KDD/DM (Technical)

KDD DM

DM - -selecting data set/sub-set,
- applying algo.,
- using data for predictive modeling analysis

* DW - data warehousing = implementation of enterprise data in unified structure (cube)
* OLAP - ? why - faster query response

* MS BI perspective (many data sources, ...)
* BIDS
* client tools - excel, SP, SSRS, SSIS

DM processing

Training data
↓
DM Engine
↓
Mining Model — Predictive models

* one way

Mining model, data to be predicted
↓
data — with predictions

* other way

Steps for building DM model

1. Model definition (define columns for cases: visually (BIDS), using DMX, or from PMML)
2. Model Training (feed lots of data from a real DB, or from a system log)
3. Model Testing (testing data must be different from training)
4. Model Use (exploration and prediction)
   - use the model on new data to predict outcomes
5. Model update (monthly, weekly, nightly, ... and re-test)

Mining structure
- ~~dad~~ describes data to be mined
  - columns from a data source and their:
    - data type
    - content type
- ~~contains mining model~~
- contains Mining Models (often we build several different models in one structure)
- holds training data, known as Cases (if required)
- holds ~~training~~ testing data, known as Holdout (in SQL 2008)

DM model
- container of patterns discovered by DM Algorithm amongst the training (cases) (data)
  - a table containing patterns
  - expressed by visualizers
- specifies usage of columns already defined in the mining structure

---

Cases: (the things we study)
- case - set of columns (attributes) you want to analyse
  ex; age, gender, region, annual spending
- case key - unique ID of a case
- a column has:
  * data type
  * content type
  * and optionally:
    * distribution
    * discretization
    * Related columns
    * flags (e.g. NOT NULL)

Column distribution ③

- if you know the distribution of your data (you should), indicate it:
  - Normal (typical Gaussian bell-curve)
  - Log Normal (most values at the "beginning" of the scale
  - Uniform (flat line - equally likely or perfectly random)
- other distributions can exist, but you can not indicate them - algorithm will work fine

---

```
Create Mining Model <name> M1
{ Attributes
  - - - -

} using Algol
```

---

data training

```
INSERT INTO [Mining Model | Mining Structure] <name>
  [ (attributes) ]
  < source data >
        → data query / DMX Query / MDX Query
          Stored Procedure call / Rowset Parameter
```

---

predict

```
Select [TOP <count>]
<expression-list> from <model>
[
  [Natural] PREDICTION JOIN
    <source data> AS <alias>
    [ON <column-mapping> ]
    [Where <filter expression> ]
  [order by <expression> ]
]
```

DM Algo.

Decision Tree — finds the odds of an outcome based on values in a training set

Association Rules — identifies relationships between cases

Clustering — classifies cases into distinctive groups based on any attribute sets

Naive Bayes — clearly shows the differences in a particular variable for various data elements

Sequence clustering — groups or clusters data based on a sequence of previous events

Time series — Analyzes and forecasts time-based data combining the power of ARTXP (developed by MS Research) for short-term predictions with ARIMA (in SQL2008) for long-term accuracy.

Neural Nets — seeks to uncover non-~~trivial~~ intuitive relationships in data

Linear Regression — determines the relationship between columns in order to predict an outcome

Logistic regression — determines the relationship between columns in order to evaluate the probability that a column will contain a specific state.

DM Lift chart for comparising algorithms

Logistic regression for ~~Bide~~ association rule

deciding ~~Kans~~ merchandise

Scatter plot

DM

- pattern discovery
- Intelligent grouping
- Predictions (probability) - (values; series Event, Time-based events)

Algorithm

- clustering = grouping (~~segmenting~~ divide data into clusters segments)
- Classification = predicting a specific value
- association = correlation (market basket)
- regression = forecasting a continuous number
- sequences = process and route (clickstream)
- deviation = outliers (exception, fraud)

MS DM

Mining structures (containers to keep DM models)

Naive Bayes - simple, starting point
- Basic groupings
- Discrete (unique) content only

Decision Trees - most common, groups
- powerful viewer
- discrete or continuous attributes

Time series - forecasting
- time series viewer

(1)

Clustering - grouping

Sequence clustering - grouping + sequences
- ~~state tran~~ - state transitions

Association rules - market basket
- itemsets

(2)

Neural Network - solves the difficult problems
- finds invisible patterns
- takes significant processing overhead

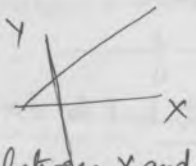
Linear/logistical regression (Determines best straight line through series of points)

Regression Analysis:-

$\boxed{\text{Dataset}} \rightarrow \boxed{\text{RA}} \rightarrow \boxed{\text{predictive model}}$

~~(output)~~ X = predictor (output) variable

Y = response variable (input)



① linear function, $Y = aX + b$ → linear (a relation between X and Y)

