Integration of Classification and Pattern Mining: A Discriminative and Frequent Pattern-Based Approach

Hong Cheng
Chinese Univ. of Hong Kong
hcheng@se.cuhk.edu.hk

Xifeng Yan
Univ. of California at Santa Barbara
xyan@cs.ucsb.edu

Jiawei Han
Univ. of Illinois at U-C
hanj@cs.uiuc.edu

Philip S. Yu
Univ. of Illinois at Chicago
psyu@cs.uic.edu
Tutorial Outline

- Frequent Pattern Mining
- Classification Overview
- Associative Classification
- Substructure-Based Graph Classification
- Direct Mining of Discriminative Patterns
- Integration with Other Machine Learning Techniques
- Conclusions and Future Directions
Frequent Patterns

Frequent Itemsets

TID | Items bought
---|----------------
10  | Beer, Nuts, Diaper
20  | Beer, Coffee, Diaper
30  | Beer, Diaper, Eggs
40  | Nuts, Eggs, Milk
50  | Nuts, Diaper, Eggs, Beer

Frequent Graphs

frequent pattern: support no less than min_sup
min_sup: the minimum frequency threshold
Major Mining Methodologies

- **Apriori approach**
  - Candidate generate-and-test, breadth-first search
  - Apriori, GSP, AGM, FSG, PATH, FFSM

- **Pattern-growth approach**
  - Divide-and-conquer, depth-first search
  - FP-Growth, PrefixSpan, MoFa, gSpan, Gaston

- **Vertical data approach**
  - ID list intersection with (item: tid list) representation
  - Eclat, CHARM, SPADE
Apriori Approach

• Join two size-k patterns to a size-(k+1) pattern

• Itemset: \{a,b,c\} + \{a,b,d\} \rightarrow \{a,b,c,d\}

• Graph:
Pattern Growth Approach

- Depth-first search, grow a size-k pattern to size-(k+1) one by adding one element

- Frequent subgraph mining
Vertical Data Approach

• Major operation: transaction list intersection

\[ t(AB) = t(A) \cap t(B) \]

<table>
<thead>
<tr>
<th>Item</th>
<th>Transaction id</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>t1, t2, t3,…</td>
</tr>
<tr>
<td>B</td>
<td>t2, t3, t4,…</td>
</tr>
<tr>
<td>C</td>
<td>t1, t3, t4,…</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Mining High Dimensional Data

• High dimensional data
  – Microarray data with 10,000 – 100,000 columns

• Row enumeration rather than column enumeration
  – CARPENTER [Pan et al., KDD’03]
  – COBBLER [Pan et al., SSDBM’04]
  – TD-Close [Liu et al., SDM’06]
Mining Colossal Patterns
[Zhu et al., ICDE’07]

• **Mining colossal patterns: challenges**
  - A small number of colossal (i.e., large) patterns, but a very large number of mid-sized patterns
  - If the mining of mid-sized patterns is explosive in size, there is no hope to find colossal patterns efficiently by insisting “complete set” mining philosophy

• **A pattern-fusion approach**
  - Jump out of the swamp of mid-sized results and quickly reach colossal patterns
  - Fuse small patterns to large ones directly
Impact to Other Data Analysis Tasks

• Association and correlation analysis
  – Association: support and confidence
  – Correlation: lift, chi-square, cosine, all_confidence, coherence
  – A comparative study [Tan, Kumar and Srivastava, KDD’02]

• Frequent pattern-based Indexing
  – Sequence Indexing [Cheng, Yan and Han, SDM’05]
  – Graph Indexing [Yan, Yu and Han, SIGMOD’04; Cheng et al.,
    SIGMOD’07; Chen et al., VLDB’07]

• Frequent pattern-based clustering
  – Subspace clustering with frequent itemsets
    • CLIQUE [Agrawal et al., SIGMOD’98]
    • ENCLUS [Cheng, Fu and Zhang, KDD’99]
    • pCluster [Wang et al., SIGMOD’02]

• Frequent pattern-based classification
  – Build classifiers with frequent patterns (our focus in this talk!)
Classification Overview

Training Instances → Model Learning → Prediction Model → Positive, Negative

Test Instances
Existing Classification Methods

- Support Vector Machine
- Decision Tree
- Neural Network
- Bayesian Network

And many more…

2008-12-23 ICDM 08 Tutorial
Many Classification Applications

- Literature and Art
- Journalism
- Education
- Society and Culture
- ...

Text Categorization

Free Text → Text Categorization System → Classified Text with Category Label

Drug Design

Spam Detection Classifier

Acceptable

Rejectable

Suspicious

Spam Detection

Face Recognition

2008-12-23

ICDM 08 Tutorial
Major Data Mining Themes

- Frequent Pattern Analysis
- Classification
- Clustering
- Outlier Analysis

Frequent Pattern-Based Classification
Why Pattern-Based Classification?

- Feature construction
  - Higher order
  - Compact
  - Discriminative

- Complex data modeling
  - Sequences
  - Graphs
  - Semi-structured/unstructured data
Feature Construction

Phrases vs. single words

... the long-awaited Apple iPhone has arrived ...
... the best apple pie recipe ...

Sequences vs. single commands

... login, changeDir, deleteFile, appendFile, logout ...
... login, setFileType, storeFile, logout ...

disambiguation

temporal order

higher order, discriminative
### Complex Data Modeling

#### Training Instances

<table>
<thead>
<tr>
<th>age</th>
<th>income</th>
<th>credit</th>
<th>Buy?</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>80k</td>
<td>good</td>
<td>Yes</td>
</tr>
<tr>
<td>50</td>
<td>200k</td>
<td>good</td>
<td>No</td>
</tr>
<tr>
<td>32</td>
<td>50k</td>
<td>fair</td>
<td>No</td>
</tr>
</tbody>
</table>

#### Classification Model

- **Predefined Feature vector**
- **NO Predefined Feature vector**

#### Prediction Model
Discriminative Frequent Pattern-Based Classification

- Discriminative Frequent Patterns
- Model Learning
- Feature Space Transformation
- Prediction Model
- Positive
- Negative

Training Instances → Discriminative Frequent Patterns → Feature Space Transformation → Prediction Model → Positive, Negative

Test Instances → Discriminative Frequent Patterns → Feature Space Transformation → Prediction Model → Positive, Negative
### Pattern-Based Classification on Transactions

#### Attributes

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, B, C</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>A, B, C</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
</tr>
<tr>
<td>A, B</td>
<td>1</td>
</tr>
<tr>
<td>A, C</td>
<td>0</td>
</tr>
<tr>
<td>B, C</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Frequent Itemset Support

<table>
<thead>
<tr>
<th>Frequent Itemset</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>3</td>
</tr>
<tr>
<td>AC</td>
<td>3</td>
</tr>
<tr>
<td>BC</td>
<td>3</td>
</tr>
</tbody>
</table>

#### Mining

- **min_sup=3**

#### Augmented

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>AB</th>
<th>AC</th>
<th>BC</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

---

2008-12-23

ICDM 08 Tutorial
Pattern-Based Classification on Graphs

Frequent Graphs

Mining $\min_{\text{sup}}=2$

Inactive

Active

Inactive

Transform

<table>
<thead>
<tr>
<th>g1</th>
<th>g2</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Applications: Drug Design

![Chemical Structures]

Test Chemical Compound

Class = Active / Inactive

Courtesy of Nikil Wale
Applications: Bug Localization

Correct executions  Incorrect executions

Calling graph:
1: makepat
2: esc
3: addstr
4: getline
5: dodash
6: in_set_2
7: stclose

Courtesy of Chao Liu
Tutorial Outline

- Frequent Pattern Mining
- Classification Overview
- Associative Classification
- Substructure-Based Graph Classification
- Direct Mining of Discriminative Patterns
- Integration with Other Machine Learning Techniques
- Conclusions and Future Directions
Associative Classification

- **Data**: transactional data, microarray data

- **Pattern**: frequent itemsets and association rules

- **Representative work**
  - CBA [Liu, Hsu and Ma, KDD’98]
  - Emerging patterns [Dong and Li, KDD’99]
  - CMAR [Li, Han and Pei, ICDM’01]
  - CPAR [Yin and Han, SDM’03]
  - RCBT [Cong et al., SIGMOD’05]
  - Lazy classifier [Veloso, Meira and Zaki, ICDM’06]
  - Integrated with classification models [Cheng et al., ICDE’07]
CBA [Liu, Hsu and Ma, KDD’98]

• Basic idea
  • Mine high-confidence, high-support class association rules with Apriori
  • Rule LHS: a conjunction of conditions
  • Rule RHS: a class label
  • Example:

R1: age < 25 & credit = ‘good’ → buy iPhone (sup=30%, conf=80%)
R2: age > 40 & income < 50k → not buy iPhone (sup=40%, conf=90%)
CBA

• Rule mining
  • Mine the set of association rules \( \text{wrt.} \) \( \text{min}_\text{sup} \) and \( \text{min}_\text{conf} \)
  • Rank rules in descending order of confidence and support
  • Select rules to ensure training instance coverage

• Prediction
  • Apply the first rule that matches a test case
  • Otherwise, apply the default rule
CMAR [Li, Han and Pei, ICDM’01]

• **Basic idea**
  – Mining: build a class distribution-associated FP-tree
  – Prediction: combine the strength of multiple rules

• **Rule mining**
  – Mine association rules from a class distribution-associated FP-tree
  – Store and retrieve association rules in a CR-tree
  – Prune rules based on confidence, correlation and database coverage
### Class Distribution-Associated FP-tree

#### Header Table

<table>
<thead>
<tr>
<th>Item</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td></td>
</tr>
<tr>
<td>b2</td>
<td></td>
</tr>
<tr>
<td>c1</td>
<td></td>
</tr>
<tr>
<td>d3</td>
<td></td>
</tr>
</tbody>
</table>

#### (a) FP-tree

- root
  - a1
    - cl (A:1)
    - b2
      - cl (B:1)
  - d3 (A:1)
    - d3 (C:1)

#### (b) FP-tree after merging nodes of d3

- root
  - a1
    - cl (A:1)
  - b2
    - cl (B:1, C:1)
CR-tree: A Prefix-tree to Store and Index Rules

Header Table

<table>
<thead>
<tr>
<th>item</th>
<th>head of node-links</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td></td>
</tr>
<tr>
<td>e</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule-id</th>
<th>Rule</th>
<th>Support</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>abc → A</td>
<td>80</td>
<td>80%</td>
</tr>
<tr>
<td>2</td>
<td>abcd → A</td>
<td>63</td>
<td>90%</td>
</tr>
<tr>
<td>3</td>
<td>abe → B</td>
<td>36</td>
<td>60%</td>
</tr>
<tr>
<td>4</td>
<td>bcd → C</td>
<td>210</td>
<td>70%</td>
</tr>
</tbody>
</table>
Prediction Based on Multiple Rules

• All rules matching a test case are collected and grouped based on class labels. The group with the most strength is used for prediction.

