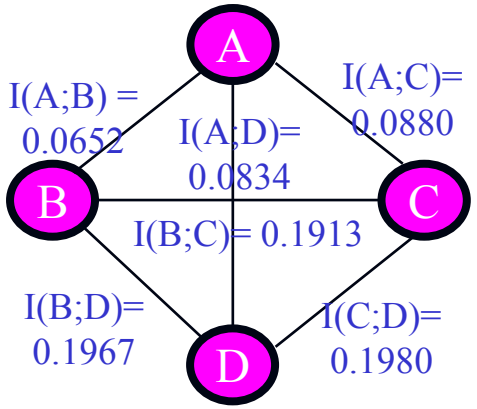
Build a *maximum weighted spanning tree*.

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

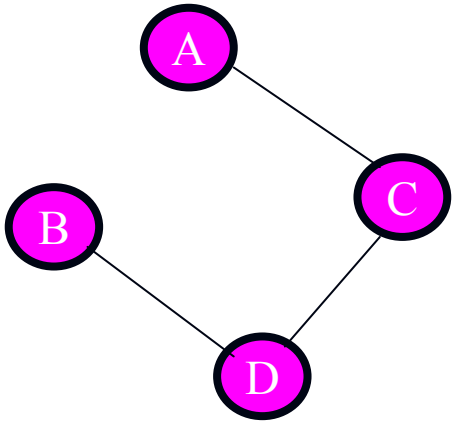


	A	B	C	D
S1	H	L	L	L
S2	H	H	H	H
	...	..	...	...
S_M	L	H	H	L

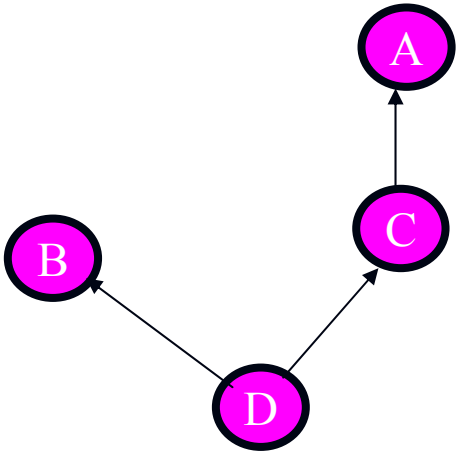
Complete undirected graph



Maximum spanning tree



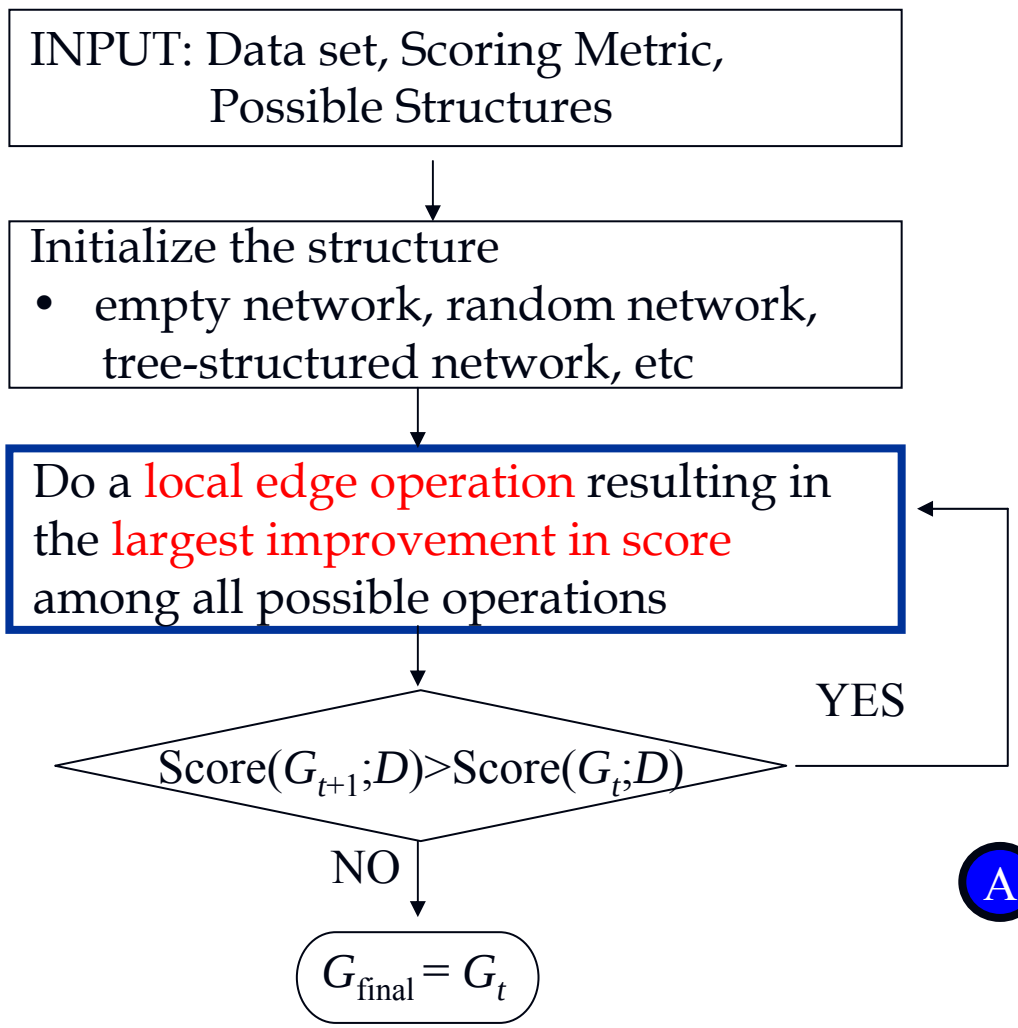
Directed tree

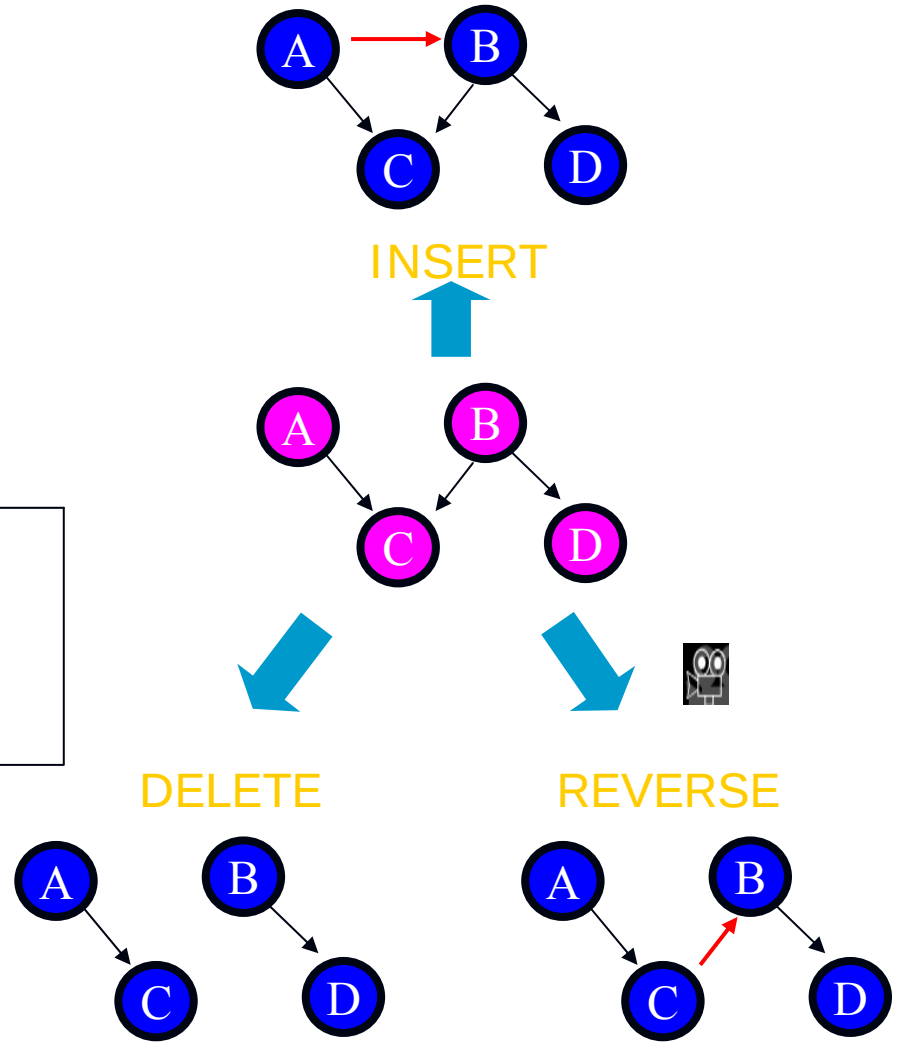


Search Strategies for General Bayesian Networks

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

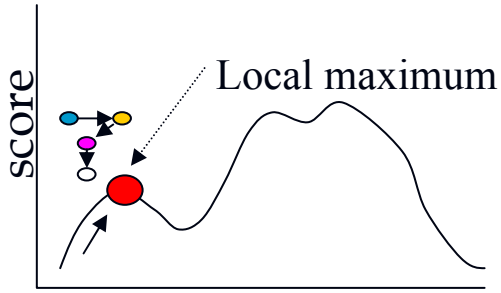
Greedy Local Search Algorithm



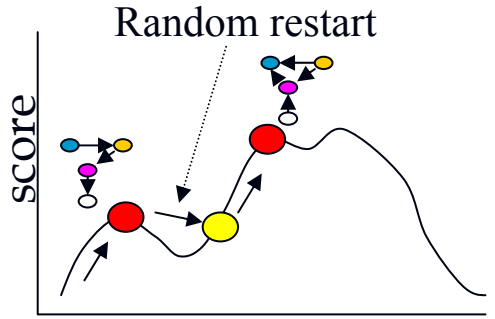


Enhanced Search

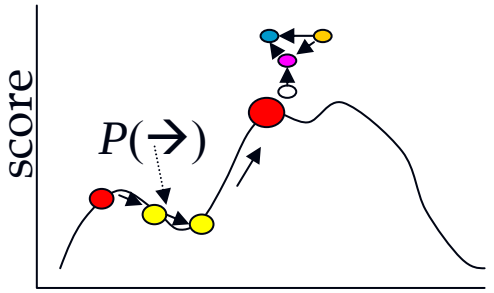
- Greedy local search can get stuck in local maxima or plateaux.
- Standard heuristics to escape the two includes
  - Search with random restarts, Simulated annealing, Tabu search
- Genetic algorithm: a population-based search.



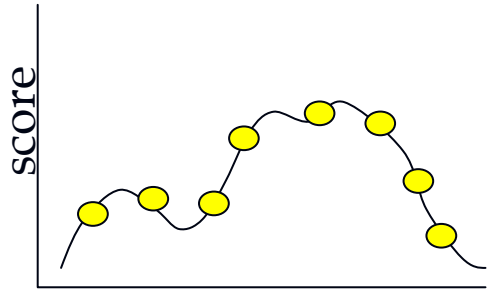
Greedy search



Greedy search with random restarts



Simulated annealing



Population-based search (GA)

Learning Bayesian Networks: Summary

- Learning Bayesian networks
  - Joint (probability) distribution ➔ product of conditional probabilities for each variable (node) (*sum* in log-based representation.)
- Parameter Learning
  - Estimation of local probability distribution
  - MLE, Bayesian estimation.
- Structure Learning
  - Score metrics
    - Likelihood, Bayesian score, BIC, MDL
  - Search strategies
    - Optimization in discrete space ➔ maximize the score
    - Key concept: the decomposability of the score
    - Tree-structured and general Bayesian networks.

- Introduction
- Basic Concepts of Bayesian Networks
- Learning Bayesian Networks
  - Parameter Learning
  - Structural Learning
- Applications
  - DNA microarray
