Data Mining: Concepts and Techniques

— Additional Applications and Emerging Topics —

Li Xiong

Slides credits: Jiawei Han and Micheline Kamber
Chris Clifton
Agrawal and Srikant
Outline

- Biological data mining
- Data mining for intrusion detection
- Privacy-preserving data mining
Biological Data Mining

- High throughput biological data
  - DNA or protein sequence data (nucleotides or amino acids).
  - 3D Protein structure data and protein-protein interaction data
  - Microarray or gene expression data
  - Flow cytometry data

- Mining biological data
  - Alignment and comparative analysis of DNA or protein sequences
  - Discover structural patterns of genetic networks and protein pathways
  - Association analysis and clustering of co-occurring/similar gene sequences
  - Classification based on gene expression patterns
Sequence Alignment

- Goal: given two or more input sequences, identify similar sequences with long conserved subsequences
  
  HEAGAWGHEE       PAWHEAE

- Substitution: probabilities of substitutions, insertions and deletions

- Scoring based on substitution

- Problem: find best alignment with maximal score
  
  *Optimal* alignment problem: NP-hard
  
  Heuristic method to find *good* alignments
## Pair-wise Sequence Alignment: Scoring Matrix

### HEAGAWGHEE  PAWHEAE

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>E</th>
<th>G</th>
<th>H</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>-1</td>
<td>0</td>
<td>-2</td>
<td>-3</td>
</tr>
<tr>
<td>E</td>
<td>-1</td>
<td>6</td>
<td>-3</td>
<td>0</td>
<td>-3</td>
</tr>
<tr>
<td>H</td>
<td>-2</td>
<td>0</td>
<td>-2</td>
<td>10</td>
<td>-3</td>
</tr>
<tr>
<td>P</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-4</td>
</tr>
<tr>
<td>W</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>15</td>
</tr>
</tbody>
</table>

- **Gap penalty:** -8
- **Gap extension:** -8

**HEAGAWGHE--E**

P-A--W-HEAE

\[ (-8) + (-8) + (-1) + (-8) + 5 + 15 + (-8) + 10 + 6 + (-8) + 6 = 1 \]

**HEAGAWGHE--E**

P-A--W-HEAE
Heuristic Alignment Algorithms

- Motivation: Complexity of alignment algorithms: $O(nm)$
  - Current protein DB: 100 million base pairs
  - Matching each sequence with a 1,000 base pair query takes about 3 hours!
- Heuristic algorithms aim at speeding up at the price of possibly missing the best scoring alignment
- Two well known programs
  - BLAST: Basic Local Alignment Search Tool
  - FASTA: Fast Alignment Tool
  - Basic idea: first locate high-scoring short stretches and then extend them
BLAST (Basic Local Alignment Search Tool)

- Approach (BLAST) (Altschul et al. 1990, developed by NCBI)
  - View sequences as sequences of short words (k-tuple)
    - DNA: 11 bases, protein: 3 amino acids
  - Create hash table of neighborhood (closely-matching) words
  - Use statistics to set threshold for “closeness”
  - Start from exact matches to neighborhood words

- Motivation
  - Good alignments should contain many close matches
  - Statistics can determine which matches are significant
    - Much more sensitive than % identity
  - Hashing can find matches in O(n) time
  - Extending matches in both directions finds alignment
    - Yields high-scoring/maximum segment pairs (HSP/MSP)
**BLAST (Basic Local Alignment Search Tool)**

1) Convert 1st sequence into words (using all frames for given word size)

2) Calculate for each word list of “neighborhood” words (scoring threshold $T$) and enter in dictionary

3) Scan 2nd sequence, find matching words in dictionary, store locations

4) For each match, extend alignment in both directions while score above threshold $S$, merge segments

5) Align best segments using dynamic programming, report statistically significant matches
Microarray Experiments

- Microarray chip with DNA sequences attaches in fixed grids.
- cDNA is produced from mRNA samples and labeled using either fluorescent dyes or radioactive isotopes.
- Hybridize cDNA over the microarray.
- Scan the microarray to read the signal intensity that reveals the expression level of transcribed genes.

www.affymetrix.com
Microarray Data

- Microarray data are usually transformed into an intensity matrix.
- The intensity matrix allows biologists to make correlations between different genes (even if they are dissimilar) and to understand how genes functions might be related.