• Multiple rules in one group are combined with a weighted chi-square as:

\[ \sum \frac{\chi^2}{\max \chi^2} \]

where \( \max \chi^2 \) is the upper bound of chi-square of a rule.
CPAR [Yin and Han, SDM’03]

- **Basic idea**
  - Combine associative classification and FOIL-based rule generation
  - **Foil gain**: criterion for selecting a literal
    
    \[
    \text{gain}(p) = |P^*| \left( \log \frac{|P^*|}{|P^*| + |N^*|} - \log \frac{|P|}{|P| + |N|} \right)
    \]
  - Improve accuracy over traditional rule-based classifiers
  - Improve efficiency and reduce number of rules over association rule-based methods
CPAR

• Rule generation
  – Build a rule by adding literals one by one in a greedy way according to foil gain measure
  – Keep all close-to-the-best literals and build several rules simultaneously

• Prediction
  – Collect all rules matching a test case
  – Select the best k rules for each class
  – Choose the class with the highest expected accuracy for prediction
## Performance Comparison

[Yin and Han, SDM’03]

<table>
<thead>
<tr>
<th>Data</th>
<th>C4.5</th>
<th>Ripper</th>
<th>CBA</th>
<th>CMAR</th>
<th>CPAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>anneal</td>
<td>94.8</td>
<td>95.8</td>
<td>97.9</td>
<td>97.3</td>
<td>98.4</td>
</tr>
<tr>
<td>austral</td>
<td>84.7</td>
<td>87.3</td>
<td>84.9</td>
<td>86.1</td>
<td>86.2</td>
</tr>
<tr>
<td>auto</td>
<td>80.1</td>
<td>72.8</td>
<td>78.3</td>
<td>78.1</td>
<td>82.0</td>
</tr>
<tr>
<td>breast</td>
<td>95.0</td>
<td>95.1</td>
<td>96.3</td>
<td>96.4</td>
<td>96.0</td>
</tr>
<tr>
<td>cleve</td>
<td>78.2</td>
<td>82.2</td>
<td>82.8</td>
<td>82.2</td>
<td>81.5</td>
</tr>
<tr>
<td>crx</td>
<td>84.9</td>
<td>84.9</td>
<td>84.7</td>
<td>84.9</td>
<td>85.7</td>
</tr>
<tr>
<td>diabetes</td>
<td>74.2</td>
<td>74.7</td>
<td>74.5</td>
<td>75.8</td>
<td>75.1</td>
</tr>
<tr>
<td>german</td>
<td>72.3</td>
<td>69.8</td>
<td>73.4</td>
<td>74.9</td>
<td>73.4</td>
</tr>
<tr>
<td>glass</td>
<td>68.7</td>
<td>69.1</td>
<td>73.9</td>
<td>70.1</td>
<td>74.4</td>
</tr>
<tr>
<td>heart</td>
<td>80.8</td>
<td>80.7</td>
<td>81.9</td>
<td>82.2</td>
<td>82.6</td>
</tr>
<tr>
<td>hepatic</td>
<td>80.6</td>
<td>76.7</td>
<td>81.8</td>
<td>80.5</td>
<td>79.4</td>
</tr>
<tr>
<td>horse</td>
<td>82.6</td>
<td>84.8</td>
<td>82.1</td>
<td>82.6</td>
<td>84.2</td>
</tr>
<tr>
<td>hypo</td>
<td>99.2</td>
<td>98.9</td>
<td>98.9</td>
<td>98.4</td>
<td>98.1</td>
</tr>
<tr>
<td>iono</td>
<td>90.0</td>
<td>91.2</td>
<td>92.3</td>
<td>91.5</td>
<td>92.6</td>
</tr>
<tr>
<td>iris</td>
<td>95.3</td>
<td>94.0</td>
<td>94.7</td>
<td>94.0</td>
<td>94.7</td>
</tr>
<tr>
<td>labor</td>
<td>79.3</td>
<td>84.0</td>
<td>86.3</td>
<td>89.7</td>
<td>84.7</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Average</td>
<td>83.34</td>
<td>82.93</td>
<td>84.69</td>
<td>85.22</td>
<td>85.17</td>
</tr>
</tbody>
</table>
Emerging Patterns
[Dong and Li, KDD’99]

• Emerging Patterns (EPs) are contrast patterns between two classes of data whose support changes significantly between the two classes.

• Change significance can be defined by:

  big support ratio:
  \[
  \frac{\text{supp}_2(X)}{\text{supp}_1(X)} \geq \text{minRatio}
  \]
  similar to RiskRatio

  big support difference:
  \[
  |\text{supp}_2(X) - \text{supp}_1(X)| \geq \text{minDiff}
  \]
  defined by Bay+Pazzani 99

• If \(\frac{\text{supp}_2(X)}{\text{supp}_1(X)} = \infty\), then \(X\) is a *jumping EP*.
  – jumping EP occurs in one class but never occurs in the other class.

*Courtesy of Bailey and Dong*
A Typical EP in the Mushroom Dataset

• The Mushroom dataset contains two classes: edible and poisonous

• Each data tuple has several features such as: odor, ring-number, stalk-surface-bellow-ring, etc.

• Consider the pattern
  \{odor = none,
  stalk-surface-bellow-ring = smooth,
  ring-number = one\}

  Its support increases from 0.2% in the poisonous class to 57.6% in the edible class (a growth rate of 288).

Courtesy of Bailey and Dong
EP-Based Classification: CAEP
[Dong et al, DS’99]

• Given a test case T, obtain T’s scores for each class, by aggregating the discriminating power of EPs contained in T; assign the class with the maximal score as T’s class.

• The discriminating power of EPs are expressed in terms of supports and growth rates. Prefer large supRatio, large support

• The contribution of one EP X (support weighted confidence):

\[
\text{strength}(X) = \frac{\text{sup}(X) \times \text{supRatio}(X)}{\text{supRatio}(X)+1}
\]

• Given a test T and a set E(Ci) of EPs for class Ci, the aggregate score of T for Ci is

\[
\text{score}(T, Ci) = \sum \text{strength}(X)
\]

(over X of Ci matching T)

• For each class, may use median (or 85%) aggregated value to normalize to avoid bias towards class with more EPs

Courtesy of Bailey and Dong
Top-k Covering Rule Groups for Gene Expression Data [Cong et al., SIGMOD’05]

• **Problem**
  – Mine strong association rules to reveal correlation between gene expression patterns and disease outcomes
  – Example: \(gene_1[a_1, b_1], ..., gene_n[a_n, b_n] \rightarrow class\)
  – Build a rule-based classifier for prediction

• **Challenges**: high dimensionality of data
  – Extremely long mining time
  – Huge number of rules generated

• **Solution**
  – Mining top-k covering rule groups with row enumeration
  – A classifier RCBT based on top-k covering rule groups
A Microarray Dataset

<table>
<thead>
<tr>
<th>Class</th>
<th>Gene1</th>
<th>Gene2</th>
<th>Gene3</th>
<th>Gene4</th>
<th>Gene5</th>
<th>Gene6</th>
<th>GeneN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample1</td>
<td>Cancer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample2</td>
<td>Cancer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SampleN-1</td>
<td>~Cancer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SampleN</td>
<td>~Cancer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Find closed patterns which occur frequently among genes.
- Find rules which associate certain combination of the columns that affect the class of the rows
  - Gene1, Gene10, Gene1001 -> Cancer

*Courtesy of Anthony Tung*
Top-k Covering Rule Groups

• Rule group
  – A set of rules which are supported by the same set of transactions
    \[ G = \{ A_i \rightarrow C \mid A_i \subseteq I \} \]
  – Rules in one group have the same \textit{sup} and \textit{conf}
  – Reduce the number of rules by clustering them into groups

• Mining top-k covering rule groups
  – For a row \( r_i \), the set of rule groups \( \{ \gamma_{i,j} \}, j \in [1,k] \)
    satisfying \textit{mins}up and there is no more significant rule groups
Row Enumeration

Figure 1: Running Example

<table>
<thead>
<tr>
<th>$i$</th>
<th>$r_i$</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a, b, c, d, e</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>a, b, c, o, p</td>
<td>C</td>
</tr>
<tr>
<td>3</td>
<td>c, d, e, f, g</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>c, d, e, f, g</td>
<td>$\neg C$</td>
</tr>
<tr>
<td>5</td>
<td>e, f, g, h, o</td>
<td>$\neg C$</td>
</tr>
</tbody>
</table>

(b) $TT|_{\emptyset}$ (or $TT$)