  - Classification (Naïve Bayes, TAN)
  - Dependency Analysis
- Summary

DNA Microarray Data Analysis

- Classification
  - Gene expression data of 72 leukemia patients.
  - Task: classification of samples into AML or ALL based on expression patterns using Bayesian networks.
- Combined analysis of gene expression data and drug activity data
  - Gene expression data and drug activity data of 60 cancer samples
  - Task: construct a dependency network of genes and drugs.
- Tools
  - WEKA (http://www.cs.waikato.ac.nz/~ml/**weka**/)
    - A collection of machine learning algorithms for data mining tasks (implemented in JAVA, open source software issued under the GNU General Public License)
    - Bayesian network learning algorithms are also included.
  - BNJ software (http://bnj.sourceforge.net/) are used for visualization when needed.

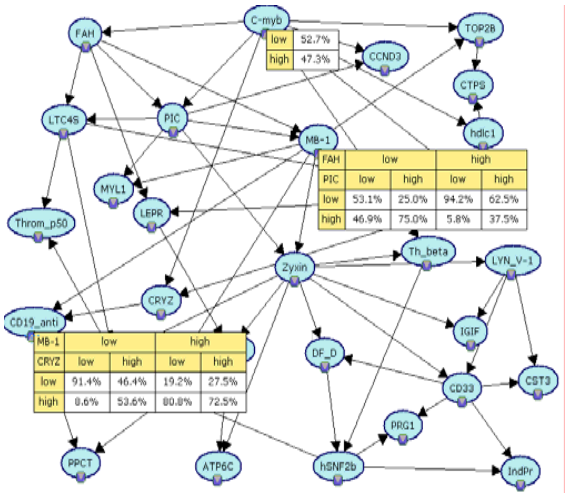
DNA Microarrays

- Recent developments in the technology for biological experiments have made it possible to produce massive biological data sets.

- Monitor thousands of gene expression levels simultaneously. ←→ traditional one gene experiments.
  - parallel view of the expression patterns of thousands of genes in a cell.
  - Bayesian networks is a useful tool for the identification of a variety of meaningful relationships among genes from the data.

Bayesian Networks in Microarray Data Analysis

Expression profiles



Bayesian network construction

Sample classification

- Disease diagnosis

Gene-gene relation analysis

- Activation or inhibition between genes

Gene regulatory network analysis

- Global view on the relations among genes

Combined analysis of other biological data

- DNA sequence data, drug activity data, and so on.

DNA Microarray

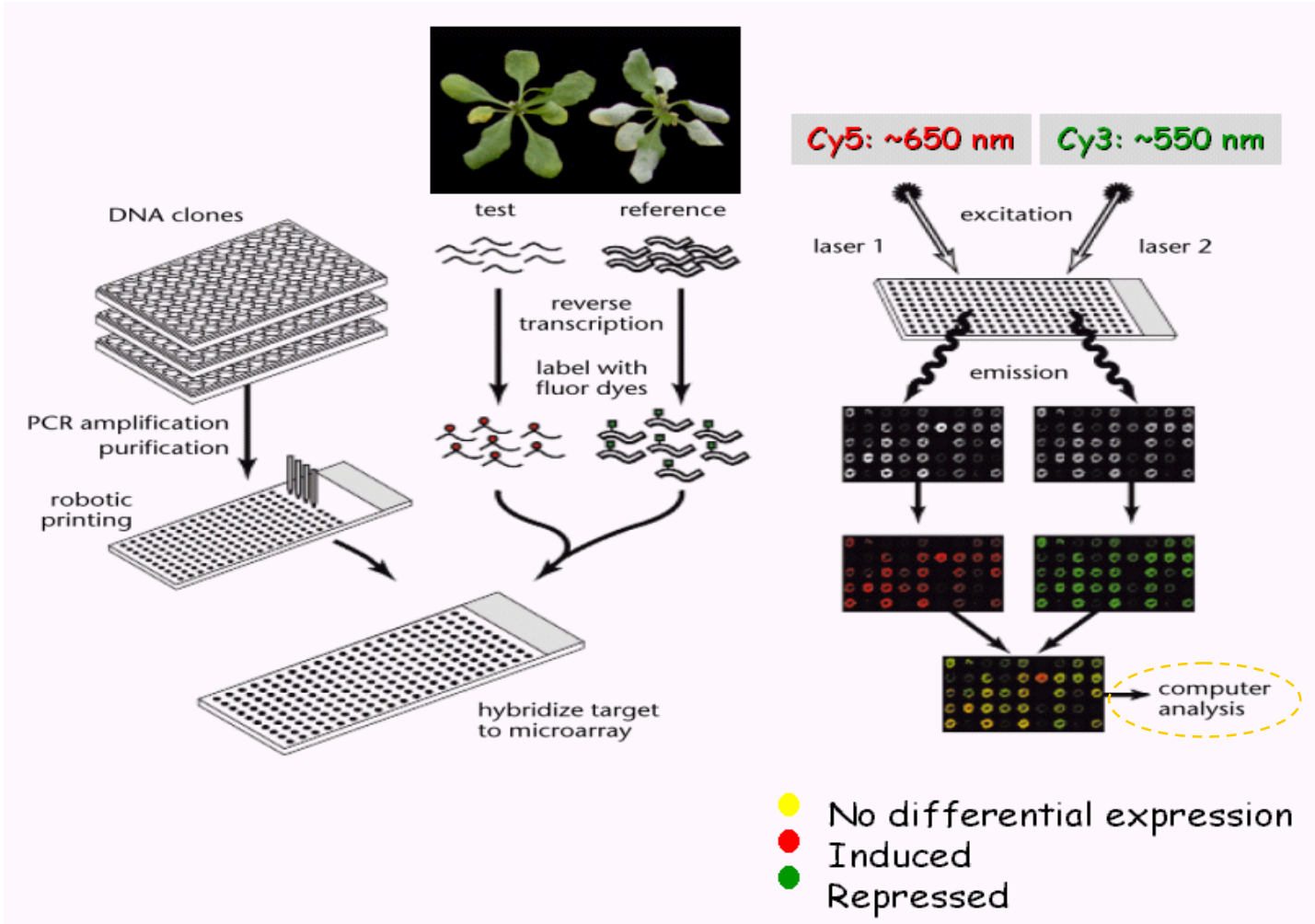
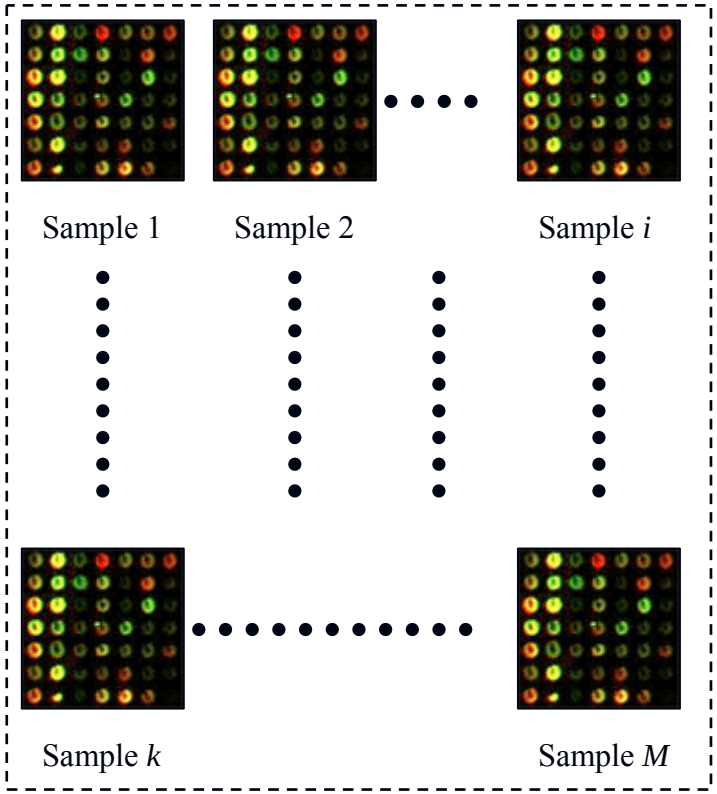