~~Score function = $\sum($ + (discrepancy)$)^2$~~

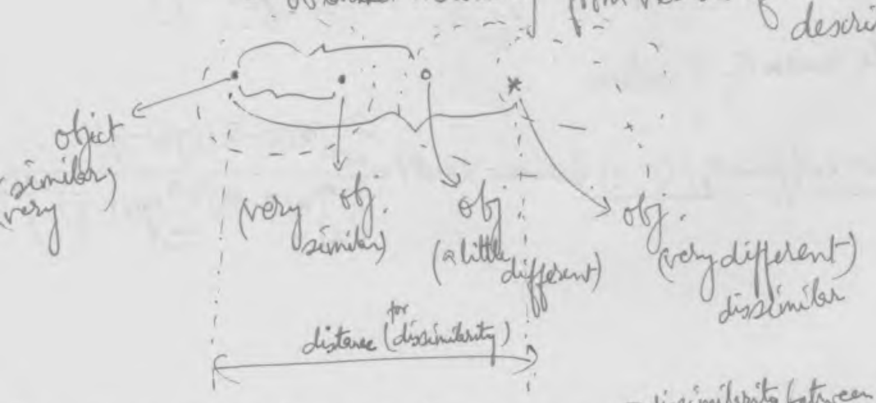
② Score function (squared error) $= \sum_{i=1}^{n} (y(i)-\hat{y}(i))^2$ → { predicting n "target" values $y(i)$, $1 \leq i \leq n$, $\hat{y}(i)$ predictions for each }

other score functions { e.g., least squares, classification accuracy, likelyhood, misclassification rate

Distance measures:-

(similarity measures between objects / dissimilarity measures ") = proximity

~~often~~
- obtained directly from the objects
- obtained indirectly from vectors of measurements/characteristics describing each object.



① $s(i,j)$, similarity between i and j objects $= 1 - d(i,j)$ → dissimilarity between objects i and j

② $d(i,j) = \sqrt{2(1-s(i,j))}$

metric (dissimilarity measure) with conditions { ① $d(i,j) \geq 0$ and $d(i,j) = 0$ if and only if $i = j$; ② $d(i,j) = d(j,i)$ for all i, j; ③ $d(i,j) \leq d(i,k) + d(k,j)$ for all i, j, k (triangle inequality) }

Euclidian distance, $d_E(i,j)$ between i and j = $\left(\sum_{k=1}^{p}\left(x_k(i)-x_k(j)\right)^2\right)^{1/2}$ ①

- for n data object with p-realvalued measurements on each object.
- for vector of observations for the ith object by $x(i) = (x_1(i), x_2(i), \cdots, x_p(i))$, $1 \leq i \leq n$

* there is assumption of some degree of commensurability between different variables.

standard deviation (for the kth variable, X_k) $= \hat{\sigma}_k = \left(\frac{1}{n}\sum_{i=1}\left(x_k(i)-\mu_k\right)^2\right)^{1/2}$

μ_k = mean for variable X_k

sample mean $\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_k(i)$

$x'_k = \left(x_k / \hat{\sigma}_k\right)$ removes effect of scale as captured by $\hat{\sigma}_k$

weighted Euclidian distance measure, $d_{WE}(i,j)$

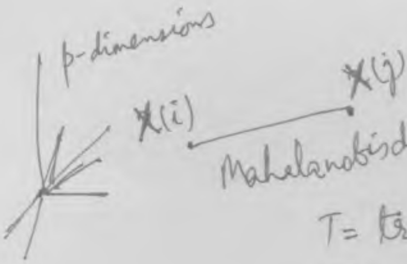
$$= \left(\sum_{k=1}^{p} w_k\left(x_k(i)-x_k(j)\right)^2\right)^{1/2}$$

sample covariance between X and Y

$$Cov(X,Y) = \frac{1}{n}\sum_{i=1}^{n}\left(x(i)-\bar{x}\right)\left(y(i)-\bar{y}\right)$$

$\bar{x}$ = sample mean of X values
$\bar{y}$ = " " Y "

sample correlation coefficient, $\rho(X,Y)$ between X and Y $= \dfrac{\sum_{i=1}^{n}\left(x(i)-\bar{x}\right)\left(y(i)-\bar{y}\right)}{\left(\sum_{i=1}^{n}\left(x(i)-\bar{x}\right)^2\sum_{i=1}^{n}\left(y(i)-\bar{y}\right)^2\right)^{1/2}}$



Mahalanobis distance, $d_{MH}(i,j) = \left((x(i) - x(j))^T \Sigma^{-1} (x(i) - x(j)) \right)^{1/2}$

T = transpose matrix

Σ = $p \times p$ sample covariance matrix

Σ^{-1} = standardizes data relative to Σ

✓ Minkowski space

✓ L_λ metric

✓ Manhattan / city-block metric

* Jaccard coefficient
* dice coefficient

date = information

small data   large data   pieces of information
a piece of information

example:
man is selfish by nature. → Philosopher ~~brain~~ data mining information

brain

knowledge discovery process

contain data mining

interacted society

neurons interact strike

Biological data mining

implementation of logic   algorithm
in terms of pseudocode
or ~~seed code~~
programming code

gathers all the pieces of information and mine them a
to discover knowledge

human brain

small but efficient miner

for large data, not repeated, computer, larger RAM

in □ow

Biological limitation
lower RAM / memory space

cluster
stars
bunches

Data mining?

- finding interesting & structure in data
  - structure: (refers to statistical patterns, predictive models, hidden relationships)
  - interesting: ?

ex: - predictive modeling (classification, regression)
- segmentation (data clustering)
- affinity (~~association~~) (summarization) - relations between fields, association, visualization

beyond data analysis
- * Scaling analysis to large databases
  - how to deal with data without having to move it out?
  - are there abstract primitive accesses to data, in database systems, that can provide mining algorithms with the information to drive the search for patterns?
- * automated search
  - enumerate and ~~search~~. create numerous hypothesis
  - fast search
  - useful data reductions
  -

* more emphasis on understandable models

? high dimensions?