<table>
<thead>
<tr>
<th>$i_j$</th>
<th>$\mathcal{R}(i_j)$</th>
<th>$\neg \mathcal{R}(i_j)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>1, 2</td>
<td></td>
</tr>
<tr>
<td>$b$</td>
<td>1, 2</td>
<td></td>
</tr>
<tr>
<td>$c$</td>
<td>1, 2, 3</td>
<td>4</td>
</tr>
<tr>
<td>$d$</td>
<td>1, 3</td>
<td>4</td>
</tr>
<tr>
<td>$e$</td>
<td>1, 3</td>
<td>4, 5</td>
</tr>
<tr>
<td>$f$</td>
<td>3</td>
<td>4, 5</td>
</tr>
<tr>
<td>$g$</td>
<td>3</td>
<td>4, 5</td>
</tr>
<tr>
<td>$h$</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>$o$</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>$p$</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

(c) $TT|_{\{1\}}$

<table>
<thead>
<tr>
<th>$i_j$</th>
<th>$\mathcal{R}(i_j)$</th>
<th>$\neg \mathcal{R}(i_j)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$b$</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$c$</td>
<td>2, 3</td>
<td>4</td>
</tr>
<tr>
<td>$d$</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$e$</td>
<td>3</td>
<td>4, 5</td>
</tr>
</tbody>
</table>

(d) $TT|_{\{1,3\}}$

Figure 2: Row Enumeration Tree.
TopkRGS Mining Algorithm

• Perform a depth-first traversal of a row enumeration tree
• \( \gamma_{r_i,j} \) for row \( r_i \) are initialized
• **Update**
  – If a new rule is more significant than existing rule groups, insert it
• **Pruning**
  – If the confidence upper bound of a subtree \( X \) is below the minconf of current top-k rule groups, prune \( X \)
RCBT

- RCBT uses a set of matching rules for a collective decision
- Given a test data \( t \), assume \( t \) satisfies \( m_i \) rules of class \( c_i \), the classification score of class \( c_i \) is

\[
Score(t)^{c_i} = \left( \sum_{i=1}^{m_i} S(\gamma(t)^{c_i}) \right) / S_{norm}^{c_i}
\]

where the score of a single rule is

\[
S(\gamma^{c_i}) = \gamma^{c_i}.conf * \gamma^{c_i}.sup / d_{c_i}
\]
Mining Efficiency

(a) ALL-AML leukemia

(b) Lung Cancer
## Classification Accuracy

<table>
<thead>
<tr>
<th>Dataset</th>
<th>RCBT</th>
<th>CBA</th>
<th>IRG Classifier</th>
<th>C4.5 family</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>single tree</td>
<td>bagging</td>
</tr>
<tr>
<td>AML/ALL (ALL)</td>
<td>91.18%</td>
<td>91.18%</td>
<td>64.71%</td>
<td>91.18%</td>
<td>91.18%</td>
</tr>
<tr>
<td>Lung Cancer(LC)</td>
<td>97.99%</td>
<td>81.88%</td>
<td>89.93%</td>
<td>81.88%</td>
<td>96.64%</td>
</tr>
<tr>
<td>Ovarian Cancer(OC)</td>
<td>97.67%</td>
<td>93.02%</td>
<td>-</td>
<td>97.67%</td>
<td>97.67%</td>
</tr>
<tr>
<td>Prostate Cancer(PC)</td>
<td>97.06%</td>
<td>82.35%</td>
<td>88.24%</td>
<td>26.47%</td>
<td>26.47%</td>
</tr>
<tr>
<td>Average Accuracy</td>
<td>95.98%</td>
<td>87.11%</td>
<td>80.96%</td>
<td>74.3%</td>
<td>77.99%</td>
</tr>
</tbody>
</table>

Table 2: Classification Results
Lazy Associative Classification
[Veloso, Meira, Zaki, ICDM’06]

• Basic idea
  – Simply stores training data, and the classification model (CARs) is built after a test instance is given
  • For a test case $t$, project training data $D$ on $t$
  • Mine association rules from $D_t$
  • Select the best rule for prediction

  – Advantages
    • Search space is reduced/focused
      – Cover small disjuncts (support can be lowered)
    • Only applicable rules are generated
      – A much smaller number of CARs are induced

  – Disadvantages
    • Several models are generated, one for each test instance
    • Potentially high computational cost

Courtesy of Mohammed Zaki
Caching for Lazy CARs

• Models for different test instances may share some CARs
  – Avoid work replication by caching common CARs

• Cache infrastructure
  – All CARs are stored in main memory
  – Each CAR has only one entry in the cache
  – Replacement policy
    • LFU heuristic

*Courtesy of Mohammed Zaki*
Integrated with Classification Models [Cheng et al., ICDE’07]

- **Framework**
  - Feature construction
    - Frequent itemset mining
  - Feature selection
    - Select discriminative features
    - Remove redundancy and correlation
  - Model learning
    - A general classifier based on SVM or C4.5 or other classification model
Information Gain vs. Frequency?

Information Gain Formula:

\[ IG(C \mid X) = H(C) - H(C \mid X) \]
Fisher Score vs. Frequency?

(a) Austral  
(b) Breast  
(c) Sonar

Fisher Score Formula:

\[ Fr = \frac{\sum_{i=1}^{c} n_i (u_i - u)^2}{\sum_{i=1}^{c} n_i \sigma_i^2} \]
Analytical Study on Information Gain

\[ IG(C \mid X) = H(C) - H(C \mid X) \]

- **Entropy**
  - Constant given data

- **Conditional Entropy**
  - Study focus
Information Gain Expressed by Pattern Frequency

$X$: feature; $C$: class labels

$$H(C \mid X) = - \sum_{c \in \{0,1\}} P(x) \sum_{c \in \{0,1\}} P(c \mid x) \log P(c \mid x)$$

- Entropy when feature appears ($x=1$)

$$H(C \mid X) = - \theta q \log q - \theta (1-q) \log(1-q)$$

- Entropy when feature not appears ($x=0$)

$$(\theta q - p) \log \frac{p - \theta q}{1 - \theta} + (\theta (1-q) - (1-p)) \log \frac{(1-p) - \theta (1-q)}{1 - \theta}$$

Conditional prob. of the positive class when pattern appears

$$q = P(c = 1 \mid x = 1)$$

Pattern frequency

$$\theta = P(x = 1)$$

Prob. of Positive Class

$$p = P(c = 1)$$

2008-12-23 ICDM 08 Tutorial 51
Conditional Entropy in a Pure Case

- When \( q = 1 \) (or \( q = 0 \))

\[
H(C \mid X) = -\theta q \log q - \theta (1 - q) \log (1 - q)
\]

\[
+ (\theta q - p) \log \frac{p - \theta q}{1 - \theta} + (\theta (1 - q) - (1 - p)) \log \frac{(1 - p) - \theta (1 - q)}{1 - \theta}
\]

\[
H(C \mid X)_{q=1} = (\theta - 1) \left( \frac{p - \theta}{1 - \theta} \log \frac{p - \theta}{1 - \theta} + \frac{1 - p}{1 - \theta} \log \frac{1 - p}{1 - \theta} \right)
\]
Frequent Is Informative

\[ H(C \mid X)_{q=1} = (\theta - 1)(\frac{p - \theta}{1 - \theta} \log \frac{p - \theta}{1 - \theta} + \frac{1 - p}{1 - \theta} \log \frac{1 - p}{1 - \theta}) \]

the \( H(C\mid X) \) minimum value when \( \theta \leq p \) (similar for \( q=0 \))

Take a partial derivative

\[ \frac{\partial H(C \mid X)_{q=1}}{\partial \theta} = \log \frac{p - \theta}{1 - \theta} \leq \log 1 = 0 \text{ since } \theta \leq p \leq 1 \]

\( H(C\mid X) \) lower bound is monotonically decreasing with frequency

\( IG(C\mid X) \) upper bound is monotonically increasing with frequency
Too Frequent is Less Informative

• For $\theta \geq p$, we have a similar conclusion:

  \[ H(C|X) \text{ lower bound is monotonically increasing with frequency} \]
  \[ \text{IG}(C|X) \text{ upper bound is monotonically decreasing with frequency} \]

• Similar analysis on Fisher score
## Accuracy

<table>
<thead>
<tr>
<th>Data</th>
<th>Item_All*</th>
<th>Item_FS</th>
<th>Pat_All</th>
<th>Pat_FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>austral</td>
<td>85.01</td>
<td>85.50</td>
<td>81.79</td>
<td>91.14</td>
</tr>
<tr>
<td>auto</td>
<td>83.25</td>
<td>84.21</td>
<td>74.97</td>
<td>90.79</td>
</tr>
<tr>
<td>cleve</td>
<td>84.81</td>
<td>84.81</td>
<td>78.55</td>
<td>95.04</td>
</tr>
<tr>
<td>diabetes</td>
<td>74.41</td>
<td>74.41</td>
<td>77.73</td>
<td>78.31</td>
</tr>
<tr>
<td>glass</td>
<td>75.19</td>
<td>75.19</td>
<td>79.91</td>
<td>81.32</td>
</tr>
<tr>
<td>heart</td>
<td>84.81</td>
<td>84.81</td>
<td>82.22</td>
<td>88.15</td>
</tr>
<tr>
<td>iono</td>
<td>93.15</td>
<td>94.30</td>
<td>89.17</td>
<td>95.44</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data</th>
<th>Item_All</th>
<th>Item_FS</th>
<th>Pat_All</th>
<th>Pat_FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>austral</td>
<td>84.53</td>
<td>84.53</td>
<td>84.21</td>
<td>88.24</td>
</tr>
<tr>
<td>auto</td>
<td>71.70</td>
<td>77.63</td>
<td>71.14</td>
<td>78.77</td>
</tr>
<tr>
<td>cleve</td>
<td>80.87</td>
<td>80.87</td>
<td>80.84</td>
<td>91.42</td>
</tr>
<tr>
<td>diabetes</td>
<td>77.02</td>
<td>77.02</td>
<td>76.00</td>
<td>76.58</td>
</tr>
<tr>
<td>glass</td>
<td>75.24</td>
<td>75.24</td>
<td>76.62</td>
<td>79.89</td>
</tr>
<tr>
<td>heart</td>
<td>81.85</td>
<td>81.85</td>
<td>80.00</td>
<td>86.30</td>
</tr>
<tr>
<td>iono</td>
<td>92.30</td>
<td>92.30</td>
<td>92.89</td>
<td>94.87</td>
</tr>
</tbody>
</table>

