Data Mining Steps

1. Data Cleaning	Removal of noise and inconsistent records
2. Data Integration	Combing multiple sources
3. Data Selection	Only data relevant for the task are retrieved from the database
4. Data Transformation	Converting data into a form more appropriate for mining
5. Data Mining	Application of intelligent methods to extract data patterns
6. Model Evaluation	Identification of truly interesting patterns representing knowledge
7. Knowledge Presentation	Visualization or other knowledge presentation techniques

Data mining could also be called Knowledge Discovery in Databases (see kdnuggets.com)

Types of Attributes

Nomial	e.g., ID numbers, eye color, zip codes
Ordinal	e.g., rankings, grades, height
Interval	e.g., calendar dates, temperatures
Ratio	e.g., length, time, counts

Distance Measures

Euclidean Distance:

$$dist = \sqrt{\sum_{k=1}^{n}(p_k - q_k)^2}$$

Minkowski Distance:

$$dist = \left(\sum_{k=1}^{n}|p_k - q_k|^r\right)^{\frac{1}{r}}$$

r=1, City Block
r=2, Euclidean
r–>inf., Chebyshev

Manhattan = City Block

Jaccard coefficient, Hamming, Cosine are a similarity / dissimilarity measures

Measures of Node Impurity

GAIN = measure before split – measure after split

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

$p(j \mid t)$ is the relative frequency of class j at node t

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

where, n_i = number of records at child i,
n = number of records at node p.
Pick the smallest

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Parent Node, p is split into k partitions;
n_i is number of records in partition i

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions;
n_i is number of records in partition i

$$Error(t) = 1 - \max_i P(i \mid t)$$

Model Evaluation

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+TN+FP}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F-measure} = \frac{2 \cdot TP}{2 \cdot TP+FN+FP}$$

$$\text{Cost} = TP \times Cost_{TP} + FN \times Cost_{FN} + TN \times Cost_{TN} + FP \times Cost_{FP}$$

$$\text{Sensitivity} = \text{Recall}$$

$$\text{Specificity} = 1 - \frac{FP}{FP+TN} = \frac{TN}{TN+FP}$$

$$\text{False Positive Rate} = 1 \text{ - Specificity}$$

Kappa = (observed agreement - chance agreement) / (1- chance agreement)

Kappa = (Dreal – Drandom) / (Dperfect – Drandom), where D indicates the sum of values in diagonal of the confusion matrix

K-Nearest Neighbor

```
* Compute distance between two points
* Determine the class from nearest neighbor list
      * Take the majority vote of class labels
           among the k-nearest neighbors
      * Weigh the vote according to distance
```

K-Nearest Neighbor (cont)

>	* weight factor, w = 1 / d^2

Rule-based Classification

Classify records by using a collection of "if…then…" rules

Rule: (Condition) --> y

where:

* Condition is a conjunction of attributes

* y is the class label

LHS: rule antecedent or condition

RHS: rule consequent

Examples of classification rules:

(Blood Type=Warm) ^ (Lay Eggs=Yes) --> Birds

(Taxable Income < 50K) ^ (Refund=Yes) --> Evade=No

Sequential covering is a rule-based classifier.

Rule Evaluation

$$\text{Accuracy} = \frac{n_c}{n}$$

$$\text{Laplace} = \frac{n_c + 1}{n + k}$$

$$\text{M-estimate} = \frac{n_c + kp}{n + k}$$

n : Number of instances covered by rule
n_c : Number of instances of class c covered by rule
k : Number of classes
p : Prior probability (for the positive class)

Bayesian Classification

Conditional Probability: $$P(C \mid A) = \frac{P(A, C)}{P(A)}$$

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

Baye's theorem:

$$P(C \mid A) = \frac{P(A \mid C) P(C)}{P(A)}$$

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{normalizing constant}}$$

Naïve Bayes Classifer:

$$\text{Original}: P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace}: P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$$

$$\text{m-estimate}: P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m}$$

c: number of classes, p: prior probability, m: parameter

P(B|A), read as *the probability of B given A*.

$$P(B|A) = \frac{P(A \text{ and } B)}{P(A)} \quad P(A \text{ and } B) = P(A) \cdot P(B|A)$$

p(a,b) is the probability that both a and b happen.

p(a|b) is the probability that a happens, knowing that b has already happened.