DM

Defining the goals (use models)

① Segmentation — to segment customers by profitability and market potential.

② Profile analysis — analysis of (prospect's customer's) profile like average age, gender, length of residence, age of customer's relationship.

③ Response — receiving response of a customer by offering a product/service.

④ Risk — financial risk (grant loan), credit card risk, risk of fraud

⑤ Activation — (predict response of customers, ~~and~~ predict activation given by response of customer)

⑥ Cross-sell and up-sell — (predict the probability or value of a current customer buying a different product/service from same company,)
(predict the probability or value of customer buying more of the same product/service)

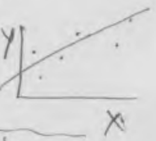
⑦ Attrition/churn — act of customers switching companies to take advantage of better deal

⑧ Net Present value — NPV predict overall profitability of a product for a predetermined length of time.

⑨ Lifetime Value — LTV predicts overall ~~profit~~ profitability of customer/business for a predetermined length of time.
(customer lifetime value)

Modeling methodology (for predictive/descriptive models)

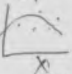
① linear regression (statistical techniques) — quantifies relationship between two continuous variables,
① dependent Y ② independent X (predictive variable)



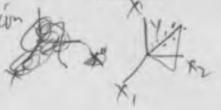
* finding a line, through the data, that minimizes the squared error from each point.

* R-square (measure of strength of relationship) — measures the amount of overall variation in data that is explained by the model %

✓ Nonlinear (curvilinear) regression

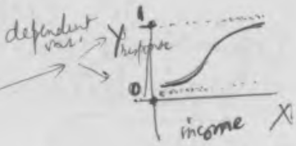


✓ multiple linear regression



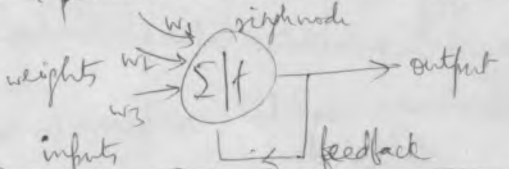
Logistic Regression –

* similar to linear regression
* dependent variable is non-continuous (discrete/categorical) logical
* based on statistical distribution



Neural network –

* the process is one of pattern recognition and error minimization.
* made up of nodes that are arranged in layers
* data is split into training and testing data sets. A third group is held out for final validation. Then weights or inputs are assigned to each of the nodes in the first layer. By iterations, the inputs are processed through the system and compared to the actual value. The error is measured and fed back to adjust the weights. The process ends when a predetermined minimum error level is reached.



Classification tree (decision tree) – to sequentially partition the data to maximize the differences in the dependent variable.)

* classify data into distinct groups/branches that create the strongest separation in the values of the dependent variable.

Associations and item-sets:

rule; if X then Y

$X \rightarrow Y$

[exception; ~~if then~~... if _ and _ then _ except if _ then _

if _ then _ else _

for any rule if $X \rightarrow Y \Rightarrow Y \rightarrow X$ then X and Y are called an interesting item-set

coverage (predict correctly) = support for a rule R = $\frac{\text{no. of occurrences of } R}{\text{total no. of all occurrences of all rules}}$

accuracy = confidence of a rule $X \xrightarrow{\text{for}} Y$ = $\frac{\text{no. of occurrences of } Y \text{ given } X\ (X \rightarrow Y)}{\text{no. of all other occurrences given } X}$

Apriori Algo: - based on combination
- min. support & min. confidence

Mining association rules using Apriori —
- use Apriori to generate frequent itemsets of different size
- at each iteration, divide ~~eq~~ each frequent itemset (I) into LHS and RHS

LHS $\rightarrow$ RHS

- confidence of such rule = support of ~~rule~~ itemset (I) / support (LHS)
- discard all rules whose confidence is ~~lo~~ less than min confidence.

Classification Techniques : (predefined classes) based on supervised learning

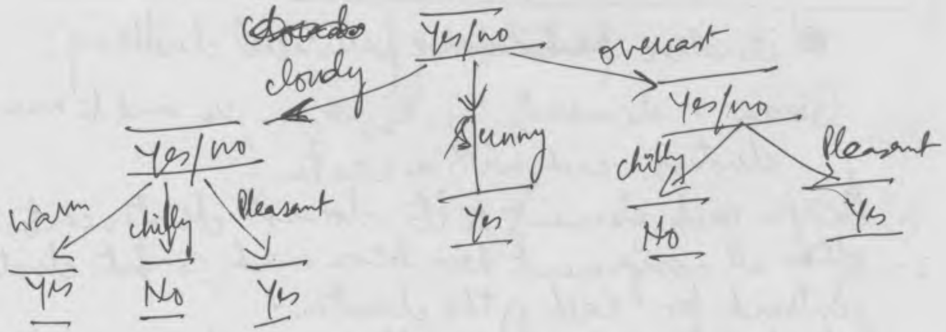
decision tree identification:
(Hunt's Algo. method for ..)

weather	Temp	play?
sunny	30	Yes
overcast	15	No
sunny	16	Yes
–	–	–
	–	–

(set these three into classes) Warm / chilly / Pleasant

So, we got, whenever
sunny → Yes ~~X~~
cloudy → Yes/No
overcast → Yes/No } make tree

So, by considering three classes,



Clustering techniques (unsupervised learning)

→ clustering partitions data set into clusters or equivalence classes (but not the predefined classes)

→ similarity among members of a class more than similarity among members across classes.

