ACSys Data Mining

- CRC for Advanced Computational Systems
  - ANU, CSIRO, (Digital), Fujitsu, Sun, SGI
  - Five programs: one is Data Mining
  - Aim to work with collaborators to solve real problems and feed research problems to the scientists
  - Brings together expertise in Machine Learning, Statistics, Numerical Algorithms, Databases, Virtual Environments
About Us

- Graham Williams, Senior Research Scientist with CSIRO Machine Learning
- Stephen Roberts, Fellow with Computer Sciences Lab, ANU Numerical Methods
- Markus Hegland, Fellow with Computer Sciences Lab, ANU Numerical Methods
Outline

• Data Mining Overview
  – History
  – Motivation

• Techniques for Data Mining
  – Link Analysis: Association Rules
  – Predictive Modeling: Classification
  – Predictive Modeling: Regression
  – Data Base Segmentation: Clustering
So What is Data Mining?

- The non-trivial extraction of novel, implicit, and actionable knowledge from large datasets.
  - Extremely large datasets
  - Discovery of the non-obvious
  - Useful knowledge that can improve processes
  - Can not be done manually

- Technology to enable data exploration, data analysis, and data visualisation of very large databases at a high level of abstraction, without a specific hypothesis in mind.
And Where Has it Come From?

Data Mining

- Machine Learning
- High Performance Computers
- Parallel Algorithms
- Database
- Visualisation
- Applied Statistics
- Pattern Recognition
Knowledge Discovery in Databases

- A six or more step process:
  - data warehousing,
  - data selection,
  - data preprocessing,
  - data transformation,
  - data mining,
  - interpretation/evaluation

- Data Mining is sometimes referred to as KDD

- DM and KDD tend to be used as synonyms
The KDD Treadmill
Techniques Used in Data Mining

- **Link Analysis**
  association rules, sequential patterns, time sequences

- **Predictive Modelling**
  tree induction, neural nets, regression

- **Database Segmentation**
  clustering, k-means,

- **Deviation Detection**
  visualisation, statistics
Typical Applications of Data Mining

Source: IDC 1998
Typical Applications of Data Mining

- **Sales/Marketing**
  - Provide better customer service
  - Improve cross-selling opportunities (beer and nappies)
  - Increase direct mail response rates

- **Customer Retention**
  - Identify patterns of defection
  - Predict likely defections

- **Risk Assessment and Fraud**
  - Identify inappropriate or unusual behaviour
ACSys

ACSys Data Mining

Mt Stromlo Observatory

NRMA Insurance Limited

Australian Taxation Office

Health Insurance Commission
Some Research

- Interestingness through Evolutionary Computation
- Virtual Environments
- Data Mining Standards
- Temporal Data Mining
- Spatial Data Mining
- Feature Selection
Outline

- Data Mining Overview
  - History
  - Motivation

- Techniques for Data Mining
  - Link Analysis: Association Rules
  - Predictive Modeling: Classification
  - Predictive Modeling: Regression
  - Data Base Segmentation: Clustering
Why Data Mining Now?

- Changes in the Business Environment
  - Customers becoming more demanding
  - Markets are saturated

- Drivers
  - Focus on the customer, competition, and data assets

- Enablers
  - Increased data hoarding
  - Cheaper and faster hardware
The Growth in KDD

- Research Community
  - KDD Conference annually since 1995
  - KDD Journal since 1997
  - ACM SIGKDD http://www.acm.org/sigkdd

- Commercially
  - Research: IBM, Amex, NAB, AT&T, HIC, NRMA
  - Services: ACSys, IBM, MIP, NCR, Magnify
  - Tools: TMC, IBM, ISL, SGI, SAS, Magnify
Outline

• Data Mining Overview
  – History
  – Motivation

• Techniques for Data Mining
  – Link Analysis: Association Rules
  – Predictive Modeling: Classification
  – Predictive Modeling: Regression
  – Data Base Segmentation: Clustering
The Scientist’s Motivation

- **The Real World**
  - Offers many challenging problems
  - Enormous databases now exist and readily available

- Statistics building models and doing analysis for years?
  - Statistics limited computationally
  - Relevance of statistics if we do not sample
  - There are not enough statisticians to go around!