Image analysis

Data Preparation for Data Mining



Microarray image samples

Image analysis

Sample 1

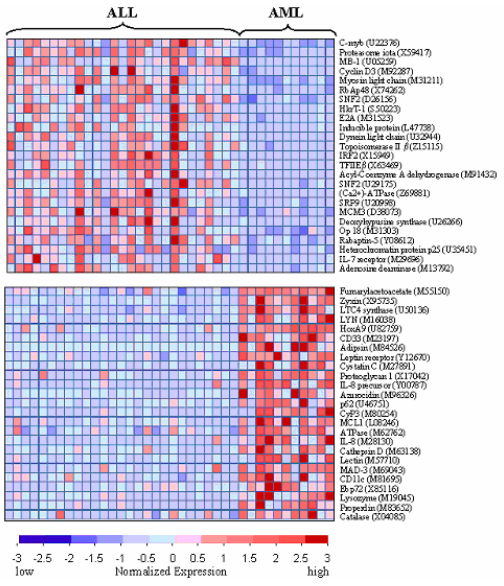
Gene 2

Name	Type	CNS:SNB-	CNS:U251	BR:BT-54	CNS:SF-2	CNS:SF-2
Human GDI	est	-0,93	0,1	-0,1	0,27	0,27
LBR Lamin	est	0,18	-0,27	-0,21	-0,99	-0,26
SID W 245	est	0,55	-0,06	-0,5	0,53	-0,42
H,sapiens M	est	0,35	-0,06	0,03	-0,19	0,37
Activated ra	protein	-0,62	-0,62	-0,62	-0,62	-0,62
PRG1 Hem	est	-0,95	-0,75	-0,72	0,39	1,44
SID 48904	est	-0,67	-0,29	-0,99	0,13	1,05
SID 31084	est	-0,95	-0,49	-1,48	-0,46	0,34
Human mRN	est	-0,11	-0,19	-1,26	-1,12	-2,38
ESTs Chr,6	est	-0,67	-0,56	0,92	0,37	0,37
SID 37153,	est	-2,01	-1,61	-0,2	-0,45	-2,67
SID W 510	est	-0,33	-0,01	-1,05	-0,31	-0,99
MYC V-myc	est	-1,73	-1,34	0,79	-1,39	-1,22
SID 36323	est	0,63	-0,81	-0,44	0,11	-2,3
EUKARYOT	est	-2,28	-0,55	-0,63	-1,26	-1,97
SID W 276	est	0,73	1,31	-0,71	-0,31	-0,79
SID 47045	est	0,19	-0,28	-0,19	-1	-0,45
SID W 135	est	0,02	-0,25	0,33	-0,01	0,28
ESTs Chr,1	est	-0,23	-0,4	-0,71	0,75	-0,66
SID 25714	est	0,51	0,03	-1,06	0,35	0,7
Homo sapie	est	-0,61	0,63	-1,21	0,04	0,99
Human G/T	est	-0,16	1,14	0,65	-0,53	-0,81
SID 28578	est	0,81	-0,27	0,37	-0,28	0,73
Human cycl	est	0,85	0,3	0,91	-1,22	1,8
SID 47000	est	0,69	0,52	-1,44	-0,39	0,52
*Brain-expr	est	0,73	0,41	-1,15	0,04	0,92

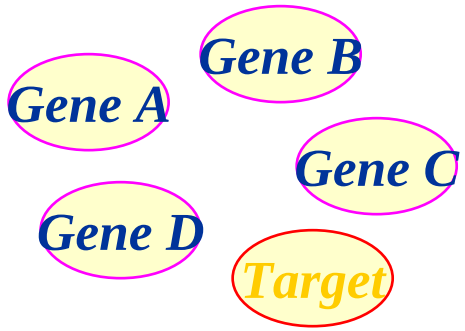
Numerical data for data mining

Example 1: Tumor Type Classification

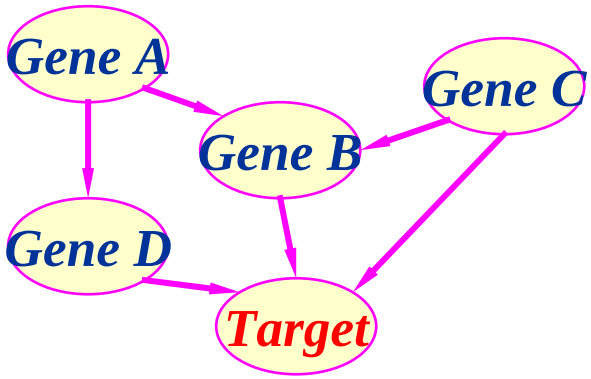
- Task: classification a leukemia samples into two classes based on gene expression patterns using Bayesian networks



- DNA microarray data from two kinds of leukemia patients



- Selected genes and the target variable



- Learned Bayesian network

Data Sets

- 72 samples in total: 25 AML (acute myeloid leukemia) samples + 47 ALL (acute lymphoblastic leukemia) samples (Golub et al., 1999).
- A data sample consists of expression measurements for over 7,000 genes.
- Preprocessing
  - 30 informative genes were selected by the P-metric

$$score(g) = \left| \frac{\mu_{AML} - \mu_{ALL}}{\sigma_{AML} + \sigma_{ALL}} \right|$$

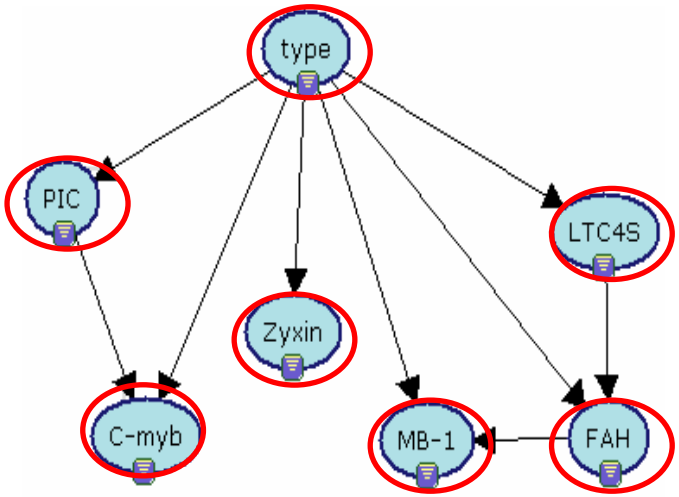
  - Expression values were discretized into 2 levels, *High* and *Low*.
  - The final data is 72 samples of which each consists of 'High' or 'Low' expression values of 30 genes

Classification by Bayesian Networks

- Classification as an inference for a variable (node) in Bayesian networks.