Terms

Association Analysis	Min-Apriori, LIFT, Simpson's Paradox, Anti-monotone property
Ensemble Methods	Staking, Random Forest

Terms (cont)

Decision Trees	C4.5, Pessimistic estimate, Occam's Razor, Hunt's Algorithm
Model Evaluation	Cross-validation, Bootstrap, Leave-one out (C-V), Misclassification error rate, Repeated holdout, Stratification
Bayes	Probabilistic classifier
Data Visualization	Chernoff faces, Data cube, Percentile plots, Parallel coordinates
Nonlinear Dimensionality Reduction	Principal components, ISOMAP, Multidimensional scaling

Ensemble Techniques

AdaBoost Algorithm:

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

error (ϵ_t) = # of misclassified divided by total

$$w_i = w_1 = w_2 = \cdots = w_{10} = \frac{1}{10} = 0.1$$

Re-weighting:

$$misclassified = w_i \times e^{+\alpha_t}$$

$$correct\ classified = w_i \times e^{-\alpha_t}$$

Manipulate training data: bagging and boosting ensemble of “experts”, each specializing on different portions of the instance space

Manipulate output values: error-correcting output coding (ensemble of “experts”, each predicting 1 bit of the {multibit} full class label)

Methods: BAGGing, Boosting, AdaBoost

Rules Analysis

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

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

$$Lift = \frac{Confidence}{P(B)}$$

Example:
Rule {b} –> {c}

	c	$\bar{c}$
b	3	4
$\bar{b}$	2	1

support = 3/10 = 0.3
confidence = 3/7 = 0.4286
lift = $\frac{3/7}{5/10}$

Apriori Algorithm

```
Let k=1
Generate frequent itemsets of length 1
Repeat until no new frequent itemsets are identified
        Gen erate length (k+1) candidate itemsets from
        length k frequent itemsets
        Prune candidate itemsets containing subsets
       of length k that are infrequent
        Count the support of each candidate by
        sca nning the DB
        Eli minate candidates that are infreq - uent,
        leaving only those that are frequent
```

K-means Clustering

```
Select K points as the initial centroids
repeat
       Form K Clusters by assigning all points to
the closest centroid
        Rec ompute the centroid of each cluster
until the centroids don't change
```

Closeness is measured by distance (e.g., Euclidean), similarity (e.g., Cosine), correlation.

Centroid is typically the mean of the points in the cluster

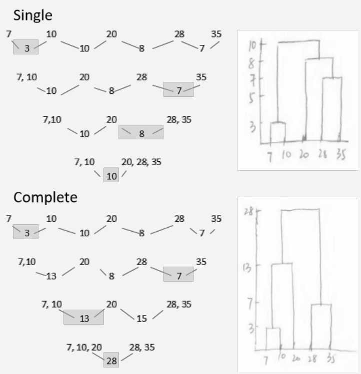
Hierarchical Clustering

Single-Link or MIN
Similarity of two clusters is based on the two most similar (closest / minimum) points in the different clusters
Determined by one pair of points, i.e., by one link in the proximity graph.
Complete or MAX
Similarity of two clusters is based on the two least similar (most distant, maximum) points in the different clusters
Determined by all pairs of points in the two clusters
Group Average
Proximity of two clusters is the average of pairwise proximity between points in the two clusters

Agglomerative clustering starts with points as individual clusters and merges closest clusters until only one cluster left.

Divisive clustering starts with one, all-inclusive cluster and splits a cluster until each cluster only has one point.

Dendrogram Example



Dataset: {7, 10, 20, 28, 35}

Density-Based Clustering

```
current_cluster_label <-- 1
for all core points do
    if the core point has no cluster label  then
                cur ren t_c lus ter _label <--
curren t_c lus ter _label +1
                Label the current core point with
the cluster label
    end if
    for all points in the Eps-ne igh bor hood,
except i-th the point itself do
        if the point does not have a cluster
label then
                        Label the point with
cluster label
        end if
    end for
```

Density-Based Clustering (cont)

> **end for**

DBSCAN is a popular algorithm

Density = number of points within a specified radius (Eps)

A point is a core point if it has more than a specified number of points (MinPts) within Eps

These are points that are at the interior of a cluster

A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point

A noise point is any point that is not a core point or a border point

Other Clustering Methods

Fuzzy is a partitional clustering method. **Fuzzy clustering** (also referred to as **soft clustering**) is a form of clustering in that each data point can belong to more than one cluster.
Graph-based methods: Jarvis-Patrick, Shared-Near Neighbor (SNN, Density), Chameleon
Model-based methods: Expectation-Maximization

Regression Analysis

* Linear Regression
  | Least squares
* Subset selection
* Stepwise selection
* Regularized regression
  | Ridge
  | Lasso

Regression Analysis (cont)

  | Elastic Net

Anomaly Detection

Anomaly is a pattern in the data that does not conform to the expected behavior (e.g., outliers, exceptions, peculiarities, surprise)
Types of Anomaly
  Point: An individual data instance is anomalous w.r.t. the data
  Contextual: An individual data instance is anomalous within a context
  Collective: A collection of related data instances is anomalous
Approaches
  * Graphical (e.g., boxplots, scatter plots)
  * Statistical (e.g., normal distribution, likelihood)
    | Parametric Techniques
    | Non-parametric Techniques
  * Distance (e.g., nearest-neighbor, density, clustering)

Local outlier factor (LOF) is a density-based distance approach

Mahalanobis Distance is a clustering-based distance approach