Data Mining…

- Originally statistician’s term for overuse of data to draw invalid inferences.
- ESP?
  - 1/1000 people guessed color (red/black) of 10 out of 10 cards correctly!

Data Mining/Knowledge Discovery/
Intelligent Data Analysis

Task: discovering interesting patterns or structures in large amounts of data.
Multidisciplinary: machine learning, statistics and databases

Algorithms:
- construct particular types of models or patterns,
- rely on a built-in bias to simpler or more explanatory models,
- are designed to be efficient,
- require “interestingness” or evaluation criteria to separate pattern from noise.

Applications

- Marketing
  - Market basket associations between product sales, e.g., beer and diapers
  - Classify what types of customers buy what products
- Auto Insurance Fraud
  - Detect people who stage accidents to collect on insurance
- Money Laundering
  - Since 1993, the US Treasury’s Financial Crimes Enforcement Network agency has used a data-mining application to detect suspicious money transactions
- Spam filtering, flu or earthquake detection, ad tailoring, social trends, recommendations, uncovering secrets…

Key Issue: Feedback

Supervised: ground truth or correct answer is provided by a “teacher”
Reinforcement: program receives some evaluation, but is not told the correct answer
Unsupervised: correct answer must be inferred, no hints
Key Issue: Interestingness

- What makes a pattern interesting?
  - easily understood
  - valid on new data within some certainty
  - useful
  - novel

- Measures
  - statistical significance or frequency
  - recall vs. precision (how well are data characterized?)
  - model specific

Assessing Performance: Separating Training and Testing

The set of examples is divided into two sets: training and testing.

Varying the size of randomly selected training/testing sets produces a learning curve.

3rd set for validation: used to optimize parameters or select model prior to testing

Assessing Performance: Cross-Validation

addresses overfitting when model matches data too closely to permit generalizations.

K-fold Cross-validation (aka “Leave-One-Out”)
1. Shuffle items in the problem set.
2. Divide set into k equal sized parts (aka “folds”)
3. Do $i=1$ to $k$ times:
   1. Call the $i$th set the test set and put aside.
   2. Train the system on the other $k-1$ sets; test on the $i$th set and record performance
   3. Reset
4. Calculate average performance for the $k$ tests

Assessing Performance: Stratification

- Cross validation random folds assumes the data include all of the classes.
- But… for rare classes, can get unlucky.
- Stratification is when folds are selected so that each class is represented in both training and testing
Assessing Performance: Bootstrap

- Sampling with replacement
- Good for small datasets
- 0.632 bootstrap (approximate % of instances that make it into the training set)
  1. Collect \( n \) samples from dataset of size \( n \) with replacement which becomes the training set
  2. Instances not sampled become the testing set

Data Mining Models

- Classifiers
  - Supervised: training data includes known class
    - 1R, Decision Trees, Naïve Bayes
  - Unsupervised: training data do not have correct answer
    - Clusters
- Patterns in Logs:
  - Association rules
  - Sequence detection

Classification: Generalizing from Examples

Given a collection of examples (input and classification pairs), return a function that approximates the true classification.

Typically, data are vectors of feature values (e.g., Vector model of terms in Information Retrieval).

Alternate Classification Models

Bias is any preference for one hypothesis over another beyond mere consistency of examples.

Hard part is determining what makes a good classification representation.

Ockham’s razor: the most likely hypothesis is the simplest one that is consistent with the observations.
Supervised I: Nearest-Neighbors

- Select a classification for a new point based on the labels for the \( k \) nearest neighbors
  - For discrete, take majority vote.
  - For continuous, average, do linear or weighted regression.
- Issues:
  - How to pick \( k \)?
  - How to compute distances?