Bayes optimal classifier

$$c^*(\mathbf{g}) = \arg\max_{c \in C} P(c \mid \mathbf{g}, D)$$
$$= \arg\max_{c \in C} \sum_h P(c \mid \mathbf{g}, h) P(h \mid D)$$



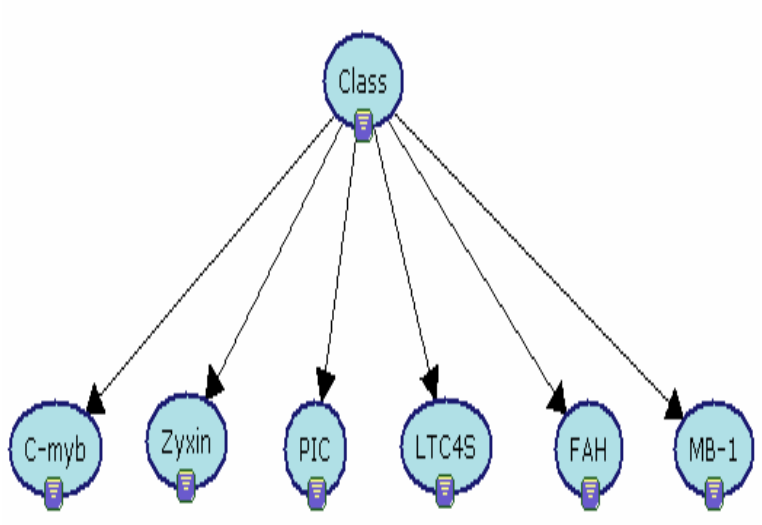
$$\text{if} \quad P(\hat{h} \mid D) = 1 \quad \Rightarrow \quad c^*(\mathbf{g}) = \text{argmax}_{c \in C} \; P(c \mid \mathbf{g}, \hat{h})$$

$$P_h(c_k \mid \mathbf{g}) = \frac{P_h(c_k, \mathbf{g})}{P_h(\mathbf{g})} = \frac{P_h(c_k, \mathbf{g})}{\sum_l P_h(c_l, \mathbf{g})} \propto P_h(c_k, \mathbf{g})$$

$$P(type, \mathbf{g}) = P(type) P(pic \mid type) \; P(ltc4s \mid type) P(zyxin \mid type)$$
$$P(c\_myb \mid type, pic) P(fah \mid type, ltc4s) P(mb-1 \mid type, fah)$$

Naïve Bayes Classifier

- Very restricted form of Bayesian network
  - Conditional independence of all variables (nodes) given the value of the class variable.
  - Though simple, widely used in classification tasks up to now.



$$P(type, \mathbf{s}) = P(type)P(\mathbf{s} \mid type)$$

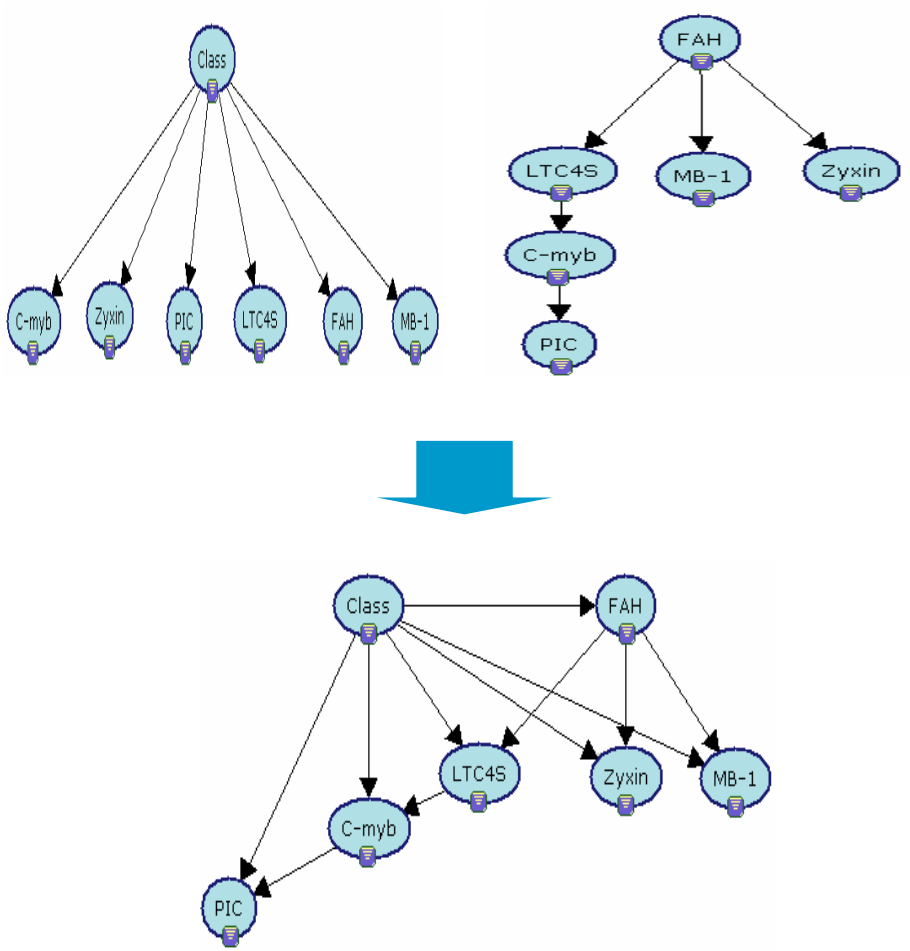
$$P(\mathbf{s} \mid type) = P(c\_myb \mid type)P(zyxin \mid type)$$
$$P(pic \mid type)P(ltc4s \mid type)$$
$$P(fah \mid type)P(mb\_1 \mid type)$$

Tree-Augmented Network

- (Friedman *et al.*, 1997)
- Naïve Bayes + tree-structured network of variables but class node.
  - Express the dependency between variables in a restricted way.
  - 'Class' is the root node
  - X_N can have at most one parent, besides 'Class' node.

$$P(zyxin \mid type)$$

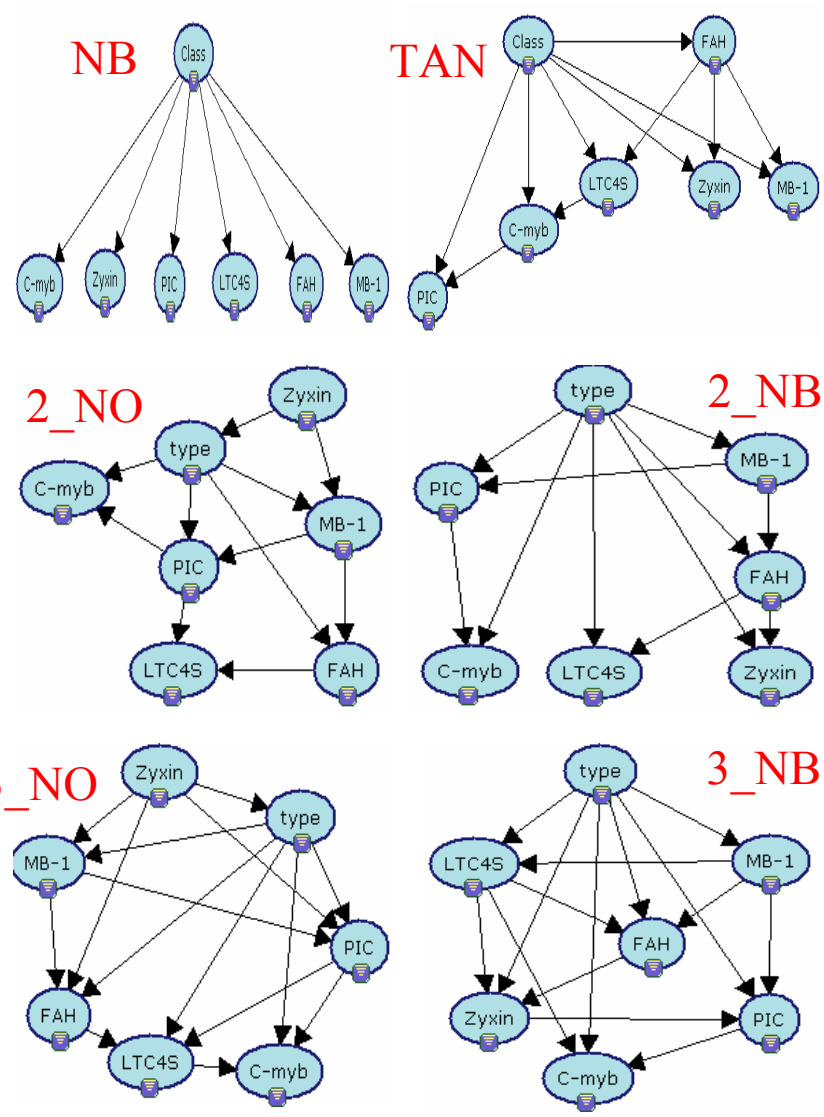
$$\leftrightarrow \quad P(zyxin \mid type, fah)$$



