Bayesian Belief Networks

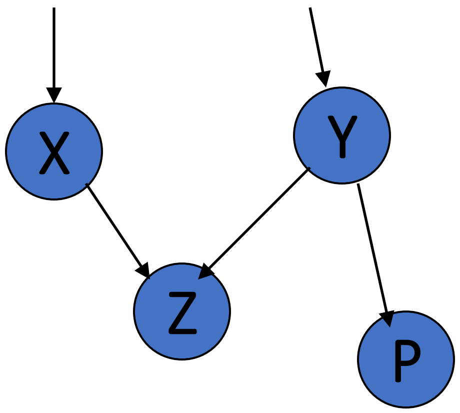
Adopted from ‘Data Mining Concepts and Techniques’ and Tim Finin

Introduction

- Bayesian Belief Networks (BBNs) can reason with networks of propositions and associated probabilities
  - Useful for many AI problems
  - Diagnosis
  - Expert systems
  - Planning
  - Learning

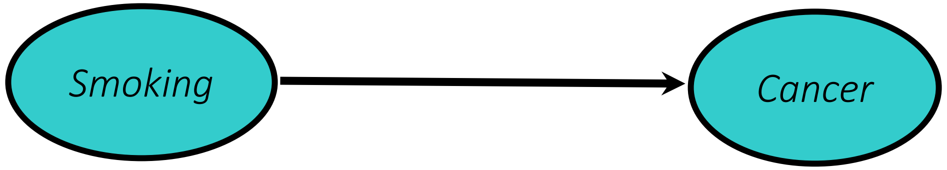
Introduction

- Bayesian belief networks (also known as Bayesian networks, probabilistic networks): allow class conditional independencies between subsets of variables
- A (directed acyclic) graphical model of causal relationships represents dependency among the variables
- Gives a specification of joint probability distribution
- Nodes: random variables
- Links: dependency
- X and Y are the parents of Z, and Y is the parent of P
- No dependency between Z and P
- Has no loops/cycles



Example

$S \in \{no, light, heavy\}$

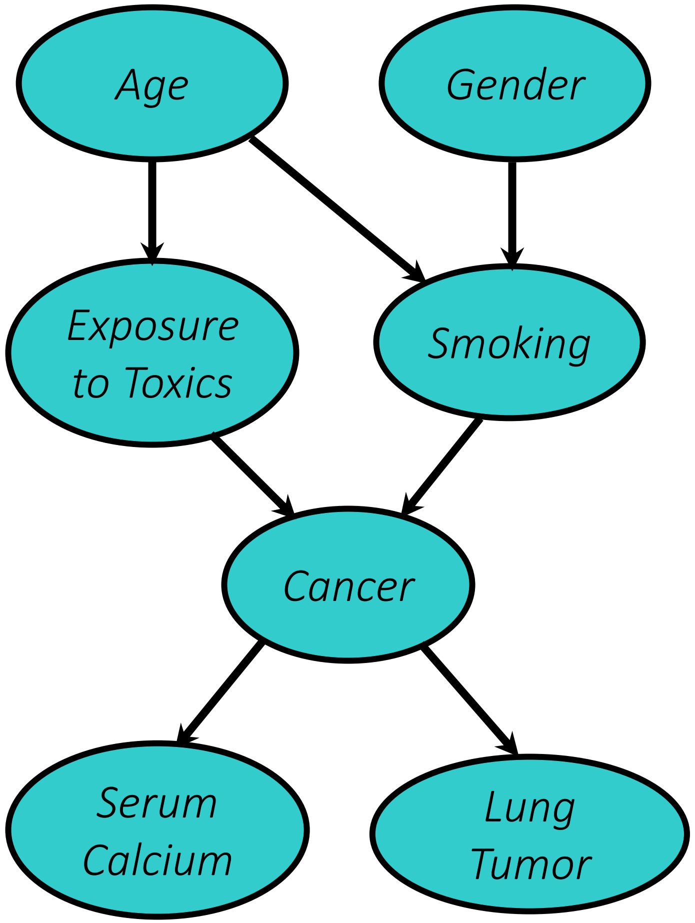


P(S=no)	0.80
P(S=light)	0.15
P(S=heavy)	0.05

$C \in \{none, benign, malignant\}$

Smoking=	no	light	heavy
P(C=none)	0.96	0.88	0.60
P(C=benign)	0.03	0.08	0.25
P(C=malig)	0.01	0.04	0.15