Accuracy based on SVM

Accuracy based on Decision Tree

* **Item_All**: all single features
  
* **Pat_All**: all frequent patterns
  
* **Item_FS**: single features with selection
  
* **Pat_FS**: frequent patterns with selection
## Classification with A Small Feature Set

**Accuracy and Time on Chess**

<table>
<thead>
<tr>
<th>min_sup</th>
<th># Patterns</th>
<th>Time</th>
<th>SVM (%)</th>
<th>Decision Tree (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>2000</td>
<td>68,967</td>
<td>44.70</td>
<td>92.52</td>
<td>97.59</td>
</tr>
<tr>
<td>2200</td>
<td>28,358</td>
<td>19.94</td>
<td>91.68</td>
<td>97.84</td>
</tr>
<tr>
<td>2500</td>
<td>6,837</td>
<td>2.91</td>
<td>91.68</td>
<td>97.62</td>
</tr>
<tr>
<td>2800</td>
<td>1,031</td>
<td>0.47</td>
<td>91.84</td>
<td>97.37</td>
</tr>
<tr>
<td>3000</td>
<td>136</td>
<td>0.06</td>
<td>91.90</td>
<td>97.06</td>
</tr>
</tbody>
</table>
Tutorial Outline

- Frequent Pattern Mining
- Classification Overview
- Associative Classification
- Substructure-Based Graph Classification
- Direct Mining of Discriminative Patterns
- Integration with Other Machine Learning Techniques
- Conclusions and Future Directions
Substructure-Based Graph Classification

- **Data:** graph data with labels, e.g., chemical compounds, software behavior graphs, social networks

- **Basic idea**
  - Extract graph substructures \( F = \{g_1, \ldots, g_n\} \)
  - Represent a graph with a feature vector \( X = \{x_1, \ldots, x_n\} \), where \( x_i \) is the frequency of \( g_i \) in that graph
  - Build a classification model

- **Different features and representative work**
  - Fingerprint
  - Maccs keys
  - Tree and cyclic patterns [Horvath et al., KDD’04]
  - Minimal contrast subgraph [Ting and Bailey, SDM’06]
  - Frequent subgraphs [Deshpande et al., TKDE’05; Liu et al., SDM’05]
  - Graph fragments [Wale and Karypis, ICDM’06]
Fingerprints (fp-n)

Chemical Compounds

Enumerate all paths up to length $l$ and certain cycles

Hash features to position(s) in a fixed length bit-vector

Courtesy of Nikil Wale
Maccs Keys (MK)

Each Fragment forms a fixed dimension in the descriptor-space

Domain Expert

Identify “Important” Fragments for bioactivity

Courtesy of Nikil Wale
Cycles and Trees (CT) [Horvath et al., KDD’04]

Identify Bi-connected components

Bounded Cyclicity Using Bi-connected components

Delete Bi-connected Components from the compound

Chemical Compound

Fixed number of cycles

Left-over Trees

 Courtesy of Nikil Wale
Frequent Subgraphs (FS) [Deshpande et al., TKDE’05]

Discovering Features

- Chemical Compounds

Topological features – captured by graph representation

- Discovered Subgraphs

Frequent Subgraph Discovery

Min. Support.

- Sup:+ve:30% -ve:5%
- Sup:+ve:40% -ve:0%
- Sup:+ve:1% -ve:30%

Courtesy of Nikil Wale
Graph Fragments (GF)
[Wale and Karypis, ICDM’06]

- Tree Fragments (TF): At least one node of the tree fragment has a degree greater than 2 (no cycles).

- Path Fragments (PF): All nodes have degree less than or equal to 2 but does not include cycles.

- Acyclic Fragments (AF): TF U PF
  - Acyclic fragments are also termed as free trees.

Courtesy of Nikil Wale
Comparison of Different Features
[Wale and Karypis, ICDM’06]

Table 1. Design choices made by the descriptor spaces.

<table>
<thead>
<tr>
<th>Previously developed descriptors</th>
<th>Topological Complexity</th>
<th>Generation</th>
<th>Precise</th>
<th>Complete Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MK</td>
<td>Low to High</td>
<td>static</td>
<td>Yes</td>
<td>Maybe</td>
</tr>
<tr>
<td>fp</td>
<td>Low</td>
<td>dynamic</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>CT</td>
<td>Medium</td>
<td>dynamic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>FS</td>
<td>Low to High</td>
<td>dynamic</td>
<td>Yes</td>
<td>Maybe</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GF-based descriptors</th>
<th>Topological Complexity</th>
<th>Generation</th>
<th>Precise</th>
<th>Complete Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF</td>
<td>Low</td>
<td>dynamic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>TF</td>
<td>Medium</td>
<td>dynamic</td>
<td>Yes</td>
<td>Maybe</td>
</tr>
<tr>
<td>AF</td>
<td>Medium</td>
<td>dynamic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GF</td>
<td>Low to High</td>
<td>dynamic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Minimal Contrast Subgraphs
[Ting and Bailey, SDM’06]

• A contrast graph is a subgraph appearing in one class of graphs and never in another class of graphs
  – Minimal if none of its subgraphs are contrasts
  – May be disconnected
    • Allows succinct description of differences
    • But requires larger search space
Mining Contrast Subgraphs

• Main idea
  – Find the maximal common edge sets
    • These may be disconnected
  – Apply a minimal hypergraph transversal operation to derive the minimal contrast edge sets from the maximal common edge sets
  – Must compute minimal contrast vertex sets separately and then minimal union with the minimal contrast edge sets

Courtesy of Bailey and Dong
Frequent Subgraph-Based Classification
[Deshpande et al., TKDE’05]

• **Frequent subgraphs**
  – A graph is frequent if its support (occurrence frequency) in a given dataset is no less than a minimum support threshold

• **Feature generation**
  – Frequent topological subgraphs by FSG
  – Frequent geometric subgraphs with 3D shape information

• **Feature selection**
  – Sequential covering paradigm

• **Classification**
  – Use SVM to learn a classifier based on feature vectors
  – Assign different misclassification costs for different classes to address skewed class distribution
Varying Minimum Support

TABLE 2  
Varying Minimum Support Threshold ($\sigma$)

<table>
<thead>
<tr>
<th>$D$</th>
<th>$\sigma = 10.0%$</th>
<th>$\sigma = 15.0%$</th>
<th>$\sigma = 20.0%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$N_f$</td>
<td>A</td>
<td>$N_f$</td>
</tr>
<tr>
<td>P1</td>
<td>66.0</td>
<td>1211</td>
<td>65.5</td>
</tr>
<tr>
<td>P2</td>
<td>65.0</td>
<td>967</td>
<td>64.0</td>
</tr>
<tr>
<td>P3</td>
<td>60.5</td>
<td>597</td>
<td>60.7</td>
</tr>
<tr>
<td>P4</td>
<td>54.3</td>
<td>275</td>
<td>55.4</td>
</tr>
<tr>
<td>H1</td>
<td>81.0</td>
<td>27034</td>
<td>82.1</td>
</tr>
<tr>
<td>H2</td>
<td>70.1</td>
<td>1797</td>
<td>76.0</td>
</tr>
<tr>
<td>H3</td>
<td>83.9</td>
<td>27019</td>
<td>89.5</td>
</tr>
<tr>
<td>A1</td>
<td>78.2</td>
<td>476</td>
<td>79.0</td>
</tr>
</tbody>
</table>

“$A$” denotes the area under the ROC curve and “$N_f$” denotes the number of discovered frequent subgraphs.
### Varying Misclassification Cost

**TABLE 4**
The Area under the ROC Curve Obtained by Varying the Misclassification Cost

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Topo</th>
<th>Geom</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\beta = 1.0$</td>
<td>$\beta = \text{EqCost}$</td>
</tr>
<tr>
<td>P1</td>
<td>65.5</td>
<td>65.3</td>
</tr>
<tr>
<td>P2</td>
<td>67.3</td>
<td>66.8</td>
</tr>
<tr>
<td>P3</td>
<td>62.6</td>
<td>62.6</td>
</tr>
<tr>
<td>P4</td>
<td>63.4</td>
<td>65.2</td>
</tr>
<tr>
<td>H1</td>
<td>81.0</td>
<td>79.2</td>
</tr>
<tr>
<td>H2</td>
<td>76.5</td>
<td>79.4</td>
</tr>
<tr>
<td>H3</td>
<td>83.9</td>
<td>90.8</td>
</tr>
<tr>
<td>A1</td>
<td>81.7</td>
<td>82.1</td>
</tr>
</tbody>
</table>

“$\beta = 1.0$” indicates the experiments in which each positive and negative example had a weight of one, and “$\beta = \text{EqCost}$” indicates the experiments in which the misclassification cost of the positive examples was increased to match the number of negative examples.
Frequent Subgraph-Based Classification for Bug Localization [Liu et al., SDM’05]

• **Basic idea**
  – Mine closed subgraphs from software behavior graphs
  – Build a graph classification model for software behavior prediction
  – Discover program regions that may contain bugs

• **Software behavior graphs**
  – Node: functions
  – Edge: function calls or transitions

![Software Behavior Graphs](image)