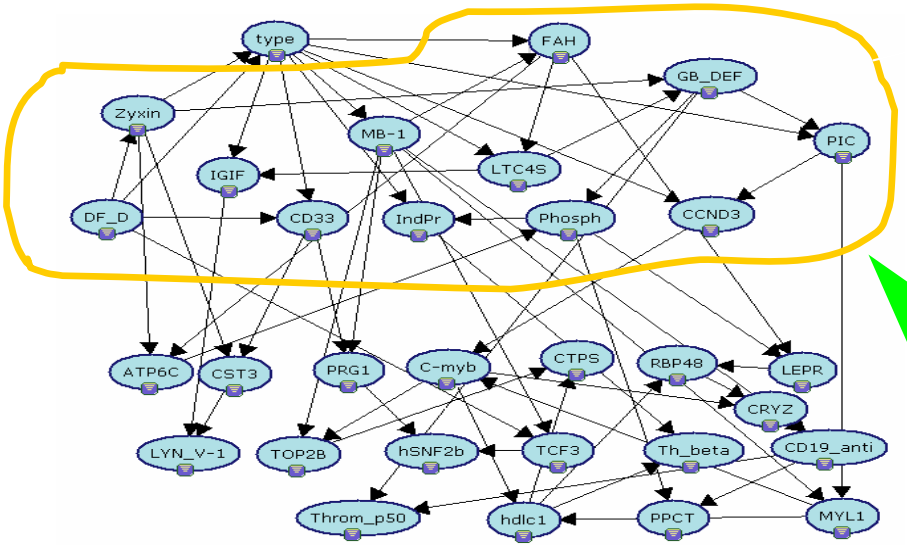
Results

- Leave-one-out cross-validation
  - Bayesian network learning with 71 samples and test for the remaining one samples.
  - 72 iteration

	6 genes	30 genes
NB	66/72	68/72
TAN	**69/72**	68/72
General BN (max #pa = 2)	66/72	67/72
	69/72 (NB)	69/72 (NB)
General BN (max #pa = 3)	**69/72**	65/72
	70/72 (NB)	67/72 (NB)



Markov Blanket in BN

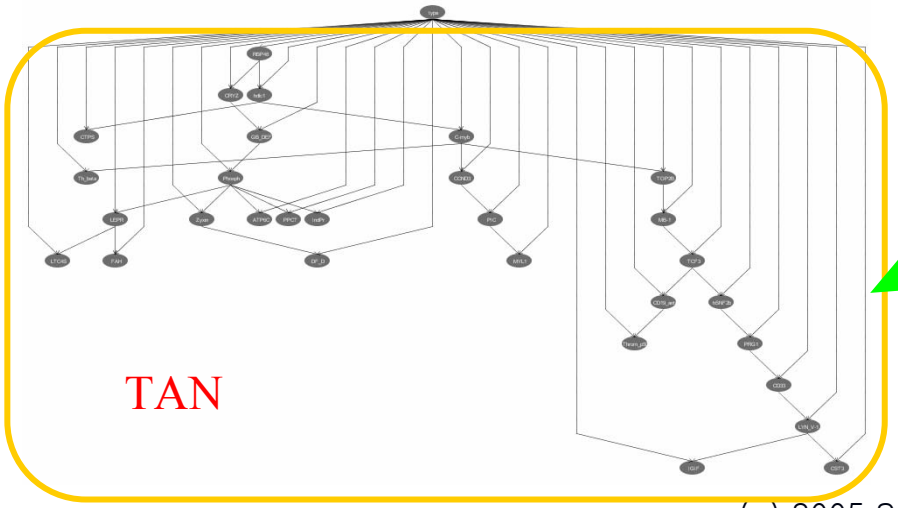


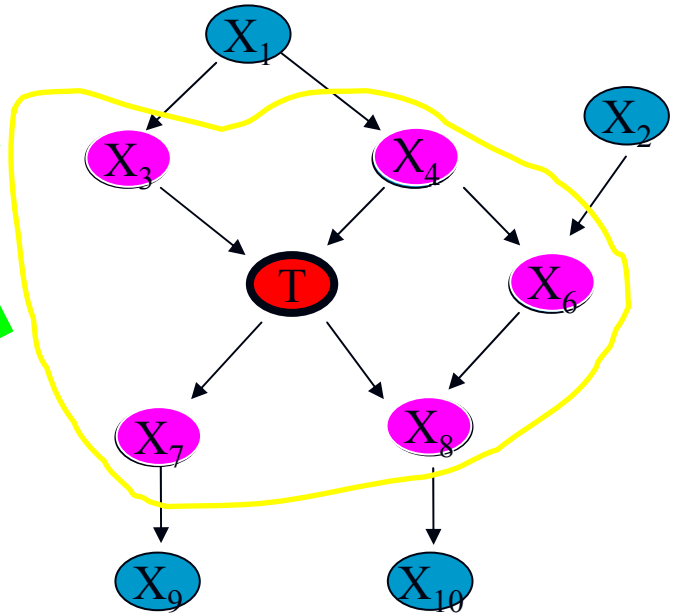
Markov Blanket of the node 'type'

Markov Blanket of the node *T*:

$P(T \mid \mathbf{X}\backslash\{T\}) = P(T \mid MB(T))$

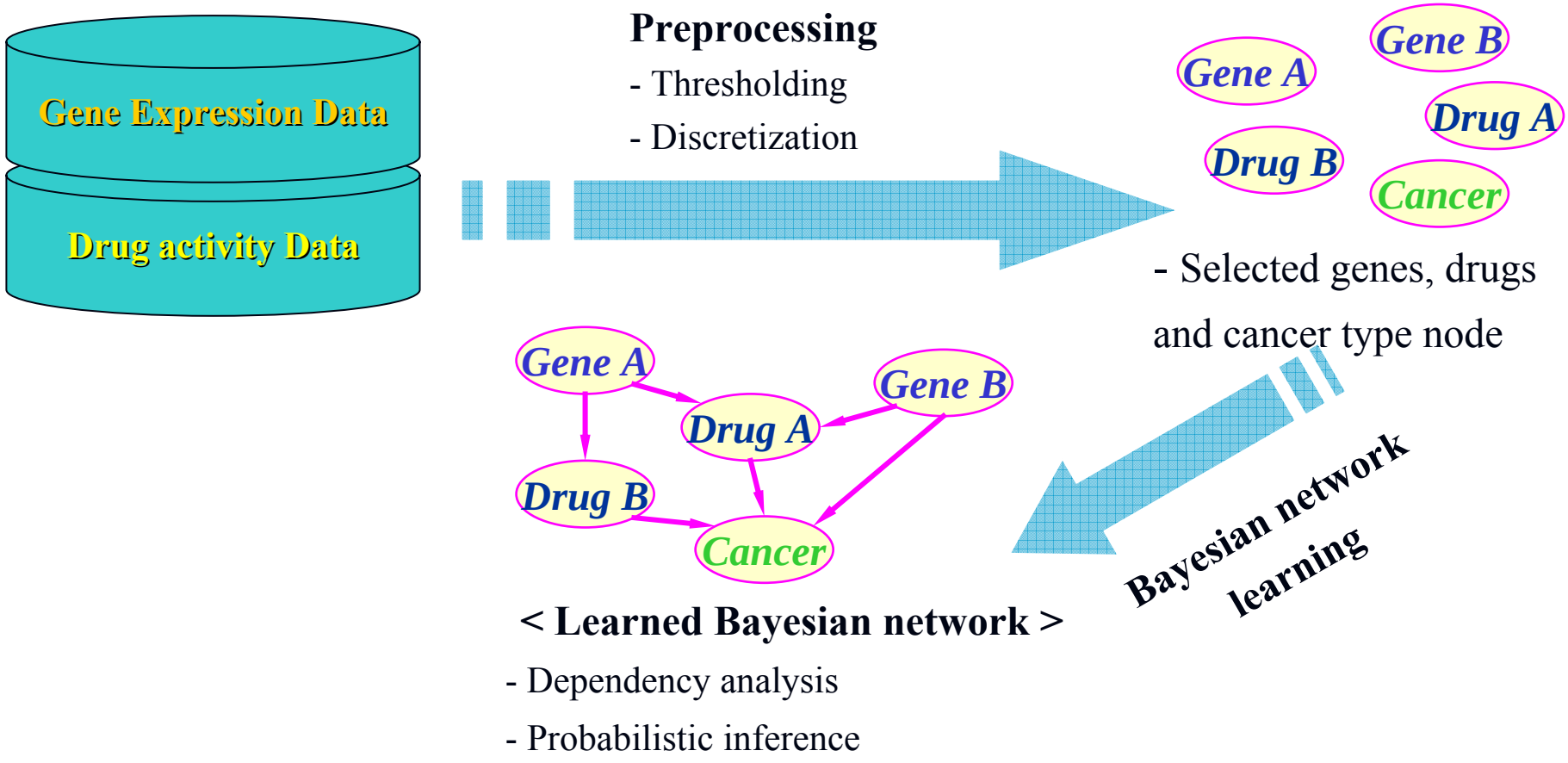
$MB(T) = \{Pa(T), Ch(T), Pa(Ch(T))\backslash\{T\}\}$