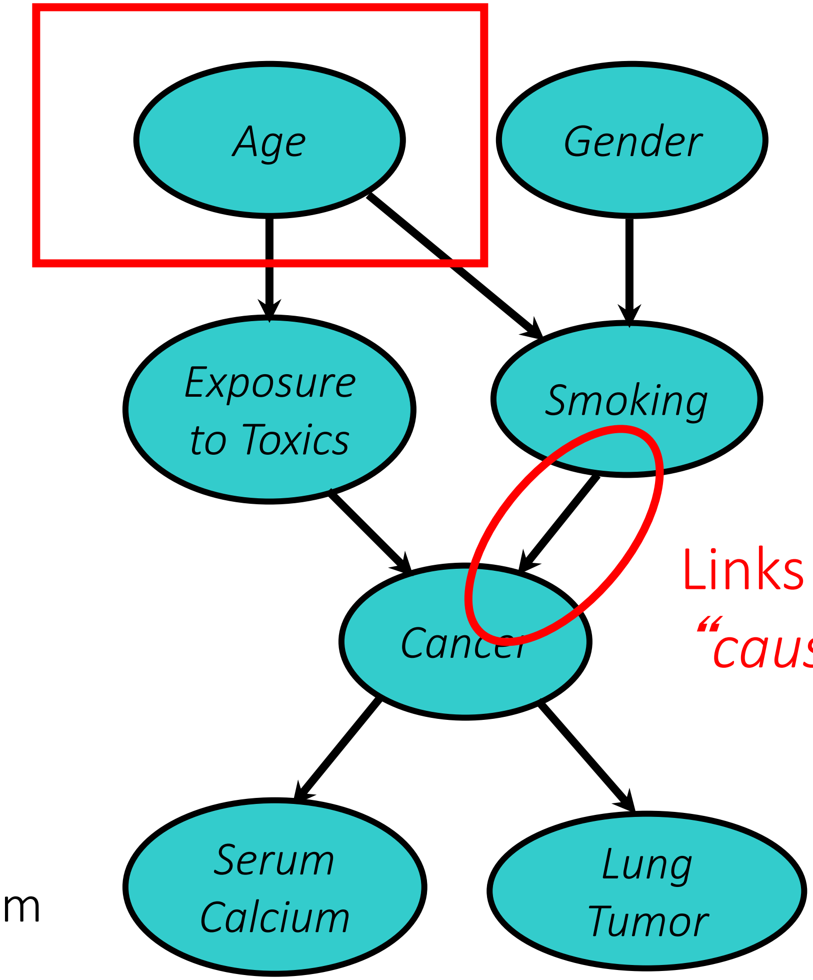
Example



Age
Gender
Exposure to Toxics
Smoking
Cancer
Serum Calcium
Lung Tumor

Example

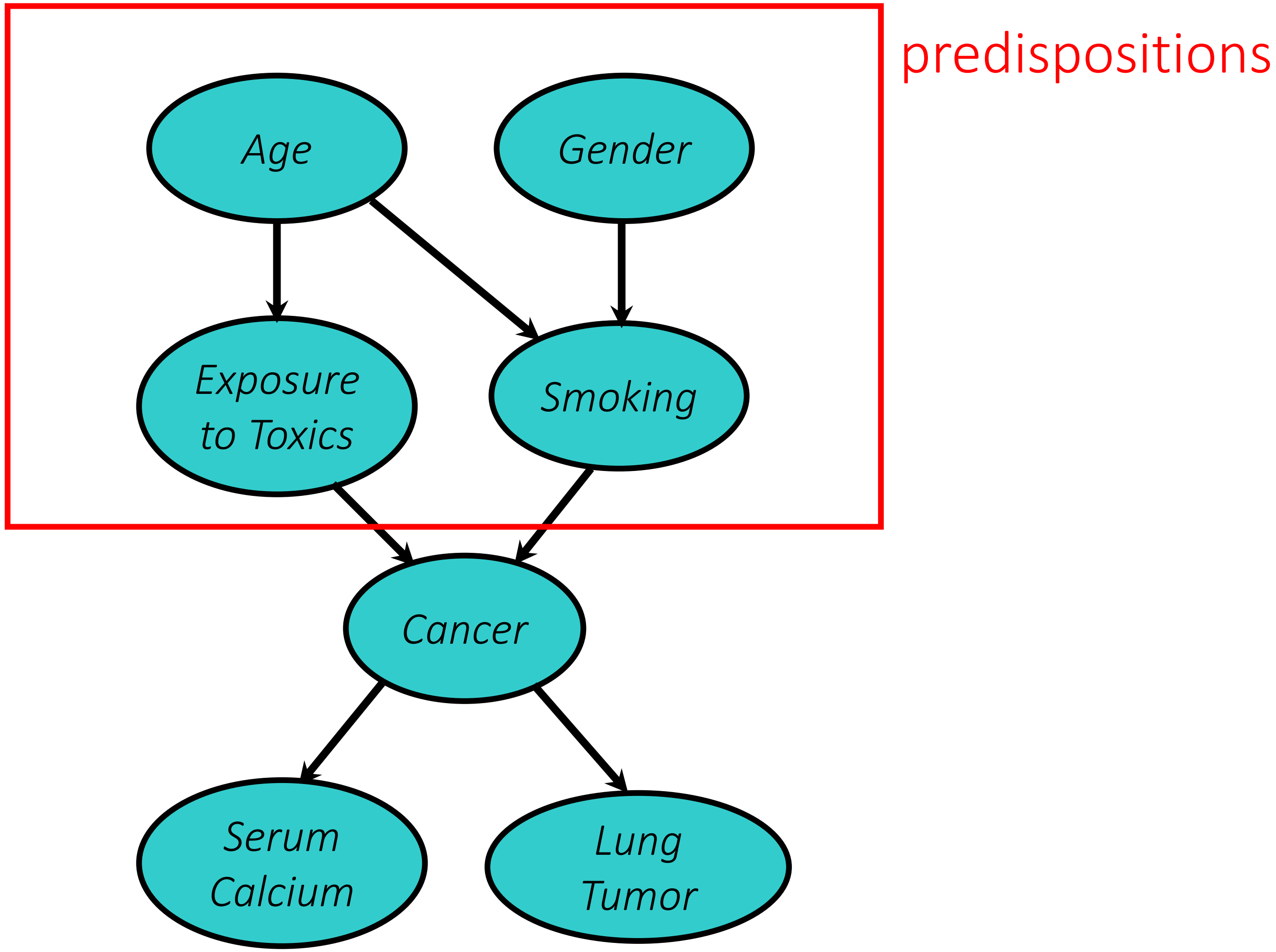
Nodes represent variables



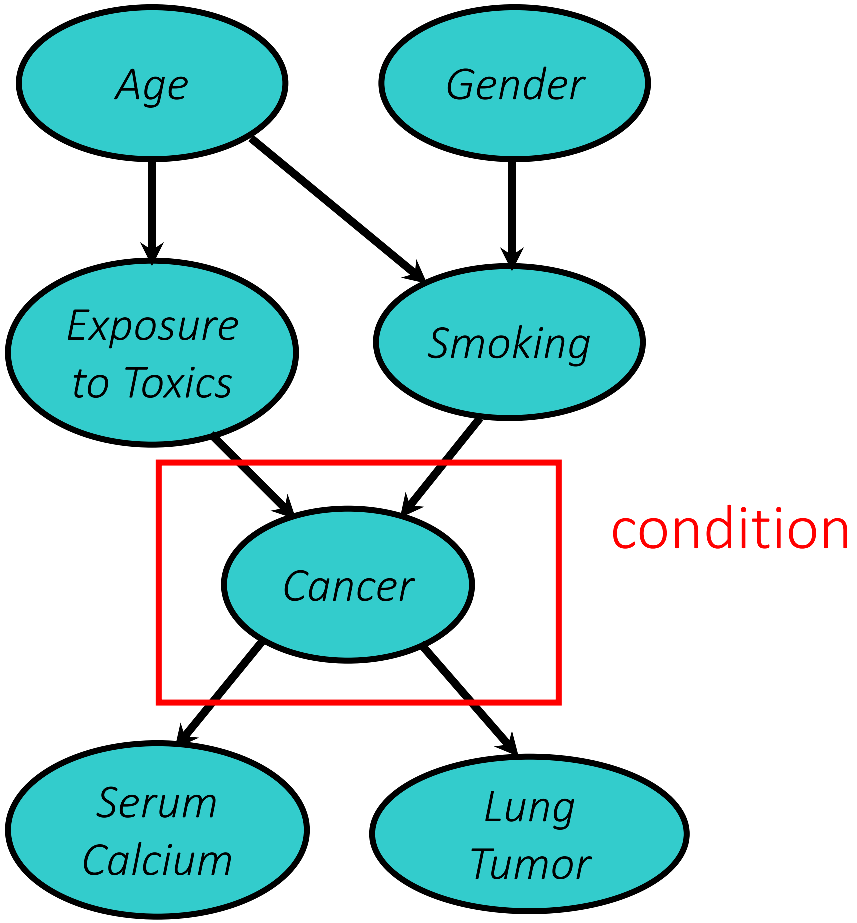
Links represent "*causal*" relations

- Does gender cause smoking?
- Influence might be a more appropriate term

Example

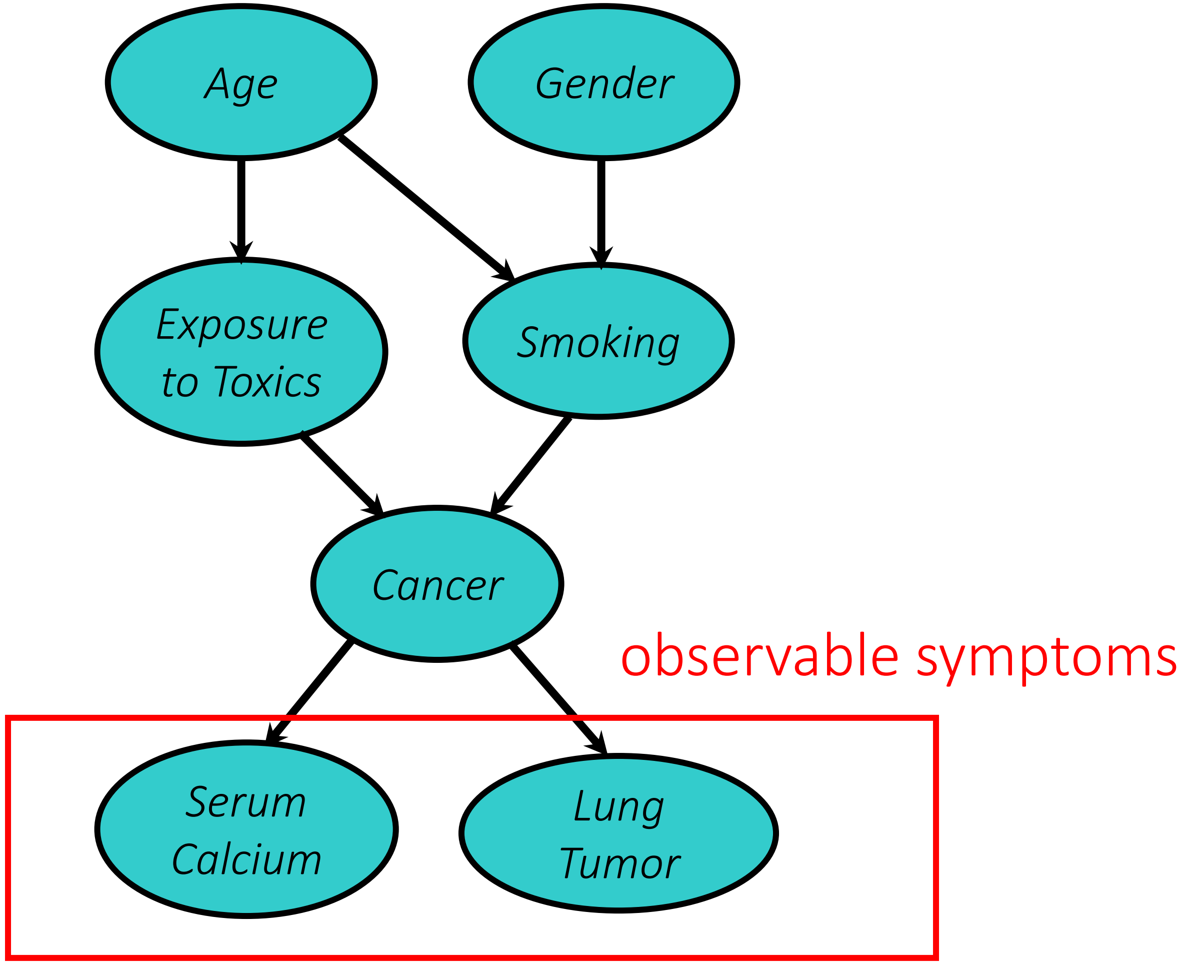