<table>
<thead>
<tr>
<th>Gene</th>
<th>Time X</th>
<th>Time Y</th>
<th>Time Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gene 1</td>
<td>10</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>Gene 2</td>
<td>10</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>Gene 3</td>
<td>4</td>
<td>8.6</td>
<td>3</td>
</tr>
<tr>
<td>Gene 4</td>
<td>7</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>Gene 5</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Intensity (expression level) of gene at measured time
Microarray Data

- Track the sample over a period of time
- Track two different samples under the same conditions

Each box represents one gene’s expression over time
Microarray Data Analysis

- Clustering
  - Gene-based clustering: cluster genes based on their expression patterns
  - Sample-based clustering: cluster samples
  - Subspace clustering: capture clusters formed by a subset of genes across a subset of samples

- Classification
  - According to clinical syndromes or cancer types

- Association analysis

- Issues
  - Large number of genes
  - Limited number of samples
Outline

- Biological data mining
- Data mining for intrusion detection
- Privacy-preserving data mining
Intrusion Detection

- **Intrusions**: Any set of actions that threaten the integrity, availability, or confidentiality of a system or network resource

- **Intrusion detection**: The process of monitoring and analyzing the events occurring in a computer and/or network system in order to detect signs of security problems
IDS Architecture
Traditional Approaches

- **Misuse detection**: use patterns of well-known attacks to identify intrusions
- **Anomaly detection**: use deviation from normal usage patterns to identify intrusions
Problems of Traditional Approaches

- Main problems: manual and ad-hoc

- Misuse detection:
  - Known intrusion patterns have to be hand-coded
  - Unable to detect any new intrusions (that have no matched patterns recorded in the system)

- Anomaly detection:
  - Selecting the right set of system features to be measured is ad hoc and based on experience
  - Unable to capture sequential interrelation between events
  - High false positive rate
Data Mining Can Help

- Frequent pattern and association rules mining
  - Correlated features for attacks
    \{\text{Src IP}=206.163.27.95, \text{Dest Port}=139, \text{Bytes} \in [150, 200)\} \rightarrow \text{attack}
    \{\text{num\_failed\_login\_attempts} = 6, \text{service} = \text{FTP}\} \rightarrow \text{attack}
  - Correlated alerts for high-level attacks (Ning et al. CCS’02)

- Frequent sequential patterns
  - Capture the signatures for attacks in a series of events

- Classification
  - Classify a pattern -- decision tree, neural network, SVM, etc

- Clustering
  - Build clusters of normal activities and intrusions -> signatures

- Data stream mining
Case Study: Building Classifiers for Anomaly Detection (J. Stolfo et al.)

- **Network tcpdump data**
  - Packets of incoming, out-going, and internal broadcast traffic
  - One trace of normal network traffic and three traces of network intrusions

- **Extract the “connection” level features:**
  - start time and duration
  - participating hosts and ports (applications)
  - statistics (e.g., # of bytes)
  - flag: normal or a connection/termination error
  - protocol: TCP or UDP

- **Lessons learned**
  - Data preprocessing requires extensive domain knowledge
  - Adding temporal features improves classification accuracy
References

- C. Kruegel and G. Vigna. Anomaly detection of web-based attacks, in ACM CCS’03
- S. Mukkamala et al., Intrusion detection using neural networks and support vector machines, in IEEE IJCNN (May 2002).
- Bertrand Portier, Data Mining Techniques for Intrusion Detection
- S. Axelsson, Intrusion Detection Systems: A Survey and Taxonomy
- J. Allen et al., State of the Practice of Intrusion Detection Technologies
- Susan M. Bridges et al. DATA MINING AND GENETIC ALGORITHMS APPLIED TO INTRUSION DETECTION
Outline

- Biological data mining
- Data mining for intrusion detection
- Privacy-preserving data mining
Privacy Preserving Data Mining

- Constraints
  - Individual privacy
  - Organizational data confidentiality

- Goal of data mining is summary results
  - Association rules
  - Classifiers
  - Clusters

- The results alone need not violate privacy
  - Contain no individually identifiable values
  - Reflect overall results, not individual organizations

*The problem is computing the results without access to the data!*
Classes of Solutions

- Data Obfuscation
  - Nobody sees the *real* data

- Summarization
  - Only the needed facts are exposed

- Data Separation
  - Data remains with trusted parties
Data Obfuscation

- **Goal:** Hide the protected information

- **Approaches**
  - Randomly modify data
  - Swap values between records
  - Controlled modification of data to hide secrets

- **Problems**
  - Does it really protect the data?
  - Can we learn from the results?