Supervised II: 1R

- Rule on one attribute
- Most frequent class for each value of the attribute
- Choose attribute with the lowest error rate:
  - % of instances that don’t belong to most frequent class for the value

Example Data

<table>
<thead>
<tr>
<th>#</th>
<th>Hair</th>
<th>Height</th>
<th>Weight</th>
<th>Lotion</th>
<th>Sunburn?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>blonde</td>
<td>average</td>
<td>light</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>blonde</td>
<td>tall</td>
<td>average</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>3</td>
<td>brown</td>
<td>short</td>
<td>average</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>blonde</td>
<td>short</td>
<td>average</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>red</td>
<td>average</td>
<td>heavy</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>brown</td>
<td>tall</td>
<td>heavy</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>brown</td>
<td>average</td>
<td>heavy</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>8</td>
<td>blonde</td>
<td>short</td>
<td>light</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
### 1R on example data

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Rules</th>
<th>Errors</th>
<th>Total Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hair</td>
<td>Blonde → yes</td>
<td>2/4</td>
<td>3/8</td>
</tr>
<tr>
<td></td>
<td>Brown → no</td>
<td>1/3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Red → yes</td>
<td>0/1</td>
<td></td>
</tr>
<tr>
<td>Height</td>
<td>Short → no</td>
<td>1/3</td>
<td>1/8</td>
</tr>
<tr>
<td></td>
<td>Average → yes</td>
<td>0/3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tall → no</td>
<td>0/2</td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>Light → yes</td>
<td>1/3</td>
<td>3/8</td>
</tr>
<tr>
<td></td>
<td>Average → no</td>
<td>1/3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Heavy → yes</td>
<td>1/3</td>
<td></td>
</tr>
<tr>
<td>Lotion</td>
<td>Yes → no</td>
<td>0/3</td>
<td>1/8</td>
</tr>
<tr>
<td></td>
<td>No → yes</td>
<td>0/5</td>
<td></td>
</tr>
</tbody>
</table>

### 1R Extension Hyperpipes

- **Procedure:**
  - Construct one rule per class: conjunction of tests for subset of values on each attribute
  - Prediction is the class returned by the most rules
  - Simple form of voting/ensemble

### Supervised III: Decision Trees

- Models a discrete valued function
- Represent concepts as a series of decisions (answers to tests)
- Internal nodes are tests with branches to different answers
- Leaf nodes are the classification for the data element

### Decision Trees, basic idea...

- if $X > 5$ then yellow
- else if $Y > 3$ then yellow
- else if $X > 2$ then blue
- else yellow

---

Figure/example taken from [www.kdnuggets.com/data_mining_course](http://www.kdnuggets.com/data_mining_course) by Piatetsky & Parker
Using Information Theory to Choose What to Ask When

Can we characterize how much it is worth to get an answer to a question?
- Given a probability distribution, the information required to predict an event is the distribution’s entropy
- One bit of information is enough to answer a yes or no question

General case: Entropy is Information
\[ I(P(v_1, ..., v_n)) = \sum -P(v_i) \log_2 P(v_i) \]

Boolean case:
\[ I(P(t), P(f)) = -P(t) \log_2 P(t) - P(f) \log_2 P(f) \]

Relate this to Coin Tossing

Consider information in tossing a fair coin.
\[ I(\frac{1}{2}, \frac{1}{2}) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1 \text{ bit} \]

Consider an unfair coin, heads 1 in 100 tosses.
\[ I(\frac{1}{100}, \frac{99}{100}) = -\frac{1}{100} \log_2 \frac{1}{100} - \frac{99}{100} \log_2 \frac{99}{100} \approx .08 \text{ bit} \]

Efficiently Splitting the Cases

Assume world is divided into two classes: P (positive) and N (negative). Assumptions for test are:
1. Any correct decision tree will classify objects in the same proportion as their representation in the example set. An object will belong to P with probability \( p/p+n \) and N with probability \( n/n+p \).
2. A decision tree returns P or N with expected information needed given by:
\[ I(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n} \]