predispositions
Age
Gender
Exposure to Toxics
Smoking
Cancer
Serum Calcium
Lung Tumor

Example



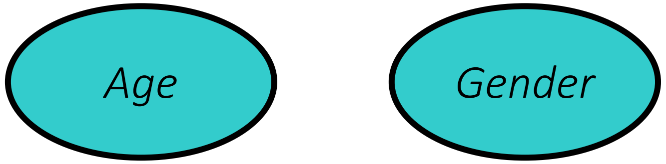
Age
Gender
Exposure to Toxics
Smoking
Cancer
condition
Serum Calcium
Lung Tumor

Example



Age
Gender
Exposure to Toxics
Smoking
Cancer
observable symptoms
Serum Calcium
Lung Tumor

Independence



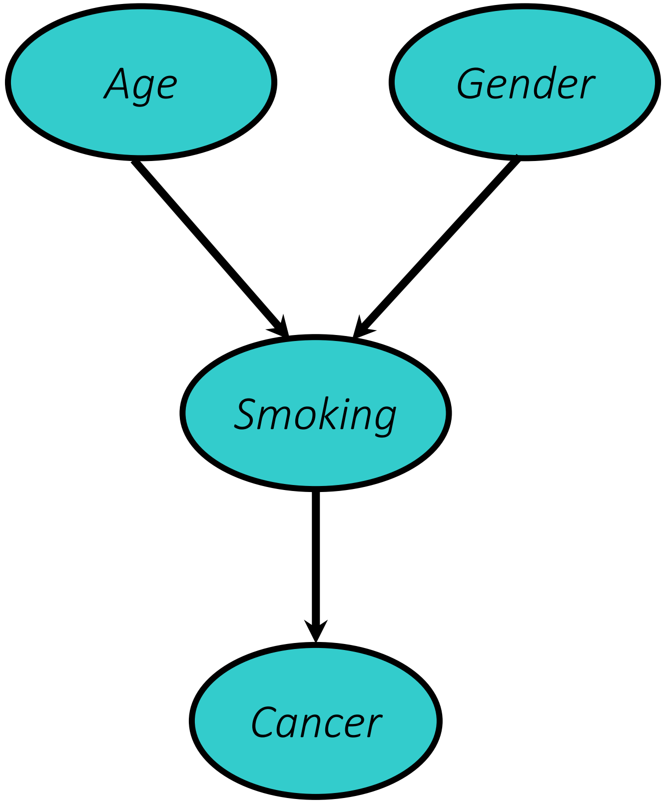
Age and *Gender* are independent.