Example 2: Gene-Drug Dependency Analysis

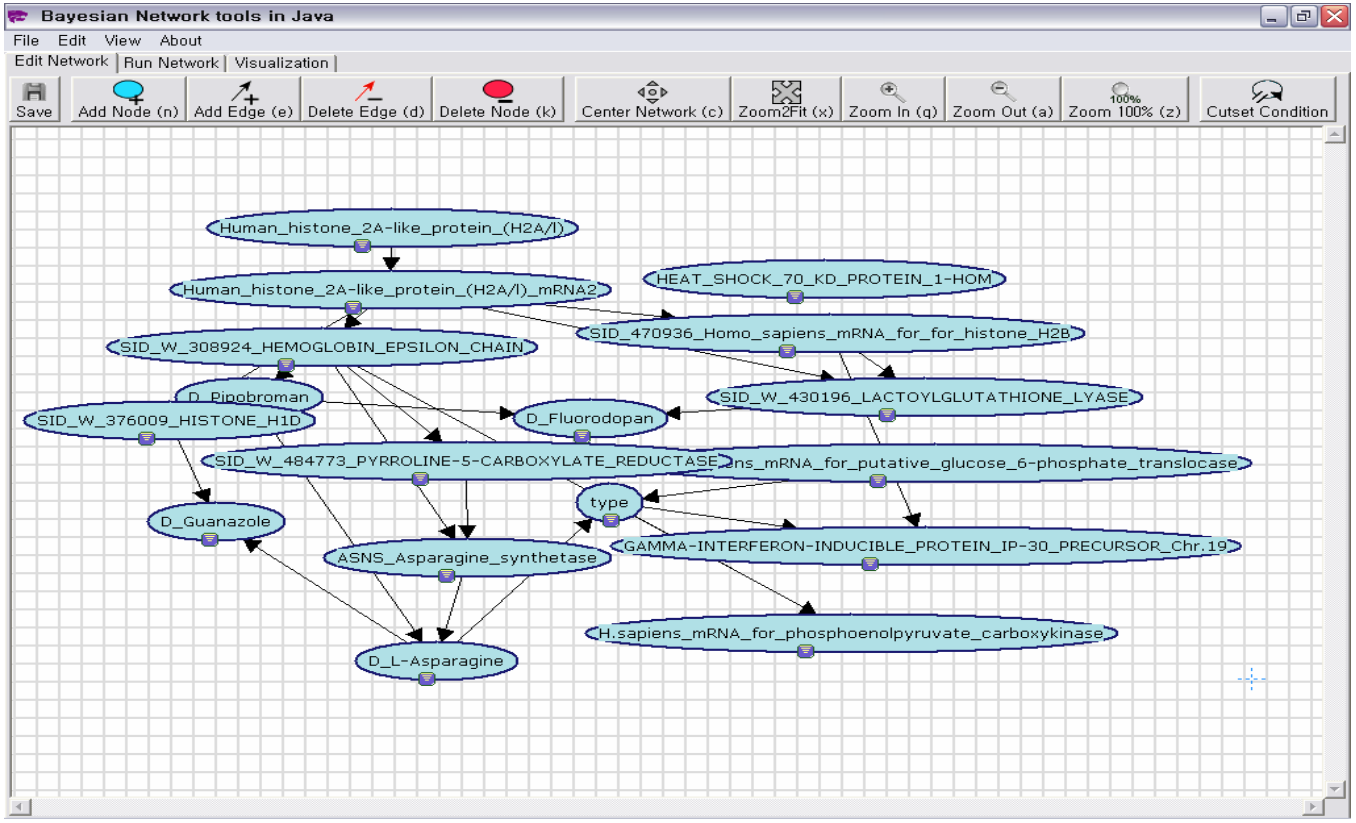
- Task: Construct a Bayesian network for the combined analysis of gene expression data and drug activity data.



Data Sets

- NCI60 data sets (Scherf et al., 2000)
  - 9703 genes and 1400 chemical compounds from 60 human cancer samples.
- Preprocessing
  - 12 genes and 4 drugs were selected based the correlation analysis result in (Scherf et al. 2000): Considering the learning time and visualization of Bayesian networks.
  - Discretize the gene expression values and drug activity values into 3 levels, *High*, *Mid*, *Low*.
- Nodes in Bayesian networks: 12 genes, 4 drugs, cancer type.

Results



Visualization by BNJ3 Software http://bnj.sourceforge.net/

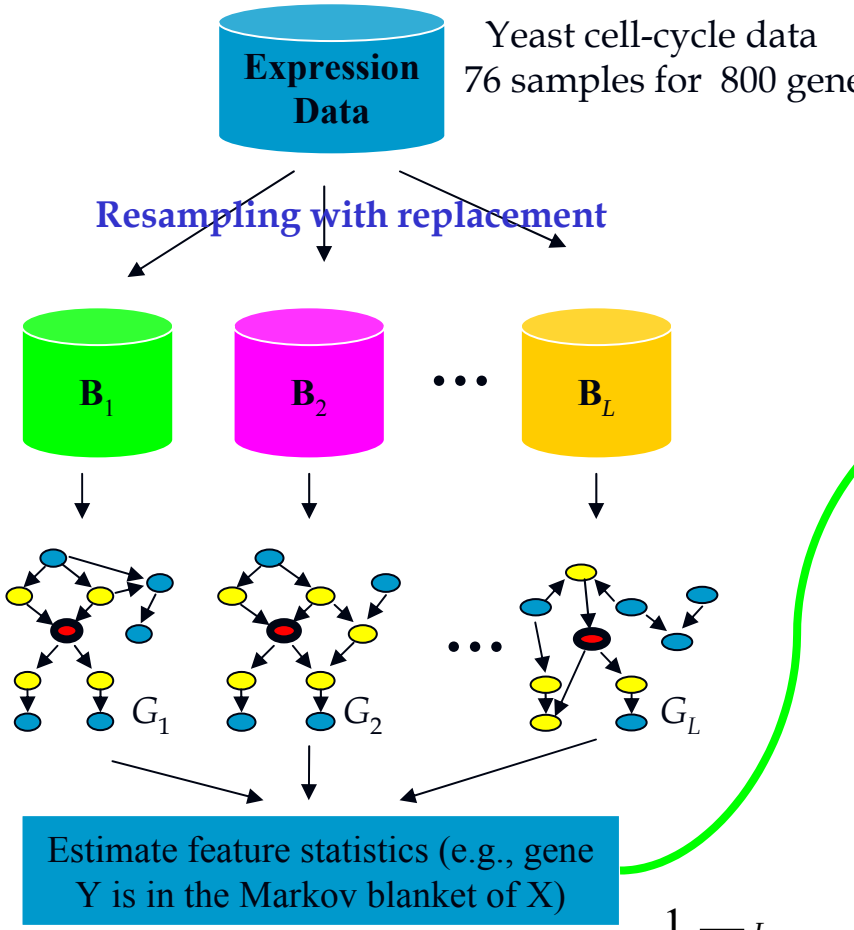
Identification of a plausible dependency among 2 genes and 1 drugs