(a) one correct run  
(b) one incorrect run

**Figure 1: Software Behavior Graphs**
Bug Localization

- Identify suspicious functions relevant to incorrect runs
  - Gradually include more trace data
  - Build multiple classification models and estimate the accuracy boost
  - A function with a significant precision boost could be bug relevant

\[ P_B - P_A \] is the accuracy boost of function B
Case Study

Table 3: Bug-Relevant Functions with $\theta = 20\%$

<table>
<thead>
<tr>
<th>function name</th>
<th>Precision$_{in}$</th>
<th>Precision$_{out}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>main</td>
<td>0</td>
<td>58.462</td>
</tr>
<tr>
<td>getpat</td>
<td>0</td>
<td>33.808</td>
</tr>
<tr>
<td>makepat</td>
<td>0</td>
<td>33.808</td>
</tr>
<tr>
<td>change</td>
<td>33.886</td>
<td>58.462</td>
</tr>
<tr>
<td>subline</td>
<td>38.356</td>
<td>56.318</td>
</tr>
<tr>
<td>amatch</td>
<td>38.356</td>
<td>56.632</td>
</tr>
</tbody>
</table>

Figure 9: Entrance Precision and Exit Precision

2008-12-23 ICDM 08 Tutorial
Graph Fragment
[Wale and Karypis, ICDM’06]

• All graph substructures up to a given length (size or # of bonds)
  – Determined dynamically → Dataset dependent descriptor space
  – Complete coverage → Descriptors for every compound
  – Precise representation → One to one mapping
  – Complex fragments → Arbitrary topology

• Recurrence relation to generate graph fragments of length l

\[
F(G, l) = \begin{cases} 
  \emptyset, & \text{if } G \text{ has fewer than } l \text{ edges or } l = 0 \\
  eF(G'\setminus e, l - 1) \cup F(G\setminus e, l), & \text{otherwise},
\end{cases}
\]
Performance Comparison

Table 9. ROC50 values for the eight descriptors using kernels derived from RBF.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>GF ((K^*_b))</th>
<th>AF ((K^*_b))</th>
<th>TF ((K^*_b))</th>
<th>PF ((K^*_b))</th>
<th>fp-8192 ((K_b))</th>
<th>CT ((K_f))</th>
<th>MK ((K_f))</th>
<th>FS ((K_b))</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCI1</td>
<td>0.303</td>
<td>0.305</td>
<td>0.302</td>
<td>0.303</td>
<td>0.198</td>
<td>0.256</td>
<td>0.192</td>
<td>0.249</td>
</tr>
<tr>
<td>NCI109</td>
<td>0.285</td>
<td>0.292</td>
<td>0.293</td>
<td>0.288</td>
<td>0.199</td>
<td>0.228</td>
<td>0.202</td>
<td>0.232</td>
</tr>
<tr>
<td>NCI123</td>
<td>0.244</td>
<td>0.247</td>
<td>0.240</td>
<td>0.249</td>
<td>0.177</td>
<td>0.223</td>
<td>0.173</td>
<td>0.234</td>
</tr>
<tr>
<td>NCI145</td>
<td>0.318</td>
<td>0.322</td>
<td>0.321</td>
<td>0.317</td>
<td>0.203</td>
<td>0.255</td>
<td>0.194</td>
<td>0.258</td>
</tr>
<tr>
<td>NCI167</td>
<td>0.060</td>
<td>0.062</td>
<td>0.062</td>
<td>0.053</td>
<td>0.043</td>
<td>0.041</td>
<td>0.043</td>
<td>0.047</td>
</tr>
<tr>
<td>NCI220</td>
<td>0.248</td>
<td>0.263</td>
<td>0.266</td>
<td>0.261</td>
<td>0.272</td>
<td>0.218</td>
<td>0.393</td>
<td>0.198</td>
</tr>
<tr>
<td>NCI33</td>
<td>0.305</td>
<td>0.304</td>
<td>0.297</td>
<td>0.286</td>
<td>0.186</td>
<td>0.238</td>
<td>0.210</td>
<td>0.242</td>
</tr>
<tr>
<td>NCI330</td>
<td>0.313</td>
<td>0.317</td>
<td>0.306</td>
<td>0.311</td>
<td>0.235</td>
<td>0.305</td>
<td>0.241</td>
<td>0.241</td>
</tr>
<tr>
<td>NCI41</td>
<td>0.341</td>
<td>0.346</td>
<td>0.344</td>
<td>0.344</td>
<td>0.237</td>
<td>0.267</td>
<td>0.213</td>
<td>0.294</td>
</tr>
<tr>
<td>NCI47</td>
<td>0.298</td>
<td>0.295</td>
<td>0.271</td>
<td>0.289</td>
<td>0.194</td>
<td>0.232</td>
<td>0.186</td>
<td>0.227</td>
</tr>
<tr>
<td>NCI81</td>
<td>0.287</td>
<td>0.284</td>
<td>0.279</td>
<td>0.286</td>
<td>0.188</td>
<td>0.230</td>
<td>0.194</td>
<td>0.231</td>
</tr>
<tr>
<td>NCI83</td>
<td>0.298</td>
<td>0.301</td>
<td>0.298</td>
<td>0.300</td>
<td>0.197</td>
<td>0.258</td>
<td>0.204</td>
<td>0.253</td>
</tr>
<tr>
<td>H1</td>
<td>0.263</td>
<td>0.264</td>
<td>0.259</td>
<td>0.265</td>
<td>0.229</td>
<td>0.223</td>
<td>0.233</td>
<td>0.220</td>
</tr>
<tr>
<td>H2</td>
<td>0.633</td>
<td>0.636</td>
<td>0.629</td>
<td>0.635</td>
<td>0.573</td>
<td>0.556</td>
<td>0.545</td>
<td>0.575</td>
</tr>
<tr>
<td>A1</td>
<td>0.202</td>
<td>0.195</td>
<td>0.167</td>
<td>0.212</td>
<td>0.125</td>
<td>0.128</td>
<td>0.062</td>
<td>0.123</td>
</tr>
<tr>
<td>H3</td>
<td>0.637</td>
<td>0.631</td>
<td>0.628</td>
<td>0.632</td>
<td>0.578</td>
<td>0.589</td>
<td>0.584</td>
<td>0.554</td>
</tr>
<tr>
<td>D1</td>
<td>0.374</td>
<td>0.357</td>
<td>0.362</td>
<td>0.358</td>
<td>0.345</td>
<td>0.317</td>
<td>0.340</td>
<td>0.307</td>
</tr>
<tr>
<td>D2</td>
<td>0.585</td>
<td>0.592</td>
<td>0.571</td>
<td>0.545</td>
<td>0.567</td>
<td>0.558</td>
<td>0.551</td>
<td>0.486</td>
</tr>
<tr>
<td>D3</td>
<td>0.512</td>
<td>0.506</td>
<td>0.497</td>
<td>0.507</td>
<td>0.430</td>
<td>0.454</td>
<td>0.424</td>
<td>0.482</td>
</tr>
<tr>
<td>D4</td>
<td>0.465</td>
<td>0.470</td>
<td>0.458</td>
<td>0.460</td>
<td>0.401</td>
<td>0.426</td>
<td>0.380</td>
<td>0.400</td>
</tr>
<tr>
<td>P1</td>
<td>0.603</td>
<td>0.599</td>
<td>0.604</td>
<td>0.604</td>
<td>0.544</td>
<td>0.542</td>
<td>0.563</td>
<td>0.553</td>
</tr>
<tr>
<td>P2</td>
<td>0.502</td>
<td>0.500</td>
<td>0.468</td>
<td>0.492</td>
<td>0.532</td>
<td>0.493</td>
<td>0.512</td>
<td>0.465</td>
</tr>
<tr>
<td>P3</td>
<td>0.572</td>
<td>0.582</td>
<td>0.458</td>
<td>0.580</td>
<td>0.553</td>
<td>0.499</td>
<td>0.583</td>
<td>0.558</td>
</tr>
<tr>
<td>P4</td>
<td>0.623</td>
<td>0.625</td>
<td>0.605</td>
<td>0.622</td>
<td>0.542</td>
<td>0.559</td>
<td>0.536</td>
<td>0.594</td>
</tr>
<tr>
<td>C1</td>
<td>0.814</td>
<td>0.815</td>
<td>0.810</td>
<td>0.808</td>
<td>0.794</td>
<td>0.744</td>
<td>0.815</td>
<td>0.813</td>
</tr>
<tr>
<td>M1</td>
<td>0.440</td>
<td>0.439</td>
<td>0.414</td>
<td>0.429</td>
<td>0.428</td>
<td>0.343</td>
<td>0.411</td>
<td>0.410</td>
</tr>
<tr>
<td>M2</td>
<td>0.613</td>
<td>0.606</td>
<td>0.573</td>
<td>0.600</td>
<td>0.567</td>
<td>0.484</td>
<td>0.577</td>
<td>0.584</td>
</tr>
<tr>
<td>M3</td>
<td>0.786</td>
<td>0.773</td>
<td>0.779</td>
<td>0.768</td>
<td>0.785</td>
<td>0.749</td>
<td>0.788</td>
<td>0.775</td>
</tr>
<tr>
<td>ARQB</td>
<td>0.976</td>
<td>0.978</td>
<td>0.948</td>
<td>0.965</td>
<td>0.803</td>
<td>0.832</td>
<td>0.800</td>
<td>0.843</td>
</tr>
</tbody>
</table>
Tutorial Outline

- Frequent Pattern Mining
- Classification Overview
- Associative Classification
- Substructure-Based Graph Classification
- Direct Mining of Discriminative Patterns
- Integration with Other Machine Learning Techniques
- Conclusions and Future Directions
Re-examination of Pattern-Based Classification

Training Instances → Pattern-Based Feature Construction → Model Learning → Prediction Model → Positive, Negative

Feature Space Transformation

Computationally Expensive!
The Computational Bottleneck

Two steps, expensive

Data

Mining

Frequent Patterns $10^4 \sim 10^6$

Filtering

Discriminative Patterns

Direct mining, efficient

Data

Transform

FP-tree

Direct Mining

Discriminative Patterns
Challenge: Non Anti-Monotonic

Non-Monotonic: Enumerate all subgraphs then check their score?