$$P(A,G) = P(G)\, P(A)$$

$$P(A \mid G) = P(A)$$
$$P(G \mid A) = P(G)$$

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

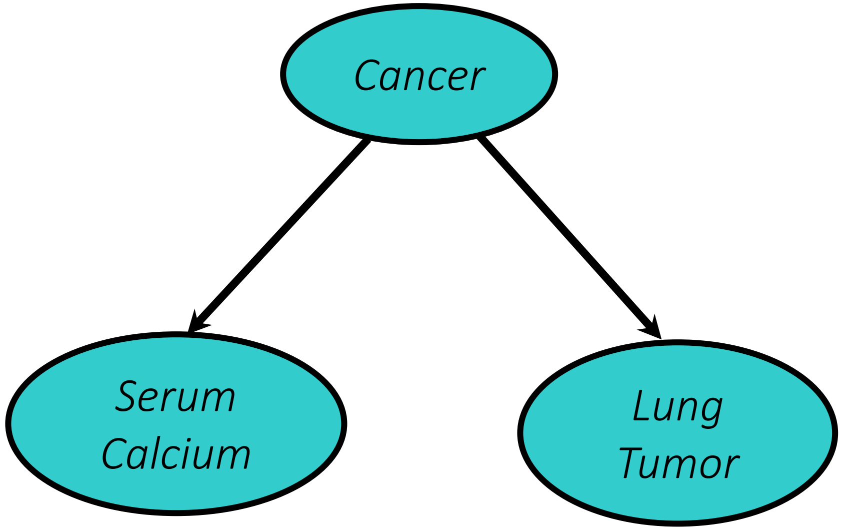
Conditional Independence



Cancer is independent of *Age* and *Gender* given *Smoking*

$$P(C \mid A,G,S) = P(C|S)$$

Conditional Independence



Serum Calcium and *Lung Tumor* are dependent

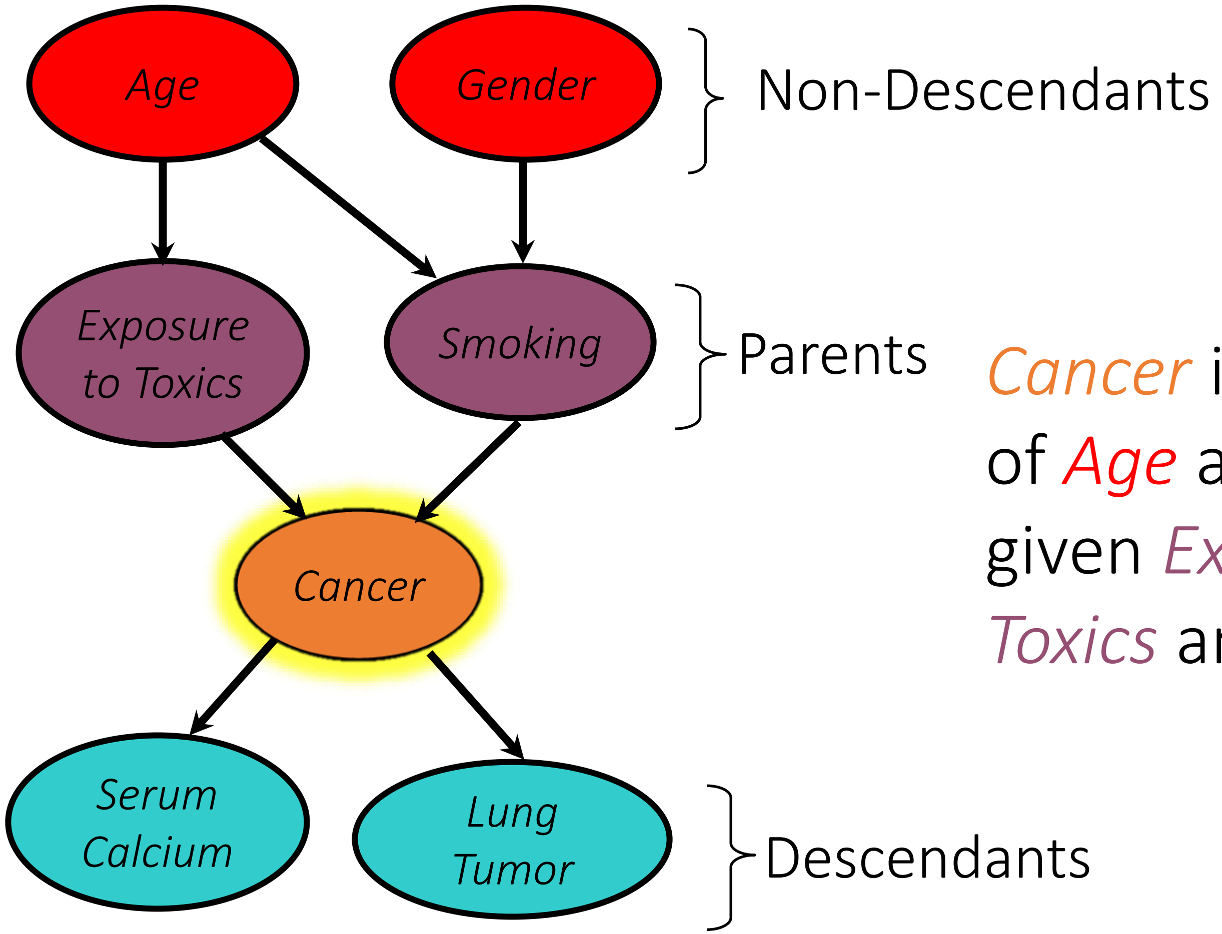
Serum Calcium is independent of *Lung Tumor*, given *Cancer*

$$P(L \mid SC,C) = P(L|C)$$
$$P(SC \mid L,C) = P(SC|C)$$

Naïve Bayes assumption: evidence (e.g., symptoms) is independent given the disease. This makes it easy to combine evidence

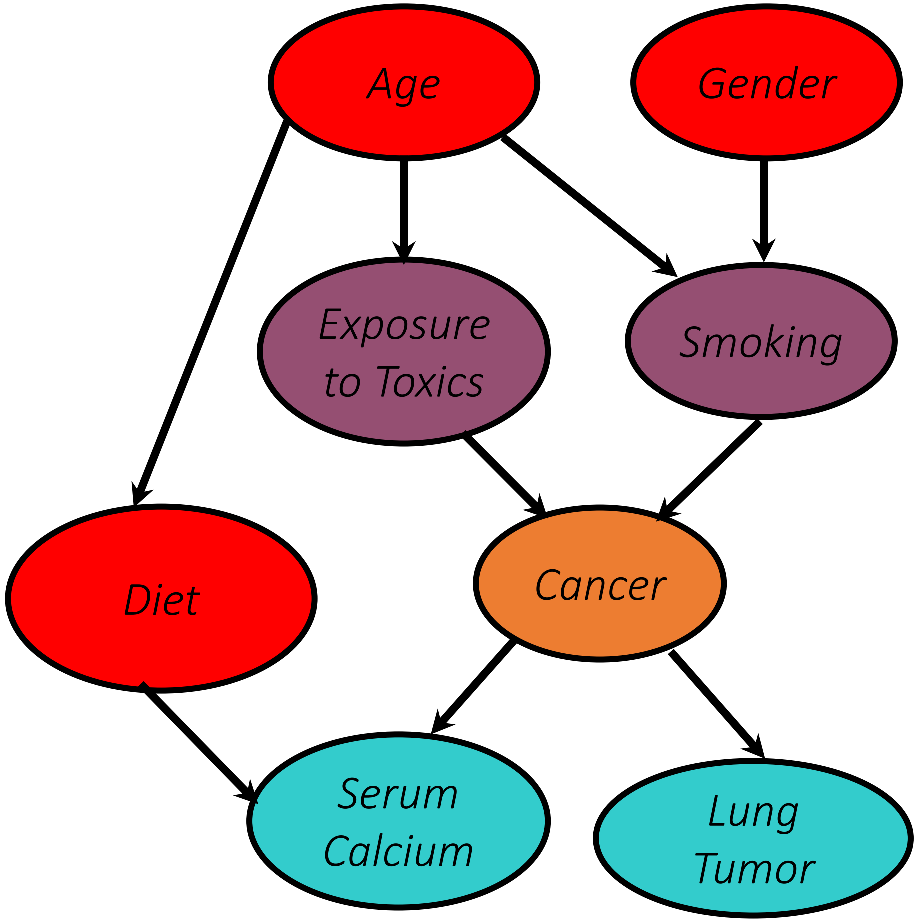
Conditional Independence

A variable (node) is conditionally independent of its non-descendants given its parents



Cancer is independent of *Age* and *Gender* given *Exposure to Toxics* and *Smoking*.

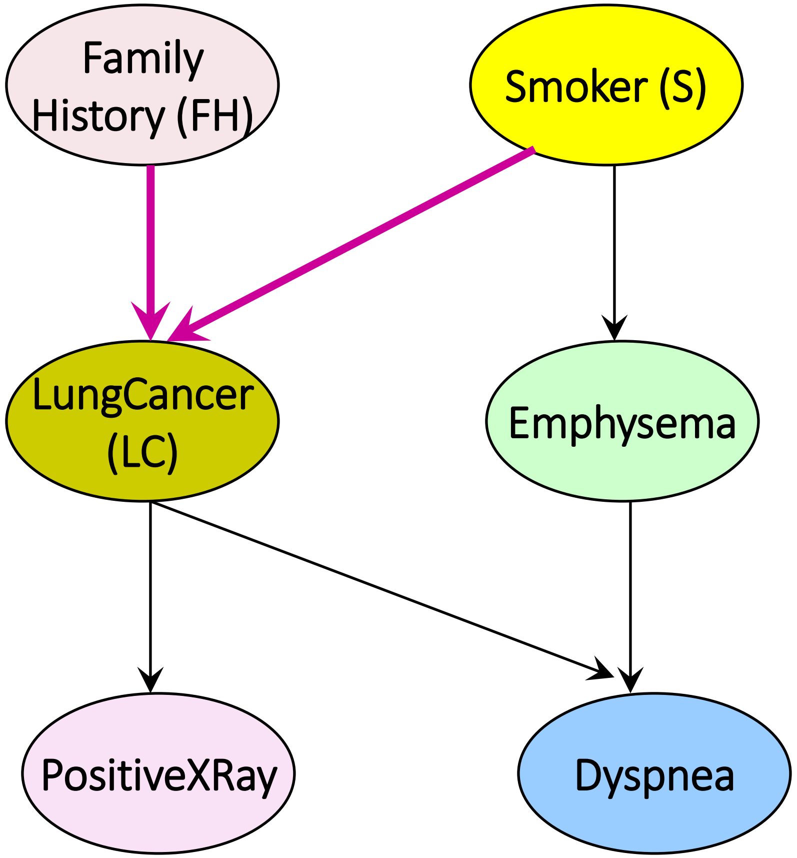
Conditional Independence



A variable is conditionally independent of its non-descendants given its parents

Cancer is independent of *Diet* given *Exposure to Toxics* and *Smoking*

Another Example



Bayesian Belief Network

CPT: **Conditional Probability Table** for variable LungCancer:

	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

shows the conditional probability for each possible combination of its parents

Derivation of the probability of a particular combination of values of **X**, from CPT:

$$P(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(Y_i))$$

Training

- Scenario 1: Given both the network structure and all variables observable: compute only the CPT entries
- Scenario 2: Network structure known, some variables hidden: gradient descent (greedy hill-climbing) method, i.e., search for a solution along the steepest descent of a criterion function
  - Weights are initialized to random probability values
  - At each iteration, it moves towards what appears to be the best solution at the moment, w.o. backtracking
  - Weights are updated at each iteration & converge to local optimum
- Scenario 3: Network structure unknown, all variables observable: search through the model space to reconstruct network topology
- Scenario 4: Unknown structure, all hidden variables: No good algorithms known for this purpose

Construction

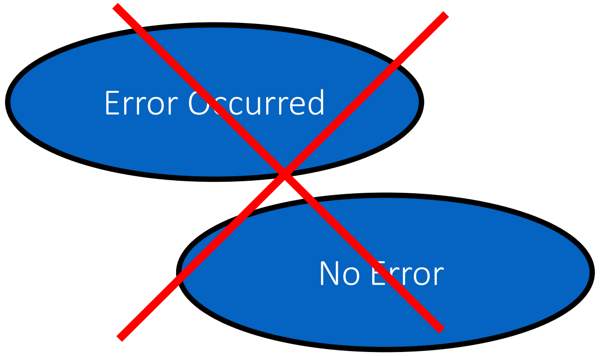
- The knowledge acquisition process for a BBN involves three steps
- Choosing appropriate variables
- Deciding on the network structure
- Obtaining data for the conditional probability tables

Construction

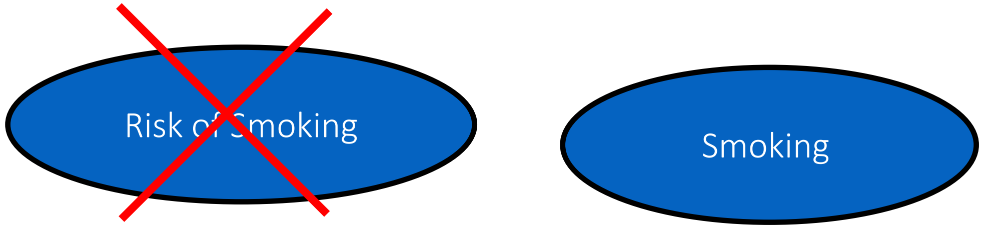
Variables should be collectively exhaustive, mutually exclusive values

$$x_1 \vee x_2 \vee x_3 \vee x_4$$

$$\neg(x_i \wedge x_j) \quad i \neq j$$



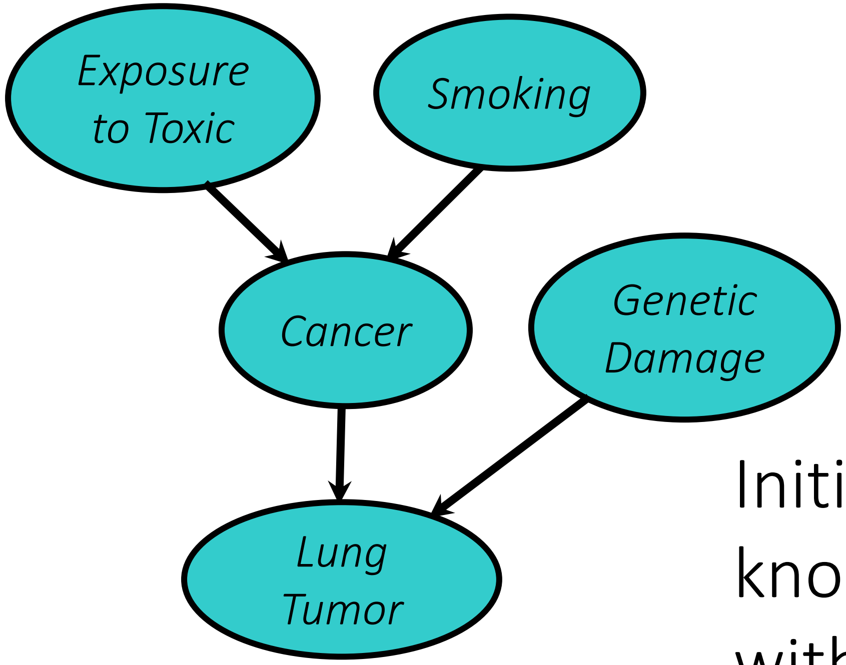
They should be values, not probabilities



Construction

- Example of good variables
  - Weather {Sunny, Cloudy, Rain, Snow}
  - Gasoline: Cents per gallon
  - Temperature { >=100F , < 100F}
  - User needs help on Excel Charting {Yes, No}
  - User’s personality {dominant, submissive}

Structure

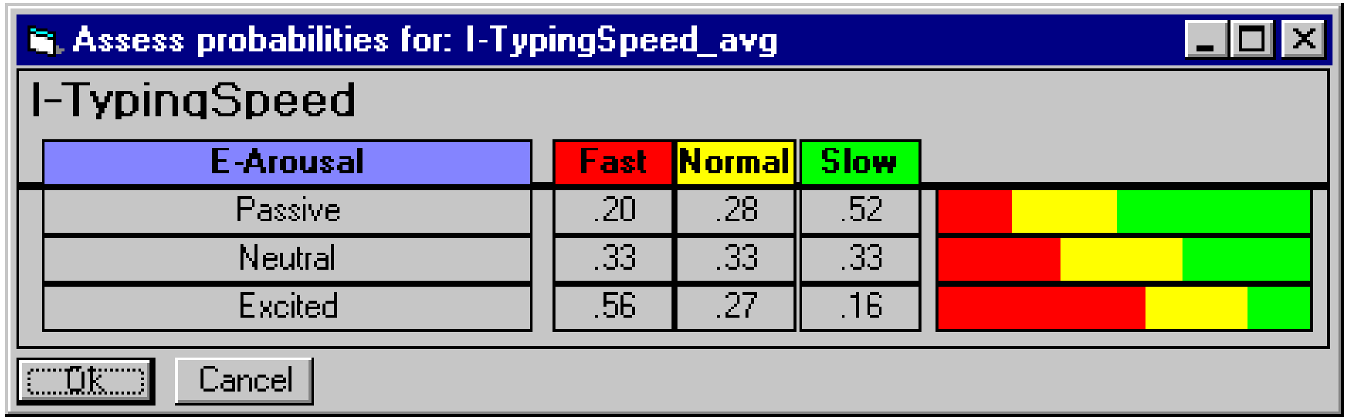


Network structure corresponding to “causality” is usually good.

