### 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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal</td>
<td>e.g., ID numbers, eye color, zip codes</td>
</tr>
<tr>
<td>Ordinal</td>
<td>e.g., rankings, grades, height</td>
</tr>
<tr>
<td>Interval</td>
<td>e.g., calendar dates, temperatures</td>
</tr>
<tr>
<td>Ratio</td>
<td>e.g., length, time, counts</td>
</tr>
</tbody>
</table>

### Distance Measures

**Euclidean Distance:**

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

**Minkowski Distance:**

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

- $r=1$, City Block
- $r=2$, Euclidean
- $r\to\infty$, Chebyshev

Manhattan = City Block

Jaccard coefficient, Hamming, Cosine are a similarity / dissimilarity measures
Measures of Node Impurity

**GAIN** = measure before split – measure after split

\[
GINI(t) = 1 - \sum_{j} \left( \frac{[p(j \mid t)]}{n} \right)^2
\]

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

\[
GINI_{\text{split}} = \sum_{i} \frac{n_i}{n} GINI(i)
\]

where, \( n_i \) = number of records at child \( i \), \( n \) = number of records at node \( p \).

Pick the smallest

**Entropy**

\[
Entropy(t) = -\sum_{j} p(j \mid t) \log p(j \mid t)
\]

Information Gain:

\[
GAIN_{\text{info}} = \text{Entropy}(t) - \left( \frac{1}{n} \sum_{i} \frac{n_i}{n} \text{Entropy}(i) \right)
\]

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

**GainRATIO** = \( \frac{GAIN_{\text{info}}}{\text{SplitINFO}} \)

**SplitINFO** = \( -\frac{1}{n} \sum_{i} \frac{n_i}{n} \log \frac{n_i}{n} \)

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

**Error**

\[
Error(t) = 1 - \max P(i \mid t)
\]

Model Evaluation

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class=Yes</td>
<td>Class=Yes</td>
</tr>
<tr>
<td>Class=No</td>
<td>Class=Yes</td>
</tr>
<tr>
<td>a (TP)</td>
<td>b (FN)</td>
</tr>
<tr>
<td>c (FP)</td>
<td>d (TN)</td>
</tr>
</tbody>
</table>

**Accuracy** = \( \frac{TP + TN}{TP + FN + FP + TN} \)

**Precision** = \( \frac{TP}{TP + FP} \)

**Recall** = \( \frac{TP}{TP + FN} \)

**F-measure** = \( \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \)

**Cost** = \( TP \times \text{Cost}_{TP} + FN \times \text{Cost}_{FN} + TN \times \text{Cost}_{TN} + FP \times \text{Cost}_{FP} \)

**Sensitivity** = \( \text{Recall} \)

**Specificity** = \( 1 - \frac{FP}{FP + TN} = \frac{TN}{TN + FP} \)

**False Positive Rate** = \( 1 - \text{Specificity} \)

Kappa = \( \frac{\text{observed agreement} - \text{chance agreement}}{(1 - \text{chance agreement})} \)

Kappa = \( \frac{D_{\text{real}} - D_{\text{random}}}{D_{\text{perfect}} - D_{\text{random}}} \), 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, \( w = \frac{1}{d^2} \)

Rule-based Classification

Classify records by using a collection of “if…then…” rules

**Rule:** (Condition) \( \rightarrow y \)

**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) \( ^\wedge \) (Lay Eggs=Yes) \( \rightarrow \) Birds

(Taxable Income < 50K) \( ^\wedge \) (Refund=Yes) \( \rightarrow \) Evade=No

Sequential covering is a rule-based classifier.

Bayesian Classification

**Conditional Probability:**

\[
P(C \mid A) = \frac{P(A \mid C) P(C)}{P(A)}
\]

Bayes' theorem:

\[
P(C \mid A) = \frac{P(A \mid C) P(C)}{P(A)}
\]

**Naive Bayes Classifier:**

Original: \( P(A_i \mid C) = \frac{N_{i,c}}{N_c} \)

Laplace: \( P(A_i \mid C) = \frac{N_{i,c} + 1}{N_c + E} \)

m-estimate: \( P(A_i \mid C) = \frac{N_{i,c} + \lambda p}{N_c + \lambda} \)

\( c \): number of classes, \( p \): prior probability, \( m \): parameter

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

\[
P(B \mid A) = \frac{P(A \text{ and } B)}{P(A)} = \frac{P(A \text{ and } B)}{P(A)}
\]

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

\( p(\text{lab}) \) is the probability that a happens, knowing that \( b \) has already happened.

**Terms**

**Association**

**Min-Apriori, LIFT, Simpson's Paradox, Anti-monotone property**

**Analysis**

**Ensemble Methods**

**Staking, Random Forest**

Published 30th April, 2017.
Last updated 30th April, 2017.
### 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 \( \epsilon_t \) = # of misclassified divided by total

- \( w_1 = w_1 = w_2 = \cdots = w_{10} = \frac{1}{10} = 0.1 \)

- 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

#### Apriori Algorithm

Let \( k=1 \)

1. Generate frequent itemsets of length 1
2. Repeat until no new frequent itemsets are identified
   - Generate 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 scanning the DB
   - Eliminate candidates that are infrequent, leaving only those that are frequent

#### Lift
\[
Lift = \frac{\text{Confidence}}{\text{P}(B)}
\]

Example:
- Rule \( (b) \rightarrow (c) \)

<table>
<thead>
<tr>
<th>( b )</th>
<th>( c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- Support = \( \frac{3}{10} = 0.3 \)
- Confidence = \( \frac{3}{7} = 0.4286 \)
- Lift = \( \frac{3/7}{3/10} \)

By HockeyPlay21

Published 30th April, 2017.
Last updated 30th April, 2017.

Sponsored by ApolloPad.com
Everyone has a novel in them. Finish Yours!
https://apollopad.com
**K-means Clustering**

Select K points as the initial centroids

**repeat**

Form K Clusters by assigning all points to the closest centroid

Recompute 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.

---

**Density-Based Clustering**

```python
current_cluster_label <-- 1
for all core points do
  if the core point has no cluster label then
    current_cluster_label <-- current_cluster_label + 1
    Label the current core point with the cluster label
  end if
  for all points in the Eps-neighborhood, except i-th the point itself do
    if the point does not have a cluster label
      Label the point with cluster label
    end if
  end for
end for
```

**Dataset:** [7, 10, 20, 28, 35]
Density-Based Clustering (cont)

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 (cont)

Regression Analysis

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

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)
  | Parametric Techniques
  | Non-parametric Techniques
* Statistical (e.g., normal distribution, likelihood)

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

Mahalanobis Distance is a clustering-based distance approach

Regression Analysis

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

By HockeyPlay21
cheatography.com/hockeyplay21/