Inferring Gene Regulation Model

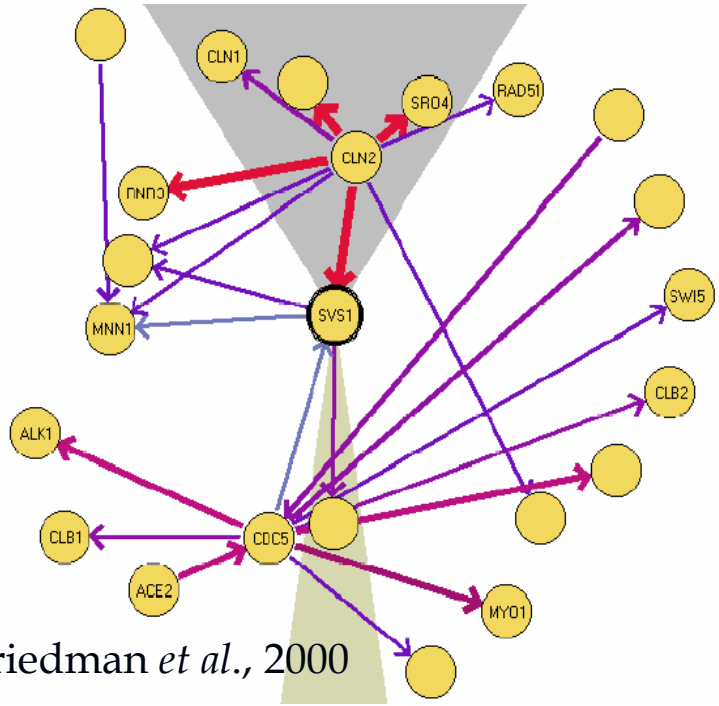
- Reconstruction of regulatory relation among genes from genome data.
- One of the challenges in reconstruction of gene regulatory networks using Bayesian networks is *statistical robustness*.
  - Arises from the sparse nature underlying gene expression data.
    - Usually, the number of samples are much smaller than that of genes (attributes).
    - Can produce unstable, spurious results.
  - Bootstrap, model averaging, incorporation of prior biological knowledge.

Bootstrap-based approach

- Multiple Bayesian networks from bootstrapped samples. (Friedman, N. *et al.*, 2000, Pe'er, D. *et al.*, 2001)



$$\mathrm{conf}(f) = \frac{1}{L}\sum_{i=1}^{L} f(G_i)$$

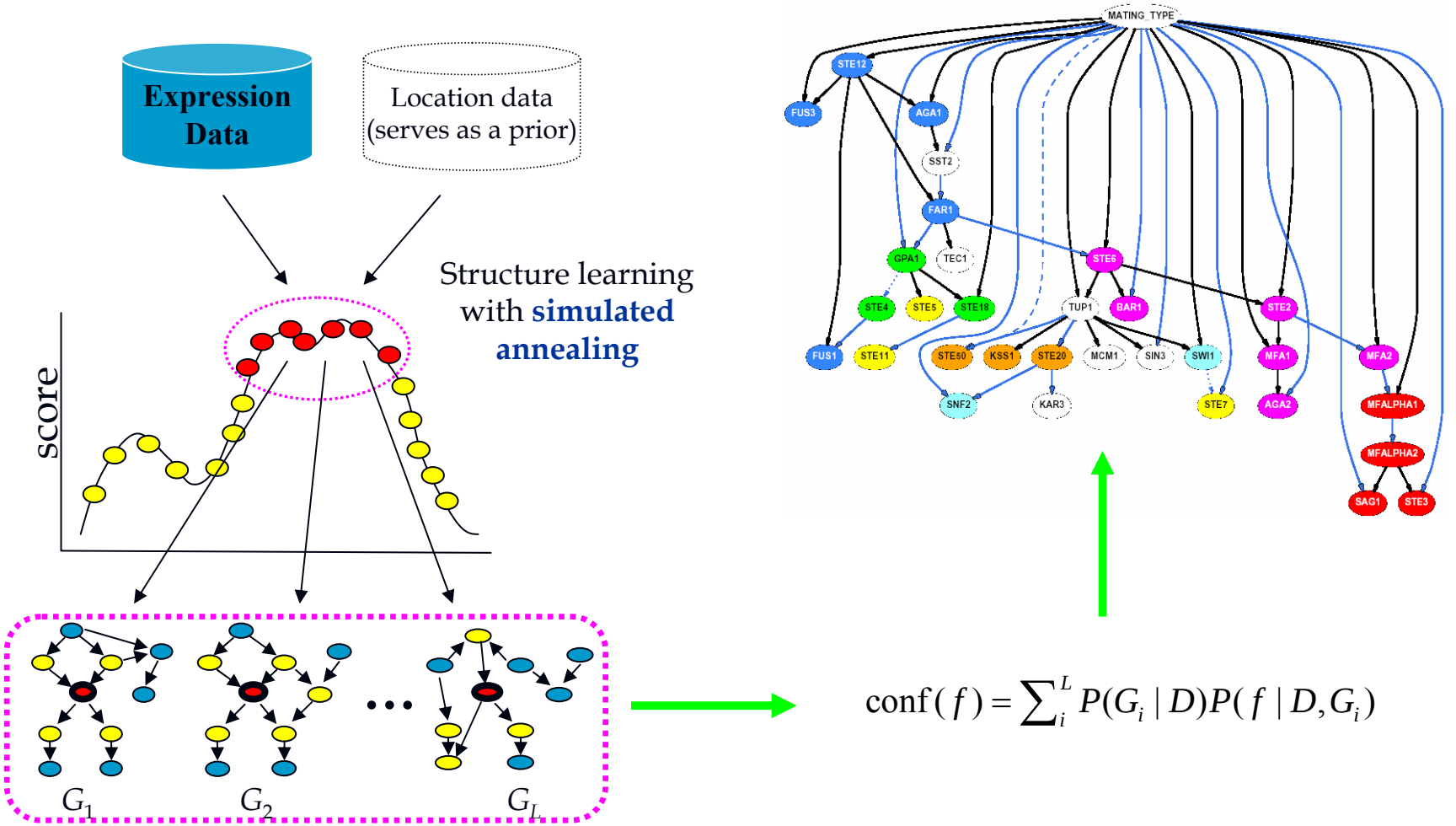


Friedman *et al.*, 2000

Identification of significant
- pairwise relations (Friedman, 2000)
- subnetworks (Pe'er, 2001)

Model averaging-based approach

- Estimate the confidence of features by averaging over posterior model (BN) distribution (Hertemink *et al.*, 2002)