Initially this uses the designer’s knowledge but can be checked with data

Structure

- Second decimal usually doesn't matter
- Relative probabilities are important



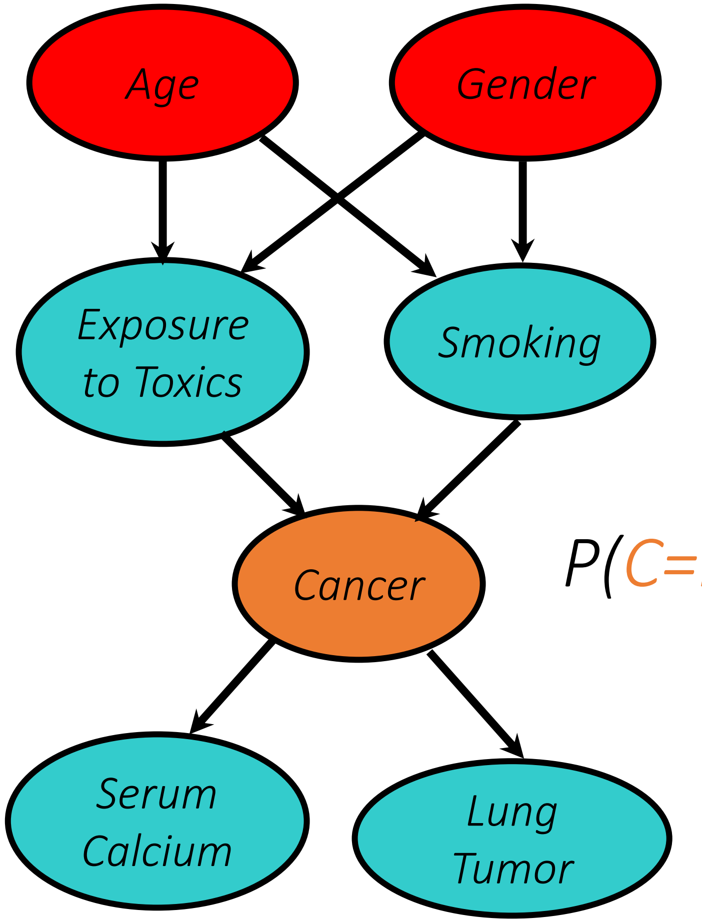
E-Arousal	Fast	Normal	Slow
Passive	.20	.28	.52
Neutral	.33	.33	.33
Excited	.56	.27	.16

- Zeros and ones are often enough
- Order of magnitude is typical: 10^{-9} vs 10^{-6}
- Sensitivity analysis can be used to decide accuracy needed

Reasoning

- BBNs support three main kinds of reasoning:
  - Predicting conditions given predispositions
  - Diagnosing conditions given symptoms (and predisposing)
  - Explaining a condition in by one or more predispositions
- To which we can add a fourth:
  - Deciding on an action based on the probabilities of the conditions

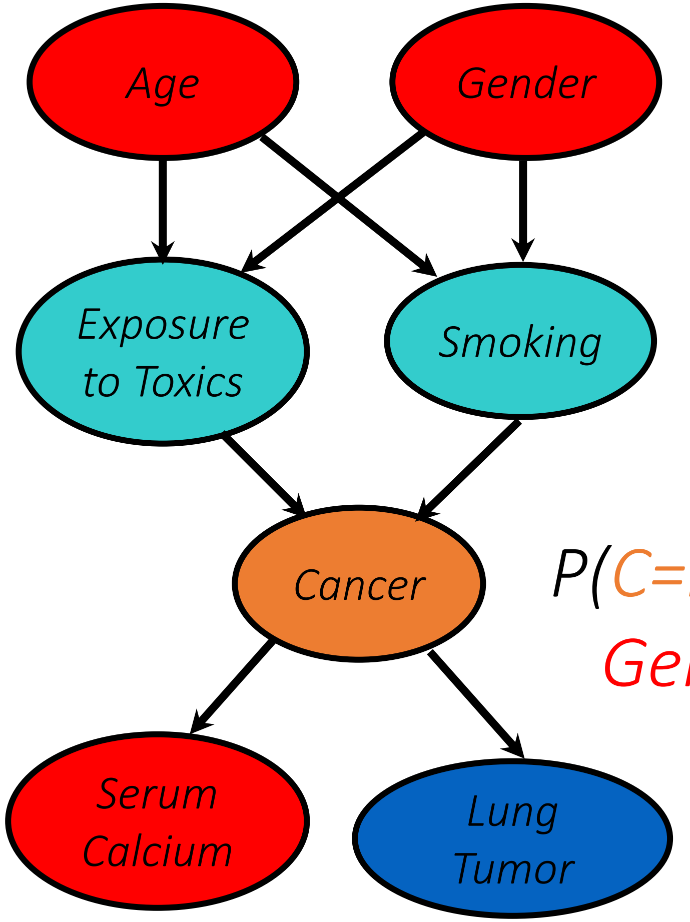
Predictive Inference



How likely are elderly males to get malignant cancer?

$P(C{=}malignant \mid Age{>}60, Gender{=}male)$

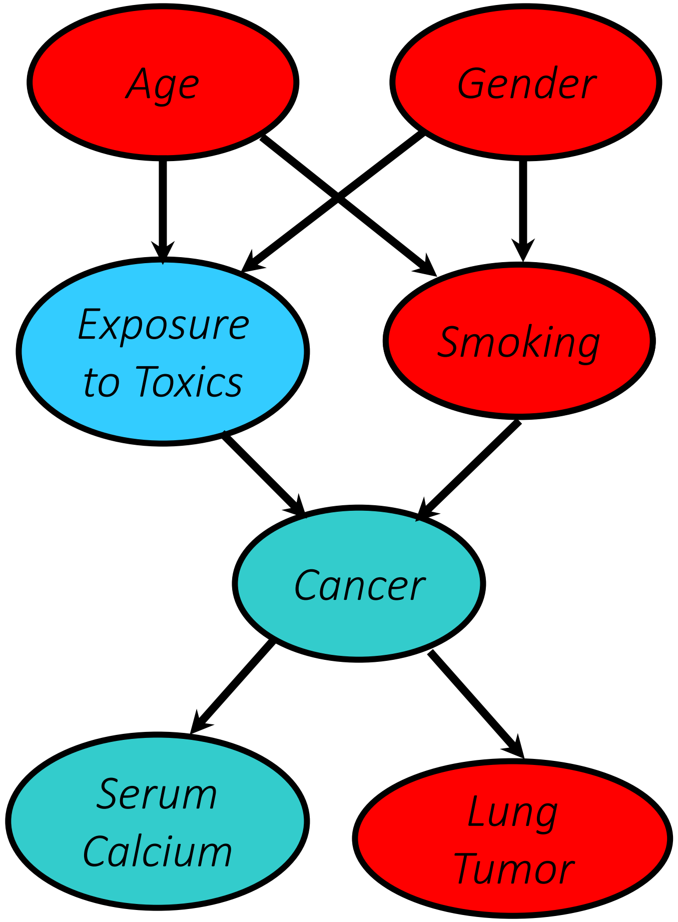
Prediction and Diagnosis



How likely is an elderly male patient with high Serum Calcium to have malignant cancer?

P(C=malignant | Age>60, Gender= male, Serum Calcium = high)

Explanation



- If we see a lung tumor, the probability of heavy smoking and of exposure to toxics both go up.

- If we then observe heavy smoking, the probability of exposure to toxics goes back down.

Decision Making

- Today’s weather forecast might be either sunny, cloudy or rainy

Should you take an umbrella when you leave?

- Your decision depends only on the forecast
- The forecast “depends on” the actual weather
- Your satisfaction depends on your decision and the weather
- Assign a utility to each of four situations: (rain|no rain) x (umbrella, no umbrella)

Decision Making

- Extend the BBN framework to include two new kinds of nodes: Decision and Utility

- A Decision node computes the expected utility of a decision given its parent(s), e.g., forecast, an a valuation

- A Utility node computes a utility value given its parents, e.g. a decision and weather

- We can assign a utility to each of four situations: (rain|no rain) x (umbrella, no umbrella)
- The value assigned to each is probably subjective

Association Rules Learning

Adopted from ‘Data Mining Concepts and Bing Liu, UIC

Introduction

- It is an important data mining model studied extensively by the database and data mining community.
- Assume all data are categorical.
- No good algorithm for numeric data.
- Initially used for Market Basket Analysis to find how items purchased by customers are related.