Information content of knowing whether \( x \in P \) or \( x \in N \).
### Supervised IV: Statistical Modeling

- Assumptions about attributes
  - Equally important
  - Statistically independent
    - Knowing the value of one attribute says nothing about the value of another if the class is known
- In practice, the assumptions do not hold, but can still derive fairly accurate models!
- Naïve Bayes (covered briefly in CS440 and more in CS545)

### Unsupervised I: Cluster or Outlier Analysis

- Group together similar data, usually when class labels are unknown
- Detect anomalous data
  - Example: connection features that do not match normal access
  - Or classify types (assume members share some characteristics)
    - Example: connection features of a smurf network attack

### Clustering... in General

- A cluster is a cone in n-dimensional space; members are found within $\varepsilon$ of an “average” vector.
- Different techniques for determining the cluster vector and $\varepsilon$.
  - **Hard** clustering techniques assign a class label to each cluster; members of clusters are mutually exclusive.
  - **Fuzzy** clustering techniques assign a fractional degree of membership to each label for each $x$.

### Clustering Decisions

- **Pattern Representation**
  - Feature selection (which to use, what type of values)
  - Number of categories/clusters
- **Pattern proximity**
  - Distance measure on pairs of patterns
- **Grouping**
  - Characteristics of clusters (e.g., fuzzy, hierarchical)

Clustering algorithms embody different assumptions about these decisions and the form of clusters.
Proximity Measures

- Generally, use Euclidean distance or mean squared distance.
- May need to normalize or use metrics less sensitive to magnitude differences (e.g., Manhattan distance aka L1 norm or Mahalanobis distance).
- May have domain specific distance measure, e.g., cosine between vectors for information retrieval.

Categories of Clustering Techniques

- Partitional:
  - Incrementally construct partitions based on some criterion (e.g., distance from data to cluster vector).
- Hierarchical:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion.
- Model-based:
  - A model is hypothesized for each of the clusters.
  - Find the best fit of that model to each other.
  - E.g., Bayesian classification (AutoClass), Cobweb.

Partitional Algorithms

- Results in set of unrelated clusters.
- Issues:
  - how many clusters is enough?
    - usually set by user
  - how to search space of possible partitions?
    - incremental, batch, exhaustive...
  - what is appropriate clustering criterion?
    - global optimal
    - heuristic: distance to centroid (k-means), distance to some median point (k-mediods)

k-Means Clustering Algorithm

1. Randomly select k samples as cluster centroids.
2. Assign each pattern to the closest cluster centroid.
3. Recompute centroids.
4. If convergence criterion (e.g., minimal decrease in error or no change in cluster composition) is not met, return to 2.
Example: K-Means Solutions

Evaluation Metrics

- How to measure cluster quality?
  - Members of each cluster should share some relevant properties.
    - high intra-class similarity
    - low inter-class similarity
  - Another metric for cluster quality: Entropy
    \[ E_j = -\sum p_{ij} \log(p_{ij}) \]
    \[ E_{cs} = \frac{\sum_j p_j E_j}{n} \]
    where \( p_{ij} \) is probability that a member of cluster \( j \) belongs to class \( i \), \( n_j \) is size of cluster \( j \), \( m \) is number of clusters, \( n \) is number of instances and \( CS \) is a clustering solution.

Associations

- frequently co-occurring features
- example “market basket” problem:
  - Amazon: customers who bought Tom Clancy novels also bought Robert Ludlum
  - apocryphal: people who buy diapers are also likely to buy beer (but not the converse)

What makes good association rules?

- Interestingness metrics
  - support
  - confidence
  - attributes of interest
  - rules with a specific structure
- Comparing rule sets
  - completeness: generate all interesting rules
  - efficiency: generate only rules that are interesting
Terminology

- Each DB transaction is a set of items (called an itemset).
- A $k$-itemset is an itemset containing sets of cardinality $k$.
- An association rule is $A \Rightarrow B$, such that $A$ and $B$ are sets of items with empty set as their intersection and including items that co-occur in transactions.