→ ~~similarity~~ similarity measures: Euclidian distance or other application specific measures.

- Nearest neighbour clustering Algo:

Given n elements $x_1, x_2, x_3, x_4, x_5, \ldots, x_n$ and threshold t.

① $j \leftarrow 1$, $k \leftarrow 1$, clusters = { }

② Repeat

   ① find nearest neighbour ~~(similarity)~~ of x_j

   ② let nearest neighbour ~~be~~ in cluster m

   ③ if distance to nearest neighbour > t, then create a new cluster and $k \leftarrow k+1$; else assign x_j to cluster m.

   ④ $j \leftarrow j+1$

③ until $j > n$

- iterative ~~partitioning~~ partitional clustering:

Given n elements $x_1, x_2, x_3 \ldots, x_n$ and k number of clusters, each with a center.

1. Assign each element to its closest cluster center
2. After all assignments have been made, compute cluster centroids for each of the cluster.
3. Repeat above two steps with new centroids untill algo. converges.

Mining streaming data :

example : stock market quotes

Running mean :

let n = no. of items read so far

avg = running ~~av~~ average calculated so far

On reading the next number num :

$avg \leftarrow (n * avg + num)/(n+1)$

$n \leftarrow n+1$

Running variance :

$var = \sum (num + avg)^2$

$= \sum num^2 + 2 * \sum num * avg + \cancel{2} avg^2$

let $A = \sum num^2$ of all numbers read so far

$B = 2 * \sum num * avg$ of all numbers read so far

$C = \sum avg^2$ of all numbers read so far

avg = average of numbers read so far

n = no. of numbers read so far

$A \leftarrow A + num^2$

$B \leftarrow B + 2 * avg * num$

$C \leftarrow C + avg^2$

$var = A + B + C$

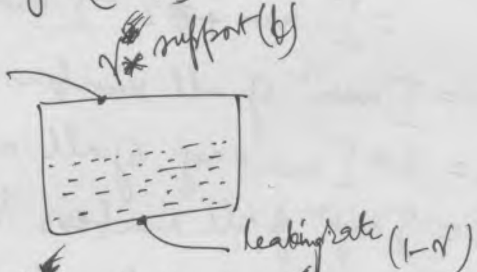
Mining streaming data

γ-consistency:

→ let streaming data be in the form of "frames" (events) ~~elements~~
where each frame (event) comprises of one or more data elements.

→ support for data element k within a frame is defined as (# occurrences of k)/(# elements in frame)

→ γ-consistency for data element k is the "sustained" support for k over all frames read so far, with a "leakage" of $(1-\gamma)$



$$level_t(k) = (1-\gamma) * level_{t-1}(k) + \gamma * support(k)$$

mining sequence data (ordered data)

A sequence is a list of itemsets of finite length.

{pen, pencil, ink} {pencil, ink} {ink, eraser} {ruler, pencil}

(itemsets)

* order of items within an itemset does not matter but order of itemsets matter

* A subsequence is a sequence with some itemsets deleted.

let a sequence $S' = \{a_1\}\{a_2\}\{a_3\} \ldots \{a_n\}$ is said to be contained within another sequence S, if

S contains a subsequence $\{b_1\}\{b_2\}\{b_3\} \ldots \{b_m\}$

such that $a_1 \subseteq b_1$, $a_2 \subseteq b_2$, $a_3 \subseteq b_3, \ldots$

$a_n \subseteq b_m$

Apriori Algo. for sequence data :-

1. $L_1 \leftarrow$ set of all interesting 1-sequences
2. $k \leftarrow 1$
3. while L_k is not empty do
   1. generate all candidate $k+1$ sequences
   2. $L_{k+1} \leftarrow$ set of all interesting $k+1$ sequences
4. done

mining sequence data (ordered combination = permutation + self concatenation

example:

a b c d e
b d a e
a c b d
b e
e a b d a
a a a e
b a a e
c b d b
a b b a b
a b d e

min. support = 0.5

interesting 1-sequences:
a
b
d
e

Candidate 2-sequences:
aa, ab, ad, ae
ba, bb, bd, be
da, db, dd, de
ea, eb, ed, ee

min. support = 0.5
interesting 2-sequences:
ab, bd

Candidate 3-sequences: [by permuting ab, bd with a, b, d, e 1-sequence]
aba, abb, abd, abe,
aab, bab, dab, eab,
bda, bdb, bdd, bde,
bbd, dbd, ebd

interesting 3-sequences = { }

mining sequence data:

Language interface:
(input set of sequences ⇒ output finite state machine)