Enumerate subgraphs: small-size to large-size

Non Monotonic

Anti-Monotonic
Direct Mining of Discriminative Patterns

• **Avoid mining the whole set of patterns**
  – Harmony [Wang and Karypis, SDM’05]
  – DDPMine [Cheng et al., ICDE’08]
  – LEAP [Yan et al., SIGMOD’08]
  – MbT [Fan et al., KDD’08]

• **Find the most discriminative pattern**
  – A search problem?
  – An optimization problem?

• **Extensions**
  – Mining top-k discriminative patterns
  – Mining approximate/weighted discriminative patterns
Harmony  
[Wang and Karypis, SDM’05]

- **Direct mining the best rules for classification**
  - Instance-centric rule generation: the highest confidence rule for each training case is included
  
  - Efficient search strategies and pruning methods
    - Support equivalence item (keep “generator itemset”)
      - e.g., prune (ab) if sup(ab)=sup(a)
    
    - Unpromising item or conditional database
      - Estimate confidence upper bound
      - Prune an item or a conditional db if it cannot generate a rule with higher confidence
  
  - Ordering of items in conditional database
    - Maximum confidence descending order
    - Entropy ascending order
    - Correlation coefficient ascending order
Harmony

• Prediction
  – For a test case, partition the rules into k groups based on class labels
  – Compute the score for each rule group
  – Predict based the rule group with the highest score
## Accuracy of Harmony

<table>
<thead>
<tr>
<th>Database</th>
<th>FOIL</th>
<th>CPAR</th>
<th>SVM</th>
<th>HARMONY</th>
</tr>
</thead>
<tbody>
<tr>
<td>adult</td>
<td>82.5</td>
<td>76.7</td>
<td>84.16</td>
<td>81.9</td>
</tr>
<tr>
<td>chess</td>
<td>42.6</td>
<td>32.8</td>
<td>29.83</td>
<td>44.87</td>
</tr>
<tr>
<td>connect</td>
<td>65.7</td>
<td>54.3</td>
<td>72.5</td>
<td>68.05</td>
</tr>
<tr>
<td>led7</td>
<td>62.3</td>
<td>71.2</td>
<td>73.78</td>
<td>74.56</td>
</tr>
<tr>
<td>letRecog</td>
<td>57.5</td>
<td>59.9</td>
<td>67.76</td>
<td>76.81</td>
</tr>
<tr>
<td>mushroom</td>
<td>99.5</td>
<td>98.8</td>
<td>99.67</td>
<td>99.94</td>
</tr>
<tr>
<td>nursery</td>
<td>91.3</td>
<td>78.5</td>
<td>91.35</td>
<td>92.83</td>
</tr>
<tr>
<td>pageBlocks</td>
<td>91.6</td>
<td>76.2</td>
<td>91.21</td>
<td>91.6</td>
</tr>
<tr>
<td>penDigits</td>
<td>88.0</td>
<td>83.0</td>
<td>93.2</td>
<td>96.23</td>
</tr>
<tr>
<td>waveform</td>
<td>75.6</td>
<td>75.4</td>
<td>83.16</td>
<td>80.46</td>
</tr>
<tr>
<td>average</td>
<td>75.66</td>
<td>70.68</td>
<td>78.663</td>
<td>80.725</td>
</tr>
</tbody>
</table>
## Runtime of Harmony

<table>
<thead>
<tr>
<th>Database</th>
<th>FOIL</th>
<th>CPAR</th>
<th>SVM</th>
<th>HARMONY</th>
</tr>
</thead>
<tbody>
<tr>
<td>adult</td>
<td>10251.0</td>
<td>809.0</td>
<td>2493.1</td>
<td>1395.5</td>
</tr>
<tr>
<td>chess</td>
<td>10122.8</td>
<td>1736.0</td>
<td>13289.4</td>
<td>11.34</td>
</tr>
<tr>
<td>connect</td>
<td>35572.5</td>
<td>24047.1</td>
<td>74541.1</td>
<td>85.44</td>
</tr>
<tr>
<td>led7</td>
<td>11.5</td>
<td>5.7</td>
<td>17.12</td>
<td>1.29</td>
</tr>
<tr>
<td>letRecog</td>
<td>4365.6</td>
<td>764.0</td>
<td>17825.2</td>
<td>778.91</td>
</tr>
<tr>
<td>mushroom</td>
<td>38.3</td>
<td>15.4</td>
<td>16.6</td>
<td>8.78</td>
</tr>
<tr>
<td>nursery</td>
<td>73.1</td>
<td>51.7</td>
<td>322.4</td>
<td>6.21</td>
</tr>
<tr>
<td>pageBlocks</td>
<td>43.1</td>
<td>15.5</td>
<td>11.2</td>
<td>2.5</td>
</tr>
<tr>
<td>penDigits</td>
<td>821.1</td>
<td>101.9</td>
<td>512.7</td>
<td>82.6</td>
</tr>
<tr>
<td>waveform</td>
<td>295.3</td>
<td>38.1</td>
<td>36.2</td>
<td>130.0</td>
</tr>
<tr>
<td>total</td>
<td>61594.3</td>
<td>27584.4</td>
<td>109065.02</td>
<td>2502.57</td>
</tr>
</tbody>
</table>
DDPMine [Cheng et al., ICDE’08]

• Basic idea
  – Integration of branch-and-bound search with FP-growth mining
  – Iteratively eliminate training instance and progressively shrink FP-tree

• Performance
  – Maintain high accuracy
  – Improve mining efficiency
FP-growth Mining with Depth-first Search

\[ \text{sup}(\text{child}) \leq \text{sup}(\text{parent}) \]
\[ \text{sup}(ab) \leq \text{sup}(a) \]
Branch-and-Bound Search

maximize $IG(C|b)$
subject to
$\min sup \leq sup(b) \leq sup(a)$
$0 \leq sup_+(b) \leq sup_+(a)$
$0 \leq sup_-(b) \leq sup_-(a)$

a: constant, a parent node
b: variable, a descendent

Association between information gain and frequency
Training Instance Elimination

Examples covered by feature 1 (1st BB)

Examples covered by feature 2 (2nd BB)

Examples covered by feature 3 (3rd BB)

Training examples
DDPMine Algorithm Pipeline

1. Branch-and-Bound Search

2. Training Instance Elimination

3. Output discriminative patterns

Is Training Set Empty?
Efficiency Analysis: Iteration Number

- \( \text{min\_sup} = \theta_0 \); frequent itemset \( \alpha_i \) at i-th iteration
  since \( |T(\alpha_i)| \geq \theta_0 |D_{i-1}| \)

\[
|D_i| = |D_{i-1}| - |T(\alpha_i)| \leq (1 - \theta_0) |D_{i-1}| \leq ... \leq (1 - \theta_0)^i |D_0| 
\]

- Number of iterations:

\[
n \leq \log \frac{1}{1 - \theta_0} |D_0| 
\]

- If \( \theta_0 = 0.5 \) \( n \leq \log_2 |D_0| \); \( \theta_0 = 0.2 \) \( n \leq \log_{1.25} |D_0| \)
## Accuracy

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Harmony</th>
<th>PatClass</th>
<th>DDPMine</th>
</tr>
</thead>
<tbody>
<tr>
<td>adult</td>
<td>81.90</td>
<td>84.24</td>
<td>84.82</td>
</tr>
<tr>
<td>chess</td>
<td>43.00</td>
<td>91.68</td>
<td>91.85</td>
</tr>
<tr>
<td>crx</td>
<td>82.46</td>
<td>85.06</td>
<td>84.93</td>
</tr>
<tr>
<td>hypo</td>
<td>95.24</td>
<td>99.24</td>
<td>99.24</td>
</tr>
<tr>
<td>mushroom</td>
<td>99.94</td>
<td>99.97</td>
<td>100.00</td>
</tr>
<tr>
<td>sick</td>
<td>93.88</td>
<td>97.49</td>
<td>98.36</td>
</tr>
<tr>
<td>sonar</td>
<td>77.44</td>
<td>90.86</td>
<td>88.74</td>
</tr>
<tr>
<td>waveform</td>
<td>87.28</td>
<td>91.22</td>
<td>91.83</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>82.643</strong></td>
<td><strong>92.470</strong></td>
<td><strong>92.471</strong></td>
</tr>
</tbody>
</table>

*Accuracy Comparison*
Efficiency: Runtime

![Graph showing efficiency comparison between Harmony, DDPMine, and PatClass based on minimum support and running time.]
Branch-and-Bound Search: Runtime

![Graph showing runtime vs minimum support with two lines: one for No Prune and one for Prune. The No Prune line shows a sharp increase in runtime as minimum support increases, while the Prune line remains relatively flat.]
Mining Most Significant Graph with Leap Search [Yan et al., SIGMOD’08]

Given a graph dataset $D$ and an objective function $F(g)$, find a graph pattern $g^*$, s.t.

$$g^* = \arg \max_g F(g).$$

Objective functions

1. **Contrast**: $p/q$,

2. **G-test**: $p \cdot \ln^p_q + (1 - p) \cdot \ln^{1-p}_{1-q}$,

3. **Information Gain**: $H(C) - H(C|X)$

4. **Cosine**

5. **many others.**
Upper-Bound

Idea: derive an upper bound, \( \hat{F}(g) \), s.t., \( \hat{F}(g) \) is monotonic to \( \text{freq}(g) \).

\[
G_t(p, q) = p \cdot \ln\frac{p}{q} + (1 - p) \cdot \ln\frac{1-p}{1-q},
\]

\[
\frac{\partial G_t}{\partial q} = \frac{q - p}{(1 - q)q},
\]

\[
\frac{\partial G_t}{\partial p} = \ln\frac{p(1 - q)}{q(1 - p)}.
\]