Terminology (cont.)

- A rule has support $s$, where $s$ is % of transactions that contain both $A$ and $B$: $P(A \cap B)$.
- A rule has confidence $c$, where $c$ is % of transactions containing $A$ that also contain $B$: $P(B|A)$.
- Rules with support and confidence exceeding a threshold are called strong.
- A frequent itemset is the set of itemsets whose minimum support exceeds a threshold.

Association Rule Mining

Process

1) Find all frequent itemsets based on minimum support
2) Identify strong association rules from the frequent itemsets

Rule Representations

- types of values: Boolean, quantitative (binned)
- # of dimensions/predicates
- levels of abstraction/type hierarchy

Apriori Algorithm for Finding Single-Dimensional Boolean Itemsets

- Apriori property: all nonempty subsets of a frequent itemset must also be frequent.
- Generate $k$-itemsets by joining large $k-1$-itemsets and deleting any that is not large.
- Assume items in lexicographic order
- Notation:

<table>
<thead>
<tr>
<th>$k$-itemset</th>
<th>itemset of size $k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_k$</td>
<td>$k$-itemsets with minsup</td>
</tr>
<tr>
<td>$C_k$</td>
<td>candidate $k$-itemsets</td>
</tr>
</tbody>
</table>
Apriori Algorithm

$L_1 = \{\text{itemsets} \text{ of large 1-itemsets}\}$

for $k = 2; L_{k-1} \neq \{\}; k++$ do begin

$C_k = \text{apriori-gen}(L_{k-1})$; new candidates

transactions $t \in D$ do begin

$C'_t = \text{subset}(C_t, t)$; candidates in transaction

candidates $c \in C'_t$ do

$c.\text{count} + +$; determine support

end

$L_k = \{c \in C_t | c.\text{count} \geq \text{minsup}\}$ create new set

end

Answer $\bigcup_k L_k$;

Apriori-gen: join

Apriori-gen: prune

Apriori Algorithm

$L_1 = \{\text{itemsets of large 1-itemsets}\}$

for $k = 2; L_{k-1} \neq \{\}; k++$ do begin

$C_k = \text{apriori-gen}(L_{k-1})$; new candidates

transactions $t \in D$ do begin

$C'_t = \text{subset}(C_t, t)$; candidates in transactions

candidates $c \in C'_t$ do

$c.\text{count} + +$; determine support

end

$L_k = \{c \in C_t | c.\text{count} \geq \text{minsup}\}$ create new set

end

Answer $\bigcup_k L_k$;
Apriori: Subset

Find all transactions associated with candidate itemsets.

- Candidate itemsets are stored in hash-tree
  - Leaves contain a list of itemsets.
  - Interior nodes contain a hash table
  - Depth corresponds to position of item of itemset
- Find all candidates contained in transaction \( t \):
  - If at leaf, check itemsets against \( t \)
  - If interior for \( i \)th item, hash on items after \( i \) in \( t \) and recurse

Apriori Example

\( D = \{(c \ e \ f), (b \ c \ e \ f), (b \ d), (b \ f \ g), (a \ c \ e), (a \ b), (a \ c \ g), (c \ e \ f), (a \ d \ g), (c \ f), (a \ b \ c \ e \ f), (a \ f \ g), (c \ e \ f), (a \ b \ c \ g), (a), (a \ g), (c \ e)\} \)

\( \text{minsup} = 4 \)

\( L_1 = \{\{a\} 10, \{b\} 6, \{c\} 11, \{e\} 7, \{f\} 8, \{g\} 7\} \) no \( d \)

\( C_2 = \{\{a, b\}, \{a, c\}, \{a, e\}, \{a, f\}, \{a, g\}, \{b, c\}, \{b, e\}, \{b, f\}, \{b, g\}, \{c, e\}, \{c, f\}, \{c, g\}, \{e, f\}, \{e, g\}, \{f, g\}\} \) after join&prune