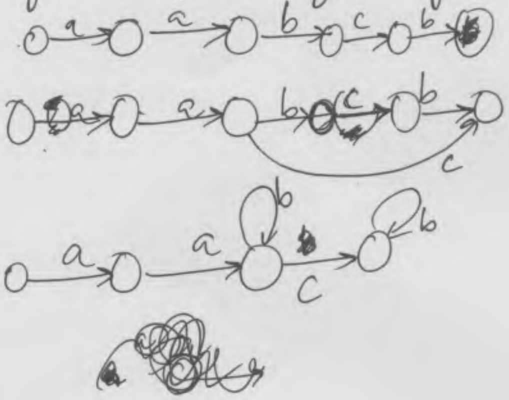
"shortest run generalization" Algo. by (Srinivasa & Spiliopoulou 2000)

aabcb

aac

~~aabc~~

aabc



~~ABD~~
ABCD ~~ABCD~~
~~ABCD~~
~~BCD~~

	A	B	C	D
1001	ink	pen	cheese	bag
1002	milk (E)	pen (B)	juice (F)	cheese (C)
1003	milk (E)		juice (F)	
1004	juice (F)	milk (E)	cheese (C)	

min. support = 50%
" confidence = 70%

~~$[A] = \frac{1}{4} = 25\%$~~
$[B] = \frac{2}{4} = 50\%$
$[C] = \frac{3}{4} = 75\%$
~~$[D] = \frac{1}{4} = 25\%$~~
$[E] = \frac{3}{4} = 75\%$
$[F] = \frac{3}{4} = 75\%$

$\frac{2}{3}$

$B \to C$ $\frac{2}{2}$

BCEF
BCEF
BCEF
BCEF
BCEF
BCEF
BCEF
BCEF
BCEF
BCEF
BCEF

BC 50% 100%
~~BE 25%~~
~~BF 25%~~
CE 50%
CF 50%
EF $\frac{3}{4}$ 75%

BC
CE
CF
EF

~~BCE~~
~~BCF~~
~~BCEF~~
CEF 50%
~~CEF~~

C → EF

GM   SNA   Multirelational DM

graph mining patterns: frequent substructures → how to: ① generate frequent substructure candidates. ② check frequency of each candidate)

for characterizing graph sets,
discriminating different groups of graphs,
classifying and clustering graphs,
building graph indices,
facilitating similarity search

~~Methods for mining frequent subgraphs~~

$G = (V, E)$ — a set of vertices; a set of edges

$E = \{\{(u,v)\} \mid u, v \in V\}$

basic idea: Let $V(g)$ be vertex set of graph,
$E(g)$ " edgeset " ",
Assume L be label function, maps a vertex or edge to label.

$f: V \Rightarrow R^+$

$g' \subset g$

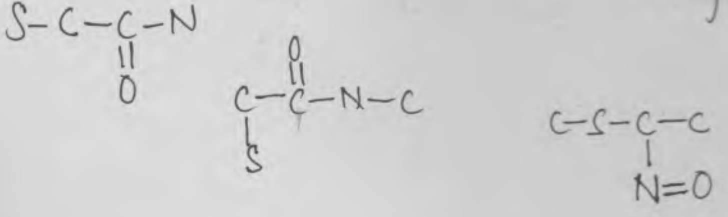
(g, graph is a subgraph of another graph g'), if there exists a subgraph isomorphism from g to g'.

Given a labeled graph data set, $D = \{G_1, G_2, \cdots G_n\}$
the support(g) or frequency(g) is percentage or number of graphs in D where g is a subgraph.

A frequent graph is a graph whose support is no less than a min. support threshold, min_sup.

Methods for mining frequent subgraphs

① Apriori-based Approach (search for frequent graphs)
(AprioriGraph) start with graphs of small size,
proceed in bottomup manner by generating candidates having an extra vertex, edge, path.



① AprioriGraph:

input { D, graph data set
min_sup, minimum support threshold

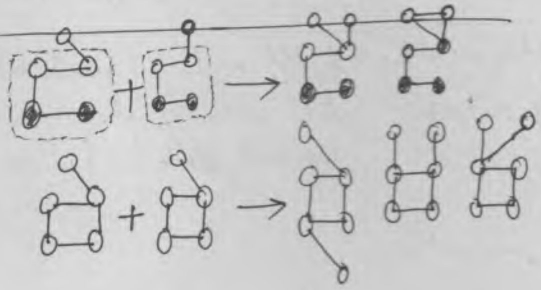
output { S_k, frequent substructure set of size k

method: { $S_1 \leftarrow$ frequent single-elements in the data set;
Call AprioriGraph(D, min_sup, S_k)

procedure: AprioriGraph(D, min_sup, S_k)

(1) $S_{k+1} \leftarrow \emptyset$
(2) for each frequent $g_i \in S_k$ do
(3) for each frequent $g_j \in S_k$ do
(4) for each size(k+1) graph g formed by the merge of g_i and g_j do
(5) if g is frequent in D and $g \notin S_{k+1}$ then
(6) insert g into S_{k+1};
(7) if $S_{k+1} \neq \emptyset$ then
(8) AprioriGraph(D, min_sup, S_{k+1})
(9) return

Algo { AGM
FSG
path-join method (edge-disjoint path)



② Pattern-Growth Approach

Input: { g, a frequent graph
D, a graph data set
min_sup, minimum support threshold

Output: { the frequent graph set, S

Method:

$S \leftarrow \emptyset$

call ~~call~~ PatternGrowthGraph(g, D, min_sup, S)

procedure PatternGrowthGraph(g, D, min_sup, S)

① if $g \in S$ then return;