- Randomization-based decision tree learning
  * (Agrawal & Srikant ’00)
Randomization Based Decision Tree Learning (Agrawal and Srikant ’00)

- **Basic idea:** Perturb Data with Value Distortion
  - User provides $x_i + r$ instead of $x_i$
  - $r$ is a random value
    - Uniform, uniform distribution between $[-\alpha, \alpha]$
    - Gaussian, normal distribution with $\mu = 0, \sigma$

- **Hypothesis**
  - Miner doesn’t see the real data or can’t reconstruct real values
  - Miner can reconstruct enough information to identify patterns
Randomization Approach Overview

Alice's age

Add random number to Age

30 becomes 65 (30+35)

Randomizer

Reconstruct Distribution of Age

Reconstruct Distribution of Salary

Classification Algorithm

Model
Output: A Decision Tree for “buys_computer”

- age?
  - <=30
    - student?
      - no
      - yes
    - yes
  - 31..40
  - >40
    - credit rating?
      - excellent
      - fair
      - yes
      - no
Attribute Selection Measure: Gini index (CART)

- If a data set $D$ contains examples from $n$ classes, gini index, $gini(D)$ is defined as

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

where $p_j$ is the relative frequency of class $j$ in $D$

- If a data set $D$ is split on $A$ into two subsets $D_1$ and $D_2$, the gini index $gini(D)$ is defined as

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- Reduction in Impurity:

$$\Delta gini(A) = gini(D) - gini_A(D)$$

- The attribute provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node
Original Distribution Reconstruction

- $x_1, x_2, \ldots, x_n$ are the $n$ original data values
  - Drawn from $n$ iid random variables $X_1, X_2, \ldots, X_n$ similar to $X$

- Using value distortion,
  - The given values are $w_1 = x_1 + y_1$, $w_2 = x_2 + y_2$, $\ldots$, $w_n = x_n + y_n$
  - $y_i$’s are from $n$ iid random variables $Y_1, Y_2, \ldots, Y_n$ similar to $Y$

- Reconstruction Problem:
  - Given $F_Y$ and $w_i$’s, estimate $F_X$
Original Distribution Reconstruction: Method

- Bayes’ theorem for continuous distribution

\[
f(x|y) = \frac{f(x, y)}{f(y)} = \frac{f(y|x) f(x)}{f(y)} = \frac{f(y|x) f(x)}{\int_{-\infty}^{\infty} f(y|x) f(x) \, dx}.
\]

- The estimated density function:

\[
f'_X(a) = \frac{1}{n} \sum_{i=1}^{n} \frac{f_Y(w_i - a)f_X(a)}{\int_{-\infty}^{\infty} f_Y(w_i - z)f_X(z) \, dz}
\]

- Iterative estimation
  - The initial estimate for \(f_X\) at \(j=0\): uniform distribution
  - Iterative estimation

\[
f_{X}^{j+1}(a) = \frac{1}{n} \sum_{i=1}^{n} \frac{f_Y(w_i - a)f_{X}^{j}(a)}{\int_{-\infty}^{\infty} f_Y(w_i - z)f_{X}^{j}(z) \, dz}
\]

- Stopping Criterion: \(\chi^2\) test between successive iterations
Reconstruction of Distribution
Original Distribution Reconstruction
Original Distribution Construction for Decision Tree

- When are the distributions reconstructed?
  - Global
    - Reconstruct for each attribute once at the beginning
    - Build the decision tree using the reconstructed data
  - ByClass
    - First split the training data
    - Reconstruct for each class separately
    - Build the decision tree using the reconstructed data
  - Local
    - First split the training data
    - Reconstruct for each class separately
    - Reconstruct at each node while building the tree
Accuracy vs. Randomization Level