- Machine Learning to build models?
  - Limited computationally, useful on toy problems, but ...
Motivation: The Sizes

- Databases today are huge:
  - More than 1,000,000 entities/records/rows
  - From 10 to 10,000 fields/attributes/variables
  - Giga-bytes and tera-bytes

- Databases a growing at an unprecedented rate

- The corporate world is a cut-throat world
  - Decisions must be made rapidly
  - Decisions must be made with maximum knowledge
Motivation for doing Data Mining

- Investment in Data Collection/Data Warehouse
  - Add value to the data holding
  - Competitive advantage
  - More effective decision making

- **OLTP** $\Rightarrow$ **Data Warehouse** $\Rightarrow$ **Decision Support**
  - Work to add value to the data holding
  - Support high level and long term decision making
  - Fundamental move in use of Databases
Another Angle: The Personal Data Miner

- The Microsoft Challenge
- Information overload
- Internet navigation
- Intelligent Internet catalogues
Outline

• Data Mining Overview
  – History
  – Motivation

• Techniques for Data Mining
  – Link Analysis: Association Rules
  – Predictive Modeling: Classification
  – Predictive Modeling: Regression
  – Data Base Segmentation: Clustering
ACSys

Data Mining Operations

- **Link Analysis**
  links between individuals rather than characterising whole

- **Predictive Modelling** (supervised learning)
  use observations to learn to predict

- **Database Segmentation** (unsupervised learning)
  partition data into similar groups
Outline

• Data Mining Overview
  – History
  – Motivation

• Techniques for Data Mining
  – Link Analysis: Association Rules
  – Predictive Modeling: Classification
  – Predictive Modeling: Regression
  – Data Base Segmentation: Clustering
Link Analysis: Association Rules

- A technique developed specifically for data mining
  - Given
    * A dataset of customer transactions
    * A transaction is a collection of items
  - Find
    * Correlations between items as rules

- Examples
  - Supermarket baskets
  - Attached mailing in direct marketing
Determining Interesting Association Rules

- Rules have confidence and support
  - IF x and y THEN z with confidence c
    * if x and y are in the basket, then so is z in c% of cases
  - IF x and y THEN z with support s
    * the rule holds in s% of all transactions
Example

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345</td>
<td>A B C</td>
</tr>
<tr>
<td>12346</td>
<td>A C</td>
</tr>
<tr>
<td>12347</td>
<td>A D</td>
</tr>
<tr>
<td>12348</td>
<td>B E F</td>
</tr>
</tbody>
</table>

- Input Parameters: confidence = 50%; support = 50%
- if A then C: c = 66.6% s = 50%
- if C then A: c = 100% s = 50%
Typical Application

- Hundreds of thousands of different items
- Millions of transactions
- Many gigabytes of data
- It is a large task, but linear algorithms exist
Itemsets are Basis of Algorithm

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Items</th>
<th>Itemset</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345</td>
<td>A B C</td>
<td>A</td>
<td>75%</td>
</tr>
<tr>
<td>12346</td>
<td>A C</td>
<td>B</td>
<td>50%</td>
</tr>
<tr>
<td>12347</td>
<td>A D</td>
<td>C</td>
<td>50%</td>
</tr>
<tr>
<td>12348</td>
<td>B E F</td>
<td>A, C</td>
<td>50%</td>
</tr>
</tbody>
</table>

- Rule $A \Rightarrow C'$
- $s = s(A, C') = 50$
- $c = s(A, C')/s(A) = 66.6\%$
Algorithm Outline

- Find all large itemsets
  - sets of items with at least minimum support
  - Apriori and AprioriTid and newer algorithms

- Generate rules from large itemsets
  - For ABCD and AB in large itemset the rule $AB \Rightarrow CD$ holds if ratio $s(ABCD)/s(AB)$ is large enough
  - This ratio is the confidence of the rule
HIC Example

- Associations on episode database for pathology services
  - 6.8 million records X 120 attributes (3.5GB)
  - 15 months preprocessing then 2 weeks data mining
- Goal: find associations between tests
  - $c_{\text{min}} = 50\%$ and $s_{\text{min}} = 1\%, \ 0.5\%, \ 0.25\%$
    (1% of 6.8 million = 68,000)
  - Unexpected/unnecessary combination of services
  - Refuse cover saves $550,000 per year
Psuedo Algorithm