Bread $\rightarrow$ Milk [sup = 5%, conf = 100%]

Introduction

- Frequent pattern: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- First proposed by Agrawal, Imielinski, and Swami in the context of frequent itemsets and association rule mining
- Motivation: Finding inherent regularities in data
  - What products were often purchased together?— Beer and diapers?!
  - What are the subsequent purchases after buying a PC?
  - What kinds of DNA are sensitive to this new drug?
  - Can we automatically classify web documents?
- Applications
  - Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, and DNA sequence analysis.

Introduction

- Freq. pattern: An intrinsic and important property of datasets
- Foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: discriminative, frequent pattern analysis
  - Cluster analysis: frequent pattern-based clustering
  - Data warehousing: iceberg cube and cube-gradient
  - Semantic data compression: fascicles
  - Broad applications

Data

$I = \{i_1, i_2, \ldots, i_m\}$: a set of *items*.

Transaction t :

t a set of items, and $t \subseteq I$.

Transaction Database T: a set of transactions $T = \{t_1, t_2, \ldots, t_n\}$.

Examples

- Market basket transactions:
  - t1: {bread, cheese, milk}
  - t2: {apple, eggs, salt, yogurt}
  - … …
  - tn: {biscuit, eggs, milk}
- Concepts:
  - An *item*: an item/article in a basket
  - *I*: the set of all items sold in the store
  - A ***transaction***: items purchased in a basket; it may have TID (transaction ID)
  - A ***transactional dataset***: A set of transactions

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

Example

A text document data set. Each document is treated as a "bag" of keywords

doc1: Student, Teach, School

doc2: Student, School

doc3: Teach, School, City, Game

doc4: Baseball, Basketball

doc5: Basketball, Player, Spectator

doc6: Baseball, Coach, Game, Team

doc7: Basketball, Team, City, Game

Rules

- A transaction *t* contains *X*, a set of items (itemset) in *I*, if $X \subseteq t$.

- An association rule is an implication of the form:

$$X \rightarrow Y, \text{ where } X, Y \subset I, \text{ and } X \cap Y = \emptyset$$

- An itemset is a set of items.
  - E.g., X = {milk, bread, cereal} is an itemset.

- A *k*-itemset is an itemset with *k* items.
  - E.g., {milk, bread, cereal} is a 3-itemset

Rule Strength

Support: The rule holds with support *sup* in *T* (the transaction data set) if sup% of transactions contain $X \cup Y$.

$$sup = \Pr(X \cup Y).$$

Confidence: The rule holds in *T* with confidence *conf* if *conf*% of transactions that contain *X* also contain *Y*.

$$conf = \Pr(Y \mid X)$$

An association rule is a pattern that states when *X* occurs, *Y* occurs with certain probability.

Support and Confidence

Support count: The support count of an itemset *X*, denoted by *X.count*, in a data set *T* is the number of transactions in *T* that contain *X*.

Assume *T* has *n* transactions.

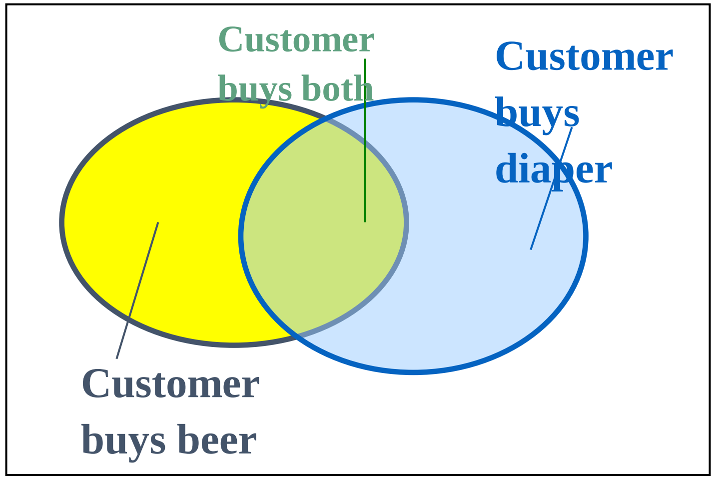
Then,

$$support = \frac{(X \cup Y).count}{n}$$

$$confidence = \frac{(X \cup Y).count}{X.count}$$

Support and Confidence

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



Find all the rules $X \rightarrow Y$ with minimum support and confidence

support, *s*, probability that a transaction contains $X \cup Y$

confidence, *c*, conditional probability that a transaction having X also contains *Y*

Let minsup = 50%, minconf = 50%

Freq. Pat.: Beer:3, Nuts:3, Diaper:4, Eggs:3, {Beer, Diaper}:3

- Association rules: (many more!)
  - *Beer* → *Diaper* (60%, 100%)
  - *Diaper* → *Beer* (60%, 75%)
  - Beer → Eggs (20%, 1/3=33%)
  - Nuts, Eggs → Milk (2/5=40%, 2/2=100%)
  - Nuts → Eggs (2/5=40%, 2/3=67%)

Target

- **Goal:** Find all rules that satisfy the user-specified *minimum support* (minsup) and *minimum confidence* (minconf).
- **Key Features**
  - Completeness: find all rules.
  - No target item(s) on the right-hand-side
  - Mining with data on hard disk (not in memory)

Patterns

- A long pattern contains a combinatorial number of sub-patterns, e.g., $\{a_1, \ldots, a_{100}\}$ contains $\binom{100}{1} + \binom{100}{2} + \cdots + \binom{100}{100} = 2^{100} - 1 = 1.27*10^{30}$ sub-patterns!
- Solution: *Mine closed patterns and max-patterns instead*

- An itemset X is closed if X is *frequent* and there exists *no super-pattern* Y ɔ X, *with the same support* as X (proposed by Pasquier, et al.)

- An itemset X is a max-pattern if X is frequent and there exists no frequent super-pattern Y ɔ X (proposed by Bayardo)

- Closed pattern is a lossless compression of freq. patterns
  - Reducing the # of patterns and rules

Patterns

Exercise.

DB = {<a_1, …, a_{100}>, < a_1, …, a_{50}>}

Min_sup = 1.

What is the set of closed itemset?

<a_1, …, a_{100}>: 1

< a_1, …, a_{50}>: 2

What is the set of max-pattern?

<a_1, …, a_{100}>: 1

What is the set of all patterns?

!!

Apriori Algorithm

- The downward closure property of frequent patterns
  - Any subset of a frequent itemset must be frequent
  - If **{beer, diaper, nuts}** is frequent, so is **{beer, diaper}** i.e., every transaction having {beer, diaper, nuts} also contains {beer, diaper}

Apriori Algorithm

- Key idea: The apriori property (downward closure property): any subsets of a frequent itemset are also frequent itemsets
- Apriori pruning principle: If there is any itemset which is infrequent, its superset should not be generated/tested! (Agrawal & Srikant & Mannila, et al.)
- A frequent *itemset* is an itemset whose support is ≥ minsup.
- Method:
  - Initially, scan DB once to get frequent 1-itemset
  - Generate length (k+1) candidate itemsets from length k frequent itemsets
  - Test the candidates against DB
  - Terminate when no frequent or candidate set can be generated

Apriori Algorithm

C_k: Candidate itemset of size k

L_k : frequent itemset of size k

L_1 = {frequent items};

for (k = 1; L_k !=$\varnothing$; k++) **do begin**

  C_{k+1} = candidates generated from L_k;

  for each transaction t in database do

    increment the count of all candidates in C_{k+1} that are contained in t

  L_{k+1} = candidates in C_{k+1} with min_support

  end

return $\cup_k L_k$;