Fn 3
More Results

- Global performs worse than ByClass and Local
- ByClass and Local have accuracy within 5% to 15% (absolute error) of the Original accuracy
- Overall, all are much better than the Randomized accuracy
Follow-up Work

- Simple additive randomization
- Multiplicative randomization
- Geometric randomization
Summarization

- **Goal:** Make only innocuous summaries of data available

- **Approaches:**
  - Overall collection statistics
  - Limited query functionality

- **Problems:**
  - Can we deduce data from statistics?
  - Is the information sufficient?
Data Separation

- **Goal:** Only trusted parties see the data
- **Approaches:**
  - Data split among trusted parties and each agrees not to release or share the data
- **Problems:**
  - Can we learn global models without sharing the data?
  - Do the analysis results disclose private information?
Secure Multiparty Computation

- **Goal:** Compute function when each party has some of the inputs

- **Yao’s Millionaire’s problem** *(Yao ’86)*
  - Secure computation possible if function can be represented as a circuit of gates
  - Idea: Securely compute gate

- **Secure multi-party computation** *(Goldreich, Micali, and Wigderson ’87)*
  - Given a function $f$ and $n$ inputs distributed at $n$ sites, compute the result without revealing to any site anything except its own input(s) and the result.
  \[ y = f(x_1, x_2, \ldots, x_n) \]
Decision Tree Construction \textit{(Lindell \& Pinkas ‘00)}

- Scenario: two-party horizontal partitioning
  - Each site has same schema
  - Attribute set known
  - Individual entities private
- Problem: Learn a decision tree classifier ID3 while meeting Secure Multiparty Computation Definitions
- Key assumptions
  - Semi-honest model
  - Only Two-party case considered
    - Extension to multiple parties is not trivial
  - Deals only with categorical attributes
ID3

- R - the set of attributes
- C - the class attribute
- T - the set of transactions

ID3(R, C, T)

1. If R is empty, return a leaf-node with the class value assigned to the most transactions in T.

2. If T consists of transactions which all have the same value c for the class attribute, return a leaf-node with the value c (finished classification path).

3. Otherwise,
   
   (a) Determine the attribute that best classifies the transactions in T, let it be A.
   
   (b) Let a₁, ..., aₘ be the values of attribute A and let T(a₁), ..., T(aₘ) be a partition of T such that every transaction in T(aᵢ) has the attribute value aᵢ.
   
   (c) Return a tree whose root is labeled A (this is the test attribute) and has edges labeled a₁, ..., aₘ such that for every i, the edge aᵢ goes to the tree ID3(R − {A}, C, T(aᵢ)).
Privacy Preserving ID3

Step 1: If $R$ is empty, return a leaf-node with the class value assigned to the most transactions in $T$

- Inputs: $(|T_1(c_1)|, \ldots, |T_1(c_L)|), (|T_2(c_1)|, \ldots, |T_2(c_L)|)$
- Output: $i$ where $|T_1(c_i)| + |T_2(c_i)|$ is largest
- Yao’s protocol
Privacy Preserving ID3

Step 2: If $T$ consists of transactions which have all the same value $c$ for the class attribute, return a leaf node with the value $c$

- Input: Either a symbol representing having more than one class or $c_i$
- Output: whether they have the same class attribute
- Equality checking protocols
  - Yao’86
  - Fagin, Naor ’96
  - Naor, Pinkas ‘01
Privacy Preserving ID3

- Step 3(a): *Determine the attribute that best classifies the transactions in T, let it be A*
  - Essentially done by securely computing $x^*(\ln x)$

- Step 3(b,c): *Recursively call $ID3_\delta$ for the remaining attributes on the transaction sets $T(a_1), \ldots, T(a_m)$ where $a_1, \ldots, a_m$ are the values of the attribute A*
  - Since the results of 3(a) and the attribute values are public, both parties can individually partition the database and prepare their inputs for the recursive calls
Summary

- Privacy and Security Constraints can be impediments to data mining
  - Problems with access to data
  - Restrictions on sharing
  - Limitations on use of results

- Technical solutions possible
  - Randomizing / swapping data doesn’t prevent learning good models
  - We don’t need to share data to learn global results

- Still lots of work to do!