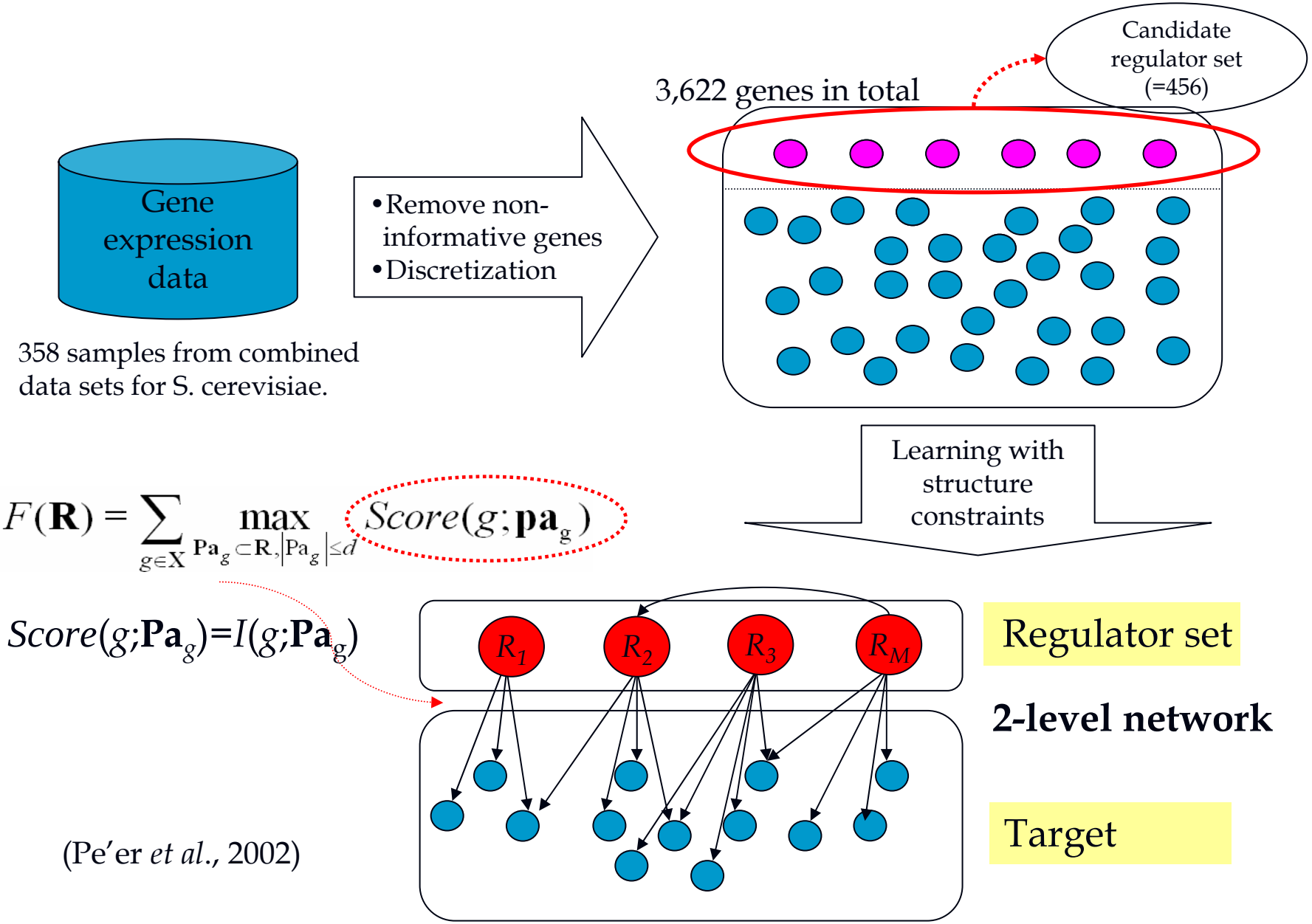


$$\mathrm{conf}(f) = \sum_i^L P(G_i \mid D) P(f \mid D, G_i)$$

- Incorporation prior knowledge or assumption.
  - Impose biologically-motivated assumptions on the network structure.
  - Construction of gene regulation model from gene expression data: inferring regulator set (Pe'er, D., *et al*., 2002)
    - **Assumption**
      - A relatively small set of genes is directly involved in transcriptional regulation
        - Limit the number of candidate regulators
        - Limit the number of parents for each node (gene) and the number of genes having outgoing edges.

➔ Significantly reduce the number of possible models.

➔ Statistical robustness of the identified regulator sets.


Gene expression data
358 samples from combined data sets for S. cerevisiae.
•Remove non-informative genes
•Discretization
3,622 genes in total
Candidate regulator set (=456)
Learning with structure constraints
$F(\mathbf{R}) = \sum_{g \in X} \max_{\mathbf{Pa}_g \subset \mathbf{R}, |\mathrm{Pa}_g| \le d} Score(g; \mathbf{pa}_g)$
$Score(g;\mathbf{Pa}_g)=I(g;\mathbf{Pa}_g)$
R_1
R_2
R_3
R_M
Regulator set
2-level network
Target
(Pe'er *et al.*, 2002)

Summary

- 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.

Summary (Cont'd)

- Not covered in this tutorial but important issues.
  - Probabilistic inference in general Bayesian networks
    - Exact & approximate inference
  - Incomplete data (missing data, hidden variables)
    - Known structure & unknown structure
    - Expectation-Maximization algorithm can be employed as in latent variable models.
  - Bayesian model averaging over structures as well as parameters.
  - Dynamic Bayesian networks for temporal data.

- Many applications
  - Text and web mining
  - Medical diagnosis
  - Intelligent agent system
  - Bioinformatics

References

- Bayesian Networks (papers & books)
  - Chow, C. and Liu, C., "*Approximating discrete probability distributions with dependency trees*", IEEE Transactions on Information Theory, 14, pp. 462-467, 1968.
  - Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, 1988.
  - Neapolitan, E., *Probabilistic Reasoning in Expert Systems*, 1990.
  - Charniak, E., Bayesian networks without tears, *AI Magazine*, 12(4):50-63, 1991.
  - Heckerman, Learning Bayesian networks: the combination of knowledge and statistical data, Machine Leanring, **20**, 197-243, 1995.
  - Jensen, F.V., *An Introduction to Bayesian Networks*, Springer-Verlag, 1996.
  - Friedman, N., Geiger, D., and Goldszmidt, M., "*Bayesian network classifiers*", Machine Learning, 29(2), pp. 131-163, 1997.
  - Mitchell, T.M., *Machine Learning*, The McGraw-Hill Companies, 1997.
  - Frey, B.J., *Graphical Models for Machine Learning and Digital Communication*, MIT Press, 1998.
  - Jordan, M. I. eds., *Learning in Graphical Models*, Kluwer Academic Publishers, 1998.
  - Friedman, N. and Goldszmidt, M., Learning Bayesian networks with local structure, In *Learning in Graphical Models* (Ed. Jordan, M. I.) , pp. 421-460, MIT Press, 1999.
  - Heckerman, D., A tutorial on learning with Bayesian networks, In *Learning with Graphical Models* (Ed. Jordan, M. I.), pp. 301-354, MIT Press, 1999.
  - J. Pearl and S. Russel, Bayesian networks, UCLA Cognitive Systems Laboratory, *Technical Report (R-277)*, November 2000.
  - Spirtes, P., Glymour, C., and Scheines, R., *Causation, Prediction, and Search, 2nd edition*, MIT Press, 2000.
  - Jensen, F. V., *Bayesian Networks and Decision Graphs*, Springer, 2001.
  - J. Pearl, Bayesian networks, causal inference and knowledge discovery, UCLA Cognitive Systems Laboratory, *Technical Report (R-281)*, March 2001.
  - Korb, K. B. and Nicholson, A. B., *Bayesian Artificial Intelligence*, CRC Press, 2004.

- Graphical models and Bayesian networks (tutorials & lectures)
  - Friedman, N. and Koller, D., Learning Bayesian Networks from Data, (http://www.cs.huji.ac.il/~nir/Nips01-Tutorial/)
  - Murphy, K., A Brief Introduction to Graphical Models and Bayesian Networks (http://www.cs.ubc.ca/~murphyk/Bayes/bayes.html)
  - Ghahramani, Z., Probabilistic Models for Unsupervised Learning, (http://www.gatsby.ucl.ac.uk/~zoubin/NIPStutorial.html)
  - Ghahramani, Z., Bayesian Methods for Machine Learning, (http://www.gatsby.ucl.ac.uk/~zoubin/ICML04-tutorial.html)
  - Moser, A., Probabilistic Independence Networks (I-II), (http://www.eecis.udel.edu/~bloch/eleg867/notes/PIN_I_rev2/index.htm, http://www.eecis.udel.edu/~bloch/eleg867/notes/PIN_II_rev2/index.htm)
  - Poet, M., Special Topics: Belief Networks, (http://www.stat.duke.edu/courses/Spring99/sta294/)

- **Bayesian network with application to bioinformatics**
  - Murphy, K. and Mian, S., Modeling gene expression data using dynamic Bayesian networks, Technical. report 1999: Computer Science Division, University of California, Berkeley, CA
  - Friedman, N., Linial, M., Nachman, I., and Pe'er, D., Using Bayesian networks to analyze expression data, Journal of Computational Biology, 7(3/4), pp. 601-620, 2000.
  - Cai, D., Delcher, A., Kao, B., and Kasif, S. Modeling splice sites with Bayes networks, Bioinformatics, 2000, 16(2), pp. 152-158, 2000.
  - Pe'er, D., Regev, A., Elidan, G., and Friedman, N., Inferring subnetworks from perturbed expression profiles, Bioinformatics, 17(suppl 1), pp. S215-S224, 2001.
  - Pe'er, D., Regev, A., and Tanay, A., Minreg: inferring an active regulator set, Bioinformatics, 18(suppl 1), pp. 258-267, 2002.
  - Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., and Young, R. A., Combining location and expression data for principled discovery of genetic regulatory network models, Pacific Symposium on Biocomputing, **7**, pp. 437-449, 2002.
  - Imoto, S., Kim, S., Goto, T., Aburatani, S., Tashiro, K., Kuhara, S., and Miyano, S., Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network, Journal of Bioinformatics and Computational Biology, 1(2), pp. 231-252, 2003.
  - Ronald, J., et al., A Bayesian networks approach for predicting protein-protein interactions from genomic data, Science **302**: 449-453, 2003.
  - Olga G. Troyanskaya, O. G., Dolinski, K., Owen, A. B., Altman, R. B., and Botstein, D., A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae), PNAS, 100(14), pp. 8348–8353, 2003.
  - Beer M. A. and Tavazoie, S. Predicting gene expression from sequence. Cell, **117**(2), pp. 185-198, 2004.
  - Friedman, N., "Inferring cellular networks using probabilistic graphical models", Science, 303(5659), pp. 799-805, 2004.
- **A comprehensive list is available at http://zlab.bu.edu/kasif/bayes-net.html**

- Bayesian network software packages
  - Bayes Net Toolbox (Kevin Murphy)
    - http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html
    - 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) .
  - Detailed lists and comparison are referred to
    - http://www.cs.ubc.ca/~murphyk/Bayes/bnsoft.html
    - Bayesian network list in Google
      - http://directory.google.com/Top/Computers/Artificial_Intelligence/Belief_Networks/Software/
    - Korb, K. B. and Nicholson, A. B., *Bayesian Artificial Intelligence*, CRC Press, 2004.
      - http://www.csse.monash.edu.au/bai/book/appendix_b.pdf