Apriori Algorithm

```
Function candidate-gen(F_{k-1})
        C_k ← ∅;
        forall f_1, f_2 ∈ F_{k-1}
                with f_1 = {i_1, … , i_{k-2}, i_{k-1}}
                and f_2 = {i_1, … , i_{k-2}, i'_{k-1}}
                and i_{k-1} < i'_{k-1} do
            c ← {i_1, …, i_{k-1}, i'_{k-1}};          // join f_1 and f_2
            C_k ← C_k ∪ {c};
            for each (k-1)-subset s of c do
                if (s ∉ F_{k-1}) then
                    delete c from C_k;                // prune
            end
        end
        return C_k;
```

π.χ. {Α,Β,Γ} & {Α,Β,Δ} → {Α,Β,Γ,Δ}

Apriori Algorithm

- How to generate candidates?
  - Step 1: self-joining L_k
  - Step 2: pruning
- Example of Candidate-generation
  - L_3={*abc, abd, acd, ace, bcd*}
  - Self-joining: L_3*L_3
    - *abcd* from *abc* and *abd*
    - *acde* from *acd* and *ace*
  - Pruning:
    - *acde* is removed because *ade* is not in L_3
  - C_4 = {*abcd*}

Apriori Algorithm

$F_3 = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 3, 5\}, \{2, 3, 4\}\}$

After join

$C_4 = \{\{1, 2, 3, 4\}, \{1, 3, 4, 5\}\}$

After pruning:

$C_4 = \{\{1, 2, 3, 4\}\}$

because {1, 4, 5} is not in F_3 ({1, 3, 4, 5} is removed)

Apriori Algorithm

$Sup_{min} = 2$

$Conf_{min} = 75\%$

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan →

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan ←

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan →

L_3

Itemset	sup
{B, C, E}	2

B → C,E (2/4, 2/3=67%)
C → B,E (2/4, 2/3=67%)
E → B,C (2/4, 2/3=67%)
B,C→ E (2/4, 2/2=100%)
B,E→ C (2/4, 2/3=67%)
C,E → B (2/4, 2/2=100%)

Apriori Algorithm

- Frequent itemsets ≠ association rules
- One more step is needed to generate association rules

- For each frequent itemset *X*,
- For each proper nonempty subset *A* of *X*,
  - Let *B* = X - *A*
  - A → B is an association rule if
    - Confidence(A → B) ≥ minconf,
    - support(A → B) = support(A∪B) = support(X)
    - confidence(A → B) = support(A ∪ B) / support(A)

Apriori Algorithm

- Suppose {2,3,4} is frequent, with sup=50%
  - Proper nonempty subsets: {2,3}, {2,4}, {3,4}, {2}, {3}, {4}, with sup=50%, 50%, 75%, 75%, 75%, 75% respectively
  - These generate these association rules:
    - 2,3 $\rightarrow$ 4, confidence=100%
    - 2,4 $\rightarrow$ 3, confidence=100%
    - 3,4 $\rightarrow$ 2, confidence=67%
    - 2 $\rightarrow$ 3,4, confidence=67%
    - 3 $\rightarrow$ 2,4, confidence=67%
    - 4 $\rightarrow$ 2,3, confidence=67%
    - All rules have support = 50%

Example

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Scan *D* for count of each candidate →

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count →

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2 candidates from L_1 →

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Scan *D* for count of each candidate →

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
~~{I1, I4}~~	~~1~~
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
~~{I3, I4}~~	~~0~~
~~{I3, I5}~~	~~1~~
~~{I4, I5}~~	~~0~~

Compare candidate support count with minimum support count →

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

Generate C_3 candidates from L_2 →

C_3

Itemset
{I1, I2, I3}
{I1, I2, I5}

Scan *D* for count of each candidate →

C_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Compare candidate support count with minimum support count →

L_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Example

min_sup=3

Transaction	Items
T1	I1, I2, I3, I4, I5, I6
T2	I7, I2, I3, I4, I5, I6
T3	I1, I8, I4, I5
T4	I1, I9, I0, I4, I6
T5	I0, I2, I4, I5

Example

Transaction	Items
T1	I1, I2, I3, I4, I5, I6
T2	I7, I2, I3, I4, I5, I6
T3	I1, I8, I4, I5
T4	I1, I9, I0, I4, I6
T5	I0, I2, I4, I5

Items	Support
I0	2
I1	3
I2	3
I3	2
I4	5
I5	4
I6	3
I7	1
I8	1
I9	1

Example

Transaction	Items
T1	I1, I2, I3, I4, I5, I6
T2	I7, I2, I3, I4, I5, I6
T3	I1, I8, I4, I5
T4	I1, I9, I0, I4, I6
T5	I0, I2, I4, I5

Item	Support
I1	3
I2	3
I4	5
I5	4
I6	3

Example

Transaction	Items
T1	I1, I2, I3, I4, I5, I6
T2	I7, I2, I3, I4, I5, I6
T3	I1, I8, I4, I5
T4	I1, I9, I0, I4, I6
T5	I0, I2, I4, I5

Item	Support
I1	3
I2	3
I4	5
I5	4
I6	3

Items	Support
I1 I2	1
I1 I4	3
I1 I5	2
I1 I6	2
I2 I4	3
I2 I5	3
I2 I6	2
I4 I5	4
I4 I6	3
I5 I6	2

Example

Transaction	Items
T1	I1, I2, I3, I4, I5, I6
T2	I7, I2, I3, I4, I5, I6
T3	I1, I8, I4, I5
T4	I1, I9, I0, I4, I6
T5	I0, I2, I4, I5

Items	Support
I1 I4	3
I2 I4	3
I2 I5	3
I4 I5	4
I4 I6	3

Items	Support
I2 I4 I5	3
I4 I5 I6	2

Example

Item	Support
I1	3
I2	3
I4	5
I5	4
I6	3

Items	Support
I1 I4	3
I2 I4	3
I2 I5	3
I4 I5	4
I4 I6	3

Items	Support
I2 I4 I5	3

I2→ I4 I5 Conf: 3/3=100%
I4 → I2 I5 Conf: 3/5=60%
I5 → I2 I4 Conf: 3/4=75%
I2 I4 → I5 Conf: 3/3=100%
I2 I5 → I4 Conf: 3/3=100%
I4 I5 → I2 Conf: 3/4=75%

Example

min_sup=2

Transaction	Items
T1	M1, M2, M5
T2	M2, M4
T3	M2, M3
T4	M1, M2, M4
T5	M1, M3
T6	M2, M3
T7	M1, M3
T8	M1, M2, M3, M5
T9	M1, M2, M3

Example

Transaction	Items
T1	M1, M2, M5
T2	M2, M4
T3	M2, M3
T4	M1, M2, M4
T5	M1, M3
T6	M2, M3
T7	M1, M3
T8	M1, M2, M3, M5
T9	M1, M2, M3

Items	Support
M1	6
M2	7
M3	6
M4	2
M5	2

Items	Support
M1 M2	4
M1 M3	4
M1 M4	1
M1 M5	2
M2 M3	4
M2 M4	2
M2 M5	2
M3 M4	0
M3 M5	1
M4 M5	0

Items	Support
M1 M2 M3	2
M1 M2 M5	2
M1 M3 M5	1
M2 M3 M4	0
M2 M3 M5	1
M2 M4 M5	0

Items	Support
M1 M2 M3 M5	1

Example

Items	Support
M1 M2 M3	2
M1 M2 M5	2

Items	Support
M1 M2	4
M1 M3	4
M1 M5	2
M2 M3	4
M2 M4	2
M2 M5	2

	Confidence
$M1 \wedge M2 \Longrightarrow M3$	2/4=0.5
$M2 \wedge M3 \Longrightarrow M1$	2/4=0.5
$M1 \wedge M3 \Longrightarrow M2$	2/4=0.5
$M1 \wedge M2 \Longrightarrow M5$	2/4=0.5
$M1 \wedge M5 \Longrightarrow M2$	2/2=1.0
$M2 \wedge M5 \Longrightarrow M1$	2/2=1.0