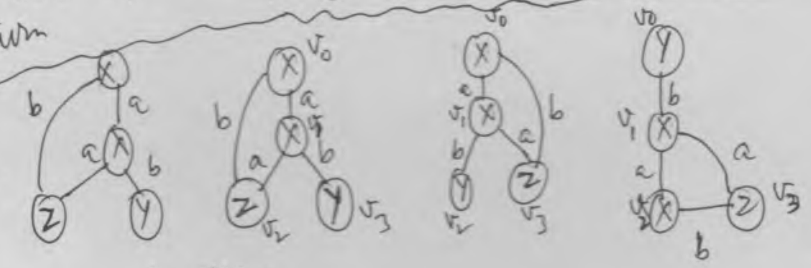
② else insert g into S

③ Scan D once, find all the edges e such that g can be extended to $g \diamond_x e$

④ for each frequent $g \diamond_x e$ do {

⑤ PatternGrowthGraph($g \diamond_x e$, D, min_sup, S) }

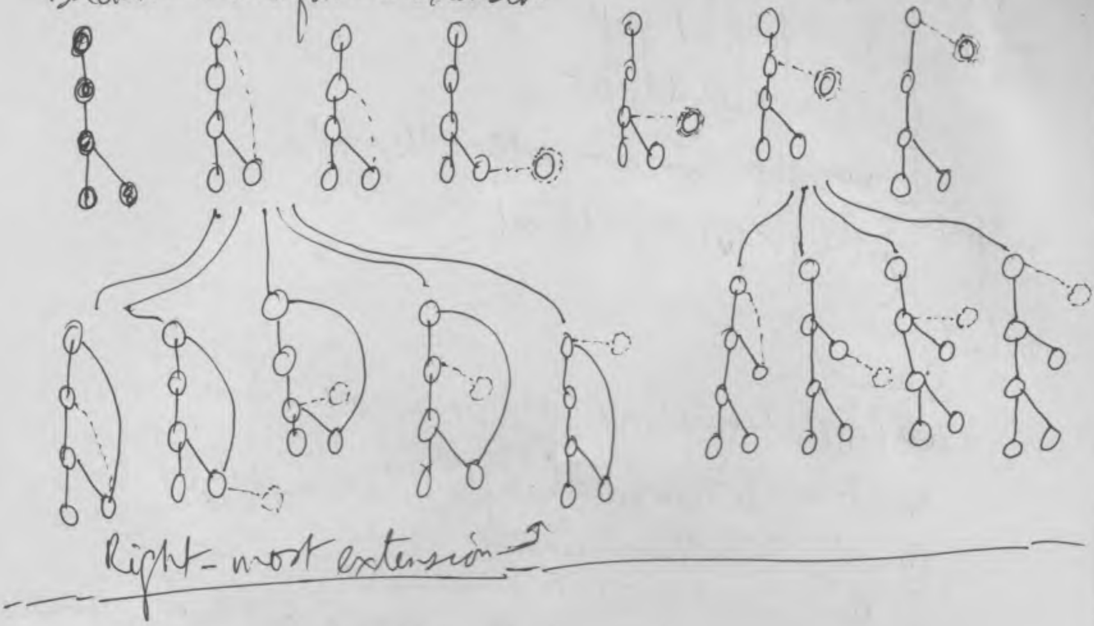
⑥ return

gSpan Algo. → 

to reduce the generating duplicate graphs in PatternGrowthGraph.

DFS subscripting (Depth First Search)

Backward and forward extension



Social Network Analysis (link analysis / link mining)

– a heterogeneous and multirelational data set represented by graph.

http://tam.cornell.edu/Strogatz.htm #pub
// www.nd.edu/~networks/publications.html #talks 0001

* structure always affects function

Characteristics – (by building graph generated models, then may be used to predict how a network may look in the future)

* if a hypothesis contradicts the generally accepted characteristics, this raises a flag as to the ~~q~~ questionable plausibility of the hypothesis. This can help detect abnormalities in existing graphs, which may indicate fraud, spam, or distributed denial of service (DDoS) attacks. Models of graph generation can also be used for simulations when real graphs are excessively large and ~~they~~ thus, impossible to collect (such as a very large network of friendships).

✓ ① examine the node's degree (no. of edges incident to each node)
② distances between a pair of nodes (shortest path length)
③ network diameter (max. ~~length~~ distance between pairs of nodes)
④ node-to node distances (the average distance between pairs)
⑤ effective diameter (~~i.e.~~ the min. distance, d, such that for at least 90% of the reachable node pairs, the path length is at most d)

Social network phenomena:

① densification power law (number of edges growing superlinearly in the number of nodes)
~~$e(t) \propto n(t)^a$~~ no. of edges → $e(t) \propto n(t)^a$ a → (1<a<2)
no. of nodes at time (t)

② Shrinking diameter (the effective diameter tends to decrease as the network grows)

③ Heavy-tailed out-degree and "bridging" in-degree distributions (the number of out-degrees for a node tends to follow a heavy-tailed distribution by observing the power law, $1/(n)^a$ where n = rank of node in the order of decreasing out degrees and typically 0<a<2. ~~⇒~~ smaller the value of a, heavier the tail. (preferential attachment model) "rich-get-richer"

Forest Fire model (new nodes attach to the network by "burning" through existing edges in epidemic fashion.)

parameters: forward burning probability, p
backward burning ~~probability~~ ratio, r

Let a new node 'v' arrives at time 't'. It attaches to G_t:

① it chooses an ambassador node, w at random, and forms a link to w.