Since \( \frac{p(1-q)}{q(1-p)} < 1 \) when \( p < q \), hence,

if \( p > q \), \( \frac{\partial G_t}{\partial p} > 0 \), \( \frac{\partial G_t}{\partial q} < 0 \), \hspace{1cm} (1)

if \( p < q \), \( \frac{\partial G_t}{\partial p} < 0 \), \( \frac{\partial G_t}{\partial q} > 0 \). \hspace{1cm} (2)
Upper-Bound: Anti-Monotonic

\[ \text{if } p > q, \frac{\partial G_t}{\partial p} > 0, \frac{\partial G_t}{\partial q} < 0, \]  
\[ \text{if } p < q, \frac{\partial G_t}{\partial p} < 0, \frac{\partial G_t}{\partial q} > 0. \]  

**Rule of Thumb:**
If the frequency difference of a graph pattern in the positive dataset and the negative dataset increases, the pattern becomes more interesting

\[ F(g) = F(p, q) < \max(F(p, \epsilon), F(\epsilon, q)). \]

Monotonic to \( p \) Monotonic to \( q \)

We can recycle the existing graph mining algorithms to accommodate non-monotonic functions.
Structural Similarity

Structural similarity → Significance similarity
\( g \sim g' \Rightarrow F(g) \sim F(g') \)

Sibling

Size-4 graph

Size-5 graph

Size-6 graph
Leap on g’ subtree if

\[
\frac{2\Delta_+(g, g')}{\sup_+(g) + \sup_+(g')} \leq \sigma
\]
\[
\frac{2\Delta_-(g, g')}{\sup_-(g) + \sup_-(g')} \leq \sigma
\]

σ: leap length, tolerance of structure/frequency dissimilarity

g: a discovered graph

g’: a sibling of g
Frequency Association

Association between pattern’s frequency and objective scores
Start with a high frequency threshold, gradually decrease it
LEAP Algorithm

1. Structural Leap Search with Frequency Threshold

2. Support Descending Mining

F(g*) converges

3. Branch-and-Bound Search with F(g*)
# Branch-and-Bound vs. LEAP

<table>
<thead>
<tr>
<th></th>
<th>Branch-and-Bound</th>
<th>LEAP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pruning base</strong></td>
<td>Parent-child bound (“vertical”) strict pruning</td>
<td>Sibling similarity (“horizontal”) approximate pruning</td>
</tr>
<tr>
<td><strong>Feature Optimality</strong></td>
<td>Guaranteed</td>
<td>Near optimal</td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td>Good</td>
<td>Better</td>
</tr>
</tbody>
</table>
## NCI Anti-Cancer Screen Datasets

<table>
<thead>
<tr>
<th>Name</th>
<th>Assay ID</th>
<th>Size</th>
<th>Tumor Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCF-7</td>
<td>83</td>
<td>27,770</td>
<td>Breast</td>
</tr>
<tr>
<td>MOLT-4</td>
<td>123</td>
<td>39,765</td>
<td>Leukemia</td>
</tr>
<tr>
<td>NCI-H23</td>
<td>1</td>
<td>40,353</td>
<td>Non-Small Cell Lung</td>
</tr>
<tr>
<td>OVCAR-8</td>
<td>109</td>
<td>40,516</td>
<td>Ovarian</td>
</tr>
<tr>
<td>P388</td>
<td>330</td>
<td>41,472</td>
<td>Leukemia</td>
</tr>
<tr>
<td>PC-3</td>
<td>41</td>
<td>27,509</td>
<td>Prostate</td>
</tr>
<tr>
<td>SF-295</td>
<td>47</td>
<td>40,271</td>
<td>Central Nerve System</td>
</tr>
<tr>
<td>SN12C</td>
<td>145</td>
<td>40,004</td>
<td>Renal</td>
</tr>
<tr>
<td>SW-620</td>
<td>81</td>
<td>40,532</td>
<td>Colon</td>
</tr>
<tr>
<td>UACC257</td>
<td>33</td>
<td>39,988</td>
<td>Melanoma</td>
</tr>
<tr>
<td>YEAST</td>
<td>167</td>
<td>79,601</td>
<td>Yeast anti-cancer</td>
</tr>
</tbody>
</table>

**Data Description**
Efficiency Tests

Search Efficiency

Search Quality: G-test
**Mining Quality: Graph Classification**

<table>
<thead>
<tr>
<th>Name</th>
<th>OA Kernel*</th>
<th>LEAP</th>
<th>OA Kernel (6x)</th>
<th>LEAP (6x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCF-7</td>
<td>0.68</td>
<td>0.67</td>
<td>0.75</td>
<td>0.76</td>
</tr>
<tr>
<td>MOLT-4</td>
<td>0.65</td>
<td>0.66</td>
<td>0.69</td>
<td>0.72</td>
</tr>
<tr>
<td>NCI-H23</td>
<td>0.79</td>
<td>0.76</td>
<td>0.77</td>
<td>0.79</td>
</tr>
<tr>
<td>OVCAR-8</td>
<td>0.67</td>
<td>0.72</td>
<td>0.79</td>
<td>0.78</td>
</tr>
<tr>
<td>P388</td>
<td>0.79</td>
<td>0.82</td>
<td>0.81</td>
<td>0.81</td>
</tr>
<tr>
<td>PC-3</td>
<td>0.66</td>
<td>0.69</td>
<td>0.79</td>
<td>0.76</td>
</tr>
<tr>
<td>Average</td>
<td>0.70</td>
<td>0.72</td>
<td>0.75</td>
<td>0.77</td>
</tr>
</tbody>
</table>

**AUC**

*OA Kernel*: Optimal Assignment Kernel  
[Fromlich et al., ICML’05]

**LEAP**: LEAP search

![Runtime vs Data Sets](image)

*OA Kernel* $O(n^2 m^3)$ scalability problem!
Direct Mining via Model-Based Search Tree
[Fan et al., KDD’08]

- Basic flows

Divide-and-Conquer Based Frequent Pattern Mining

Mined Discriminative Patterns

Compact set of highly discriminative patterns

1
2
3
4
5
6
7
...

Dataset

Mine & Select $P: 20\%$

Most discriminative $F$ based on IG

Mine & Select $P: 20\%$

Global Support:

$10\times20\%/10000 = 0.02\%$

Mine & Select $P: 20\%$

Mine & Select $P: 20\%$

Few Data

+ +

- -

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$

$Y$ $N$
Analyses (I)

1. Scalability of pattern enumeration

   - Upper bound:
     \[ O(s^s(1-p)) \]

   - “Scale down” ratio:
     \[ \approx \frac{s^s(1-p)}{s^s} = \frac{1}{s^{sp}} \]

2. Bound on number of returned features

   \[ O(2^{\log_m(s)-1} - 1) < O(s/2 - 1) = O(s) = O(n) \]
Analyses (II)

3. Subspace pattern selection

\[ IG(C|X) = H(C) - H(C|X) \]
\[ = - \sum_{c \in \{0,1\}} P(c) \log P(c) + \sum_{x \in \{0,1\}} P(x) \sum_{c \in \{0,1\}} P(c|x) \log P(c|x) \]

- Original set:
  \[ \frac{P_0}{C_0} \sim \frac{P_1}{C_1} \]
- Subset:
  \[ \frac{P_0}{C_0} \ll \frac{P_1}{C_1} \text{ or vice versa} \]

4. Non-overfitting

5. Optimality under exhaustive search
Experimental Study: Itemset Mining (I)

- **Scalability comparison**

<table>
<thead>
<tr>
<th>Datasets</th>
<th>MbT #Pat</th>
<th>#Pat using MbT sup</th>
<th>Ratio (MbT #Pat / #Pat using MbT sup)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adult</td>
<td>1039.2</td>
<td>252809</td>
<td>0.41%</td>
</tr>
<tr>
<td>Chess</td>
<td>46.8</td>
<td>+∞</td>
<td>~0%</td>
</tr>
<tr>
<td>Hypo</td>
<td>14.8</td>
<td>423439</td>
<td>0.0035%</td>
</tr>
<tr>
<td>Sick</td>
<td>15.4</td>
<td>4818391</td>
<td>0.00032%</td>
</tr>
<tr>
<td>Sonar</td>
<td>7.4</td>
<td>95507</td>
<td>0.00775%</td>
</tr>
</tbody>
</table>
Experimental Study: Itemset Mining (II)

- **Accuracy of mined itemsets**

![Bar chart showing accuracy of mined itemsets for different datasets: Adult, Chess, Hypo, Sick, Sonar.](chart)

- **4 Wins**
- **1 loss**

![Bar chart showing comparison of Log(DT #Pat) and Log(MbT #Pat) for different datasets.](chart)

much smaller number of patterns
Tutorial Outline

- Frequent Pattern Mining
- Classification Overview
- Associative Classification
- Substructure-Based Graph Classification
- Direct Mining of Discriminative Patterns
- Integration with Other Machine Learning Techniques
- Conclusions and Future Directions
Integrated with Other Machine Learning Techniques

• **Boosting**
  – Boosting an associative classifier [Sun, Wang and Wong, TKDE’06]
  – Graph classification with boosting [Kudo, Maeda and Matsumoto, NIPS’04]

• **Sampling and ensemble**
  – Data and feature ensemble for graph classification [Cheng et al., In preparation]
Boosting An Associative Classifier
[Sun, Wang and Wong, TKDE’06]