\( L_2 = \{\{a \ b \ 4\}, \{a \ c \ 5\}, \{a \ g \ 5\}, \{b \ c \ 5\}, \{c \ e \ 7\}, \{c \ f \ 6\}, \{e \ f \ 5\}\} \) after support

\( C_3 = \{\{a \ b \ c\}\} \) after join&prune

\( L_3 = \{\{c \ e \ f \ 5\}\} \) after support

Generating Association Rules from Supported Itemsets

- Rules take form \( A \Rightarrow B \) where \( A \) & \( B \) are subsets of items that co-occur.
- Rules have support above \( \text{minsup} \) threshold and confidence above \( \text{minconf} \) threshold.

\[
\text{confidence}(A \Rightarrow B) = \frac{\text{support}_\text{count}(A \cap B)}{\text{support}_\text{count}(A)}
\]

\[
\text{support}_\text{count}(A \cap B) \text{= transactions containing itemsets } A \cap B \text{ \|}
\]

\[
\text{support}_\text{count}(A) \text{= transactions containing itemsets } A \text{ \|}
\]

Apriori Issues

- Generating candidate itemsets:
  - \( n \) frequent 1-itemsets generates \( (n(n-1))/2 \) candidate 2-itemsets
  - generate itemsets for each length
- Multipass over data for frequencies
Generating Association Rules from Supported Itemsets (cont.)

- Generate all rules such that:
  - for each frequent itemset \( i \) of cardinality > 1,
    - generate all non-empty subsets.
  - for every non-empty subset \( s \),
    - if \( \text{confidence}(s \Rightarrow (i-s)) > \text{minconf} \), create rule \( s \Rightarrow (i-s) \).
- Can use cached itemset frequencies for confidence calculation.

Association Rule Example (cont.)

Frequent itemsets: \{c e f 5\}, \{a b 4\}, \{a c 5\}, \{a g 5\}, \{b c 4\}, \{c e 7\}, \{c f 6\}, \{e f 5\}
1. Non-empty subsets of \{c e f\}: \{c e\}, \{c f\}, \{e f\}, \{c\}, \{e\}, \{f\}
2. \text{confidence}(\{c e\} \Rightarrow \{f\}) = 5/7 = .71
3. \text{confidence}(\{c f\} \Rightarrow \{e\}) = 5/6 = .83
4. \text{confidence}(\{e f\} \Rightarrow \{c\}) = 5/5 = 1.0
5. \text{confidence}(\{c\} \Rightarrow \{e f\}) = 5/11 = .45
6. \text{confidence}(\{e\} \Rightarrow \{c f\}) = 5/7 = .71
7. \text{confidence}(\{f\} \Rightarrow \{c e\}) = 5/8 = .62

Interestingness Revisited

- Correlation/Lift \( \frac{P(A \land B)}{P(A)P(B)} \)
  - \( P(A \land B) = P(B)P(A) \), if \( A \) and \( B \) are independent events
  - \( A \) and \( B \) negatively correlated, if the value is less than 1;
    otherwise \( A \) and \( B \) positively correlated

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Support</th>
<th>Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>XY</td>
<td>25%</td>
<td>2</td>
</tr>
<tr>
<td>XZ</td>
<td>37.56%</td>
<td>0.8</td>
</tr>
<tr>
<td>YZ</td>
<td>12.56%</td>
<td>0.87</td>
</tr>
</tbody>
</table>

revised from Han&Kamber notes at http://www-faculty.cs.uiuc.edu/~hanj/DM_Book.html

Other Methods for Association Mining

- Improving Apriori efficiency:
  - Partitioning
  - Sampling
- Multi-level or generalized association
- Multi-Dimensional association
- Quantitative association rule mining
- Constraint-based or query-based association
- Correlation analysis