② it ~~tosses~~ selects x links incident to w, where x is a random number that is binomially distributed with mean $(1-p)^{-1}$. It chooses from out-links and in-links of w but selects in-links with probability r times lower than out-links. Let $w_1, w_2, \cdots$ $\cdots w_x$ denote the nodes at the other end of the selected edges.

③ ~~Our~~ new node, v, forms out-links to $w_1, w_2, \cdots, w_x$ and then applies step 2 recursively to each of $w_1, w_2, \cdots, w_x$. Nodes cannot be visited a second time so as to prevent the construction from cycling. The process continues until it dies out.

* (Nodes with heavy-tailed out-degrees may serve as "bridges" that connect formerly disparate parts of the network, decreasing the network diameter.

$G_i(V, E_i)$, $i = 1, \cdots n$ where n is the number of relations, V is the set of nodes (objects), E_i is the set of edges with respect to the i-th relation.

Link Mining   How can we mine social networks?

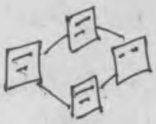
Traditional methods of machine learning & data mining, taking as input, a random sample of homogeneous objects from a single relation, may not be appropriate here. So, link mining came in. (it is a confluence of research in social networks, link analysis, hypertext and web mining, graph mining, relational learning, and inductive logic programming. It embodies descriptive and predictive modeling.

* By considering links (the relationships between objects), more information is made available to the mining process. This brings about several new tasks. Here, we list these tasks with examples:

① Link-based object classification (traditionally, objects are classified based on the attributes that describe them.

Link-based classification predicts the category of an object based not only on its attributes, but also on its links, and on the attributes of linked objects.

example1: web page classification (predicts the category of a web page based on word occurrence and hyperlink words/anchor text), both of which serve as attributes.



ex.2: In the bibliography domain, objects include papers, authors, institutions, journals and conferences. A classification task is to predict the topic of a paper based on word occurrence, citations, and cocitations (other papers that are cited within the paper), where the citations act as links.

ex.3: in epidemiology, predicting the disease type of a patient based on characteristics (e.g. symptoms) of the patient, and on characteristics of other people with whom the patient has been in contact.

② Object type prediction : (Predicts the type of an object, based on its attributes and its links, and on the attributes of objects linked to it.

ex: In the bibliographical domain, we may want to predict the venue type of a publication as either conference, journal, or workshop.

ex: In the communication domain, a similar task is to predict whether a communication contact is by e-mail, phone call, or mail.

③ Link type prediction : (predicts the type or purpose of a link, based on properties of the ~~link~~ objects involved.

ex: Given epidemiological data, for instance, we may try to predict whether two people who know each other are family members, coworkers, or acquaintances.

ex: we may want to predict whether there is an advisor-advisee relationship between two coauthors.

④ Predicting link existence : (to predict ~~about a link~~ whether a link exists between two objects or not.)

ex: predicting whether there will be a link between two web pages, and whether a paper will cite another paper.

ex: In epidemiology, we can try to predict with whom ~~~~ a patient came in contact.

⑤ Link cardinality estimation: There are two forms of link cardinality estimation.

(i) we may predict the number of links to an object (in-link)

· Similarly, the number of out-links can be used to identify web pages that act as hubs, where a hub is one or a set of web pages that point to many authoritative pages of the same topic.

(ii) the second ~~~~ form of link cardinality estimation (predicts the number of objects reached along a path ~~of~~ from an object. This is important in estimating the number of objects that will be returned by a query. (ex: web page) ~~~~

⑥ Object reconciliation : (to predict whether two objects are, in fact, the same, based on their attributes and links)

~~~~ This is common ~~~~ task in info. extraction, duplicate elimination, object consolidation, and citation matching / record linkage / identity uncertainty.

ex: ~~~~ predicting mirror sites.

ex: " for two apparent disease strains

⑦ Group detection : (predicts whether a set of objects belong to the ~~same~~ same group or cluster, based on their attributes and links.
(clustering)

ex: identification of web communities
~~~~

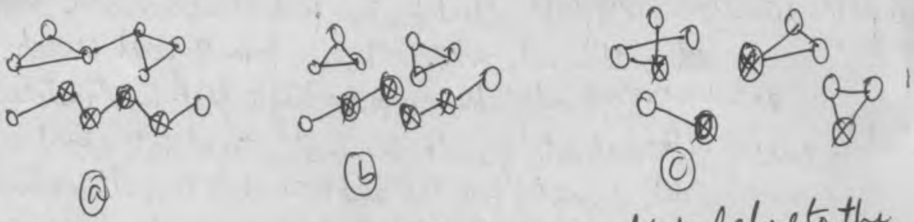
(8) Subgraph detection: (subgraph identification finds characteristic subgraphs within networks. This is a form of graph search.)
ex: discovery of subgraphs corresponding to protein structures.

(9) metadata mining: (metadata provide semi-structured data about unstructured data, ranging from text and web data to multimedia databases)
* useful for data integration tasks in many domains
ex: schema mapping; schema discovery; schema reformulation.