(1) \( F_1 = \{ \text{frequent 1-item-sets} \} \)
(2) \textbf{for} \ ((k = 2; F_{k-1} \neq \emptyset; k++) \textbf{ do begin}
(3) \( C_k = \text{apriori_gen}(F_{k-1}) \)
(4) \textbf{ for all transactions} \( t \in T \)
(5) \( \quad \text{subset}(C_k, t) \)
(6) \( F_k = \{ C \in C_k | \text{c.count} \geq \text{minsup} \} \)
(7) \textbf{ end}
(8) \text{Answer} = \bigcup F_k
Parallel Algorithm: Count Distribution Algorithm

- Each processor works (and stores) complete set of Candidates and produces local support counts for local transactions

- Global support counts obtained via a global reduction operation

- Good scalability when working with small numbers of candidates (large support), but unable to deal with large number of candidates (small support).

[Agrawal & Shafer 96]
Parallel Algorithm: Data Distribution Algorithm

- Each processor computes support counts for only $|C_k|/P$ candidates. Need to move transaction data between processors via all to all communication.

- Able to deal with large numbers of candidates, but speedups not as good as Count Distribution Algorithm for large transaction data size.

[Agrawal & Shafer 96]
Improved Parallel Algorithm: Intelligent Data Distribution

- Uses more efficient inter-processor communication scheme: point to point

- Switches to Count Distribution when the total number of candidate itemsets fall below a given threshold

- The candidate itemsets are distributed among the processors so that each processor gets itemsets that begin only with a subset of all possible items

[Han, Karypis & Kumar 97]
Improved Parallel Algorithm: Hybrid Algorithm

- Combines Count Distribution Algorithm and the Intelligent Data Distribution Algorithm
- Data divided evenly between processors
- Processors divided into groups
- In each group Intelligent Data Distribution Algorithm is run
- Each group supplies local support counts, ala the Count Distribution Algorithm

[Han, Karypis & Kumar 97]
Outline

- Data Mining Overview
  - History
  - Motivation

- Techniques for Data Mining
  - Link Analysis: Association Rules
  - Predictive Modeling: Classification
  - Predictive Modeling: Regression
  - Data Base Segmentation: Clustering
Predictive Modelling: Classification

- Goal of classification is to build structures from examples of past decisions that can be used to make decisions for unseen cases.

- Often referred to as supervised learning.

- Decision Tree and Rule induction are popular techniques

- Neural Networks also used
Classification: C5.0

- Quinlan: \(\text{ID3} \Rightarrow \text{C4.5} \Rightarrow \text{C5.0}\)
- Most widely used Machine Learning and Data Mining tool
  Started as Decision Tree Induction, now Rule Induction, also
- Available from http://www.rulequest.com/
- Many publically available alternatives
- CART developed by Breiman et al. (Stanford)
  Salford Systems http://www.salford-systems.com
Decision Tree Induction

- Decision tree induction is an example of a recursive partitioning algorithm

- Basic motivation:
  - A dataset contains a certain amount of information
  - A random dataset has high entropy
  - Work towards reducing the amount of entropy in the data
  - Alternatively, increase the amount of information exhibited by the data
ACSys

Algorithm

How to partition?

Which partition?

When to stop?

Discriminating Descriptions

A

A_1

A_2

B

B_1

B_2

D=no

D=yes

D=yes

B=B_1

B=B_2

Selection Criterion
Algorithm

- Construct set of candidate partitions $S$
- Select best $S^*$ in $S$
- Describe each cell $C_i$ in $S^*$
- Test termination condition on each $C_i$
  - true: form a leaf node
  - false: recurse with $C_i$ as new training set
Discriminating Descriptions

• Typical algorithm considers a single attribute at one time:

  • **categorical attributes**
    – define a disjoint cell for each possible value: sex = “male”
    – can be grouped: transport ∈ (car, bike)

  • **continuous attributes**
    – define many possible binary partitions
    – Split $A < 24$ and $A \geq 24$
    – Or split $A < 28$ and $A \geq 28$
Information Measure