• Apply AdaBoost to associative classification with low-order rules

• Three weighting strategies for combining classifiers
  – Classifier-based weighting (AdaBoost)
    \[ H(x) = \arg \max_{y_i, i=1\ldots k} \left( \sum_{t=1}^{T} \alpha_t [h_t(x) = y_i] \right) \]
    \[ \alpha = \frac{1}{2} \ln \frac{1 - \varepsilon}{\varepsilon} \]
  – Sample-based weighting (Evaluated to be the best)
    \[ H(x) = \arg \max_{y_i, i=1\ldots k} \left( \sum_{t=1}^{T} r_{t,i} [h_t(x) = y_i] \right) \]
    \[ r_i = \frac{W(Y = y_i / Y \neq y_i | x)}{P(x | Y \neq y_i)} = \log \frac{P(x|Y = y_i)}{P(x|Y \neq y_i)} \]
  – Hybrid weighting
    \[ H(x) = \arg \max_{y_i, i=1\ldots k} \left( \sum_{t=1}^{T} \alpha_t r_{t,i} [h_t(x) = y_i] \right) \]
Graph Classification with Boosting
[Kudo, Maeda and Matsumoto, NIPS’04]

• **Decision stump** \( < t, y > \)
  - If a molecule \( x \) contains \( t \), it is classified as \( y \)
  \[
  h_{<t,y>}(x) = \begin{cases} 
  y & \text{if } t \subseteq x, \\
  -y & \text{otherwise}
  \end{cases}
  \]

• **Gain** 
  \[
  \text{gain}(< t, y >) = \sum_{i=1}^{n} y_i h_{<t,y>}(x_i)
  \]
  - Find a decision stump (subgraph) which maximizes gain

• **Boosting with weight vector** \( d^{(k)} = (d_1^{(k)}, \ldots, d_n^{(k)}) \)
  \[
  \text{gain}(< t, y >) = \sum_{i=1}^{n} y_i d_i^{(k)} h_{<t,y>}(x_i)
  \]
Sampling and Ensemble  
[Cheng et al., In Preparation]

- Many real graph datasets are extremely skewed  
  - Aids antiviral screen data: 1% active samples  
  - NCI anti-cancer data: 5% active samples

- Traditional learning methods tend to be biased towards the majority class and ignore the minority class

- The cost of misclassifying minority examples is usually huge
Sampling

- Repeated samples of the positive class
- Under-samples of the negative class
- Re-balance the data distribution
Balanced Data Ensemble

The error of each classifier is independent, could be reduced through ensemble.

\[ f^E(x) = \frac{1}{k} \sum_{i=1}^{k} f^i(x) \]
ROC Curve

Sampling and ensemble

(a) ROC, NCI11

(b) ROC, NCI81
## ROC50 Comparison

<table>
<thead>
<tr>
<th>Datasets</th>
<th>SE</th>
<th>FS</th>
<th>GF</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCI11</td>
<td>0.5318</td>
<td>0.2630</td>
<td>0.3260</td>
</tr>
<tr>
<td>NCI109</td>
<td>0.6149</td>
<td>0.2380</td>
<td>0.3020</td>
</tr>
<tr>
<td>NCI123</td>
<td>0.6059</td>
<td>0.2400</td>
<td>0.2630</td>
</tr>
<tr>
<td>NCI145</td>
<td>0.5716</td>
<td>0.2650</td>
<td>0.3400</td>
</tr>
<tr>
<td>NCI167</td>
<td>0.5059</td>
<td>0.0540</td>
<td>0.0640</td>
</tr>
<tr>
<td>NCI33</td>
<td>0.5815</td>
<td>0.2510</td>
<td>0.3180</td>
</tr>
<tr>
<td>NCI330</td>
<td>0.4847</td>
<td>0.2420</td>
<td>0.3430</td>
</tr>
<tr>
<td>NCI41</td>
<td>0.5809</td>
<td>0.3000</td>
<td>0.3570</td>
</tr>
<tr>
<td>NCI47</td>
<td>0.6002</td>
<td>0.2430</td>
<td>0.3110</td>
</tr>
<tr>
<td>NCI81</td>
<td>0.5406</td>
<td>0.2390</td>
<td>0.2950</td>
</tr>
<tr>
<td>NCI83</td>
<td>0.6113</td>
<td>0.2670</td>
<td>0.3170</td>
</tr>
<tr>
<td>H1</td>
<td>0.5878</td>
<td>0.2280</td>
<td>0.2680</td>
</tr>
<tr>
<td>H2</td>
<td>0.6086</td>
<td>0.5810</td>
<td>0.6510</td>
</tr>
</tbody>
</table>

**SE:** Sampling + Ensemble  
**FS:** Single model with frequent subgraphs  
**GF:** Single model with graph fragments
Tutorial Outline

- Frequent Pattern Mining
- Classification Overview
- Associative Classification
- Substructure-Based Graph Classification
- Direct Mining of Discriminative Patterns
- Integration with Other Machine Learning Techniques
- Conclusions and Future Directions
Conclusions

• Frequent pattern is a discriminative feature in classifying both structured and unstructured data.

• Direct mining approach can find the most discriminative pattern with significant speedup.

• When integrated with boosting or ensemble, the performance of pattern-based classification can be further enhanced.
Future Directions

• **Mining more complicated patterns**
  – Direct mining top-k significant patterns
  – Mining approximate patterns

• **Integration with other machine learning tasks**
  – Semi-supervised and unsupervised learning
  – Domain adaptive learning

• **Applications: Mining colossal discriminative patterns?**
  – Software bug detection and localization in large programs
  – Outlier detection in large networks
    • Money laundering in wired transfer network
    • Web spam in internet
References (1)

- C. Borgelt, and M.R. Berthold. Mining Molecular Fragments: Finding Relevant Substructures of Molecules, ICDM’02.
- C. Chen, X. Yan, P.S. Yu, J. Han, D. Zhang, and X. Gu, Towards Graph Containment Search and Indexing, VLDB'07.
- H. Cheng, X. Yan, J. Han, and C.-W. Hsu, Discriminative Frequent Pattern Analysis for Effective Classification, ICDE’07.
- H. Cheng, X. Yan, J. Han, and P. S. Yu, Direct Discriminative Pattern Mining for Effective Classification, ICDE’08.
- H. Cheng, W. Fan, X. Yan, J. Gao, J. Han, and P. S. Yu, Classification with Very Large Feature Sets and Skewed Distribution, In Preparation.
- J. Cheng, Y. Ke, W. Ng, and A. Lu. FG-Index: Towards Verification-Free Query Processing on Graph Databases, SIGMOD’07.
References (2)

- G. Cong, K. Tan, A. Tung, and X. Xu. Mining Top-k Covering Rule Groups for Gene Expression Data, SIGMOD’05.
- M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis. Frequent Substructure-based Approaches for Classifying Chemical Compounds, TKDE’05.
- G. Dong and J. Li. Efficient Mining of Emerging Patterns: Discovering Trends and Differences, KDD’99.
- G. Dong, X. Zhang, L. Wong, and J. Li. CAEP: Classification by Aggregating Emerging Patterns, DS’99.
- W. Fan, K. Zhang, H. Cheng, J. Gao, X. Yan, J. Han, P. S. Yu, and O. Verscheure. Direct Mining of Discriminative and Essential Graphical and Itemset Features via Model-based Search Tree, KDD’08.
- J. Han, J. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation, SIGMOD’00.
References (3)

- J. Huan, W. Wang, and J. Prins. *Efficient Mining of Frequent Subgraph in the Presence of Isomorphism*, ICDM’03.
- A. Inokuchi, T. Washio, and H. Motoda. *An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data*, PKDD’00.
- M. Kuramochi and G. Karypis. *Frequent Subgraph Discovery*, ICDM’01.
- W. Li, J. Han, and J. Pei. *CMAR: Accurate and Efficient Classification based on Multiple Class-association Rules*, ICDM’01.
- B. Liu, W. Hsu, and Y. Ma. *Integrating Classification and Association Rule Mining*, KDD’98.
- H. Liu, J. Han, D. Xin, and Z. Shao. *Mining Frequent Patterns on Very High Dimensional Data: A Topdown Row Enumeration Approach*, SDM’06.
- S. Nijssen, and J. Kok. *A Quickstart in Frequent Structure Mining Can Make a Difference*, KDD’04.
- F. Pan, G. Cong, A. Tung, J. Yang, and M. Zaki. *CARPENTER: Finding Closed Patterns in Long Biological Datasets*, KDD’03
References (4)

- F. Pan, A. Tung, G. Cong, and X. Xu. COBBLER: Combining Column, and Row enumeration for Closed Pattern Discovery, SSDBM’04.
- J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M-C. Hsu. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-projected Pattern Growth, ICDE’01.
- Y. Sun, Y. Wang, and A. K. C. Wong. Boosting an Associative Classifier, TKDE’06.
- P-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measure for Association Patterns, KDD’02.
- R. Ting and J. Bailey. Mining Minimal Contrast Subgraph Patterns, SDM’06.
- N. Wale and G. Karypis. Comparison of Descriptor Spaces for Chemical Compound Retrieval and Classification, ICDM’06.
- H. Wang, W. Wang, J. Yang, and P.S. Yu. Clustering by Pattern Similarity in Large Data Sets, SIGMOD’02.
- J. Wang and G. Karypis. HARMONY: Efficiently Mining the Best Rules for Classification, SDM’05.
- X. Yan, H. Cheng, J. Han, and P. S. Yu. Mining Significant Graph Patterns by Scalable Leap Search, SIGMOD’08.
- X. Yan and J. Han. gSpan: Graph-based Substructure Pattern Mining, ICDM’02.
References (5)

- X. Yan, P.S. Yu, and J. Han. Graph Indexing: A Frequent Structure-based Approach, SIGMOD’04.
- X. Yin and J. Han. CPAR: Classification Based on Predictive Association Rules, SDM’03.
- M.J. Zaki. SPADE: An Efficient Algorithm for Mining Frequent Sequences, Machine Learning’01.
- F. Zhu, X. Yan, J. Han, P.S. Yu, and H. Cheng. Mining Colossal Frequent Patterns by Core Pattern Fusion, ICDE’07.
Questions?

hcheng@se.cuhk.edu.hk
http://www.se.cuhk.edu.hk/~hcheng