---

* Link prediction: what edges will be added to the network?
Approaches to link prediction have been proposed based on several measures for analyzing the "proximity" of nodes in a network.
general methodology: All methods assign a connection weight, score(X, Y) to pairs of nodes, X and Y, based on the given proximity measure and input graph, G. A ranked list in decreasing order of score(X,Y) is produced. This gives the ~~prediction~~ predicted new links in decreasing order of confidence. The predictions can be evaluated based on real observations on experimental data sets.
the simplest approach ranks pairs, ⟨X,Y⟩, by length of their shortest path in G. This embodies the small world notion that all individuals are linked through short chains. (Since the convention is to rank all pairs in order of decreasing score, here, score(X,Y) is defined as the negative of the shortest path length.) Several measures use neighborhood info. The simplest such measure is "common neighbors". The greater the number of neighbors that X and Y have in common, the more likely X and Y are to form a link in the future.

Multirelational Social network analysis:-
* homogeneous (same kind of relation) and heterogeneous (different kind of relations) links
* relation selection and extraction
*



Let a user requires that the four ⊗ objects belong to the same community and specifies this with a query.

- As per graphs, the relative importance of each of the three relations differs with respect to the user's information need. (a) is the most relevant to the user's need, (b) comes in second, (c) is noisy or negative in regards to the user's information need.

But different aspect of a user can vary the ~~outcome~~ result.

* In multirelational social network, community mining should be dependent on the user's query (or information need). A user's query can be very flexible.

An algorithm for relation extraction and selection was proposed, which models the task as an optimization problem.

for a Given V = a set of objects and a set of relations,
a set of graphs, $G_i(V, E_i)$, $i = 1, 2, \ldots, n$ (n = no. of relations)
E_i = set of edges (relations) with respect to i-th relation.

The weights on the edges can be naturally defined according to the relation strength of two objects. The algorithm characterizes each relation by a graph with a weight matrix. Let M_i denote the weight matrix associated with G_i, $i = 1, \ldots, n$. Each element in the matrix reflects the relation strength between a pair of objects in the relation.

suppose a hidden relation is represented by a graph $\hat{G}(V, \hat{E})$, and $\hat{M}$ denotes the weight matrix associated with $\hat{G}$. A user specifies her info. need as a query in the form of a set of labeled objects $X = [x_1, \ldots x_m]$ and $Y = [y_1, \ldots y_m]$ (where y_j is the label of x_j)

when such labeled objects indicate partial info. of hidden relation $\hat{G}$ →

the algo. aims at finding a linear combination of these weight matrices that can best approx. $\hat{G}$

http:// scholar.google.com/
http:// opac.dl.itc.u-tokyo.ac.jp/
http:// www.dl.itc.u-tokyo.ac.jp/gacos/

encyclopedic and dictionary: http://na.jkn21.com/
index to resource: http://resource.lib.u-tokyo.ac.jp/iri/url_search.cgi
http://webcat.nii.ac.jp/webcat_eng.html
ebook & materials: http://webcatplus.nii.ac.jp/
http:// opac.ndl.go.jp/
http:// u-tokyo.navi.little.jp/
http://ci.nii.ac.jp/en
UT Article: www.lib.u-tokyo.ac.jp/ext/utas/
web of Science: http:// isiknowledge.com/WOS
Science direct: www.sciencedirect.com/
Springer link: www.springerlink.com/
Wiley Interscience: www3.interscience.wiley.com/cgi-bin/home

Engineering village: www.engineeringvillage.org/
TU dissertation: http://gakui.dl.itc.u-tokyo.ac.jp/
doctoral dissertation: http:// dbr.nii.ac.jp/

e-journal: http:// www.lib.u-tokyo.ac.jp/ext/ejportal/
http:// ejournal.dl.itc.u-tokyo.ac.jp/

ebook: http:// www.netlibrary.org/
UT Repository: http://repository.dl.itc.u-tokyo.ac.jp/
*journal citation report: http://isiknowledge.com/JCR
impact factor
* Searching Ulrichsweb: http://www.ulrichsweb.com/
search journals, bibliographical info. on serials published

(DB) SW repository

- isfsg.org
* promisedata.org
- icu-project.org/repository/
* nasa (Nasa metric data program)
  Nasa IV & V facility
- google code (+time line, ...)
- $\boxed{\text{kdnuggets.com}}$ →

DB SW repository

promisedata SW rep.

isbsg.org

~~pro~~promisedata.org (85 datasets)

icu-project.org/~~re~~repository/

nasa metrics data program

nasa IV&V ~~facility~~ facility

google code

~~kdnu~~ kdnuggets.com

Pajek Jung Guess Prefuse .netmap RSNA

Sept 1991, Scientific American, (Communication, computers and Networks), Special issue

* virtual community

* (directed graph)

(newsgroup) usenet → Pajek

~~high~~ degree node (~~high number of connections to a node~~ a ~~large~~)

indegree (an author/node receives (~~a~~ ~~large~~ ~~high~~ number of ~~no~~ replies / connections) ~~receive~~)

~~out-degree (" " " sends " " " " ")~~

Degree of node (number of connections to the node)

in-degree (the number of replies/connection received by a node/author)

out-degree (the number of replies/connections sent by a node/author)

structural equivalent (same kind of structural ~~patterns~~ connections as others)

* distinguishing attributes (in context with newsgroup)
  - Answer person (* outward ties to local isolates)
    (* relative absence of triangles)
    (* Few intense ties)
  - Reply Magnet (* ties from local isolates often inward only)
    (* sparse, few triangles)
    (* Few intense ties)