- Estimate the gain in information from a particular partitioning of the dataset

- A decision tree produces a message which is the decision

- The information content is $\sum_{j=1}^{m} -p_j \log(p_j)$
  - $p_j$ is the probability of making a particular decision
  - there are $m$ possible decisions

- Same as entropy: $\sum_{j=1}^{m} p_j \log(1/p_j)$. 
ACSys

Information Measure

- $\text{info}(T) = \sum_{j=1}^{m} -p_j \log(p_j)$ is the amount of information needed to identify class of an object in $T$

- Maximised when all $p_j$ are equal

- Minimised (0) when all but one $p_j$ is 0 (the remaining $p_j$ is 1)

- Now partition the data into $n$ cells

- Expected information requirement is then the weighted sum:
  
  $$\text{info}_x(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \text{info}(T_i)$$
The information that is gained by partitioning $T$ is then:

$$gain(A) = info(T) - info_x(T)$$

This *gain criterion* can then be used to select the partition which maximises information gain.

Variations of the Information Gain have been developed to avoid various biases: Gini Index of Diversity.
End Result

```
A
  b c
  E Y
 Y >=63
N <63
```
Types of Parallelism

- Inter-node Parallelism: multiple nodes processed at the same time

- Inter-Attribute-Evaluation Parallelism: where candidate attributes in a node are distributed among processors

- Intra-Attribute-Evaluation Parallelism: where the calculation for a single attribute is distributed between processors
Example: ScalParC

- **Data Structures**
  - Attribute Lists: separate lists of all attributes, distributed across processors
  - Node Table: Stores node information for each record id
  - Count Matrices: stored for each attribute, for all nodes at a given level

[Joshi, Karypis & Kumar 97]
Outline of ScalParC Algorithm

(1) Sort Continuous Attributes
(2) do while (there are nodes requiring splitting at current level)
(3) Compute count matrices
(4) Compute best index for nodes requiring split
(5) Partition splitting attributes and update node table
(6) Partition non-splitting attributes
(7) end do
Pruning

- We may be able to build a decision tree which perfectly reflects the data.

- But the tree may not be generally applicable called **overfitting**.

- Pruning is a technique for simplifying and hence generalising a decision tree.
Error-Based Pruning

- Replace sub-trees with leaves
- Decision class is the majority
- Pruning based on predicted error rates
  - prune subtrees which result in lower predicted error rate
Pruning

- How to estimate error? Use a separate test set:
  - Error rate on training set (resubstitution error) not useful because pruning will always increase error
  - Two common techniques are cost-complexity pruning and reduced-error pruning

- Cost Complexity Pruning: Predicted error rate modelled as weighted sum of complexity and error on training set—the test cases used to determine weighting

- Reduced Error Pruning: Use test set to assess error rate directly
Issues

- Unknown attribute values
- Run out of attributes to split on
- Brittness of method—small perturbations of data lead to significant changes in decision trees
- Trees become too large and are no longer particularly understandable (thousands of nodes)

**Data Mining**: Accuracy, alone, is not so important
ACSys

Classification Rules

- A tree can be converted to a rule set by traversing each path

```
A
\( b \) \( A \) \( c \)
\( E \) \( \geq 63 \)
\( Y \) \( N \)
\(< 63 \)
```

- \( A = c \Rightarrow Y \)
- \( A = b \wedge E < 63 \Rightarrow Y \)
- \( A = b \wedge E \geq 63 \Rightarrow N \)
- Rule Pruning: Perhaps \( E \geq 63 \Rightarrow N \)
Pros and Cons of Decision Tree Induction

• Pros
  – Greedy Search = Fast Execution
  – High dimensionality not a problem
  – Selects important variables
  – Creates symbolic descriptions

• Cons
  – Search space is huge
  – Interaction terms not considered
  – Parallel axis tests only \((A = v)\)
Recent Research

- **Bagging**
  - Sample with resubstitution from training set
  - Build multiple decision trees from different samples
  - Use a voting method to classify new objects

- **Boosting**
  - Build multiple trees from all training data
  - Maintain a weight for each instance in the training set that reflects its importance
  - Use a voting method to classify new objects