COURSE PRESENTATION:

Discriminative Frequent Pattern Analysis for Effective Classification

Presenter: Han Liang

Outline

- Motivation
- Introduction
- Academic Background
- Methodologies
- Experimental Study
- Contributions

Motivation

- *Frequent patterns are potentially useful in many classification tasks, such as association rule-based classification, text mining, and protein structure prediction.*
- *Frequent patterns can accurately reflect underlying semantics among items (attribute-value pairs).*

Introduction

- *This paper investigates the connections between the support of a pattern and its information gain (a discriminative measure), and develops a method to set the minimum support in pattern mining. It also proposes a pattern selection algorithm. Finally, the generated frequent patterns can be used for building high quality classifiers.*
- *Experiments on UCI data sets indicate that the frequent pattern-based classification framework can achieve high classification accuracy and good scalability.*

Classification – A Two-Step Process

Step1: Classifier Construction: learning the underlying class probability distributions.

> The set of instances used for building classifiers is called *training* data set.

> The learned classifier can be represented as classification rules, decision trees, or mathematical formulae (e.g. Bayesian rules).

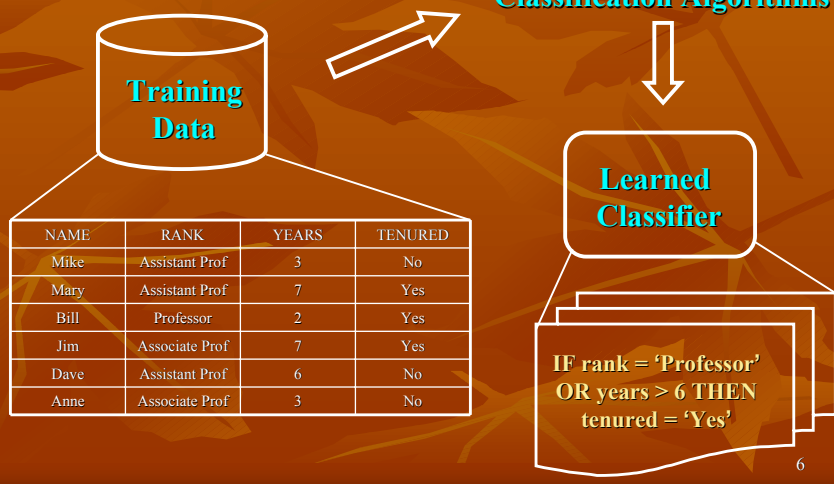
Step2: Classifier Usage: classifying unlabeled instances.

> Estimate accuracy of the classifier.

> Accuracy rate is the percentage of test instances which are correctly classified by the classifier.

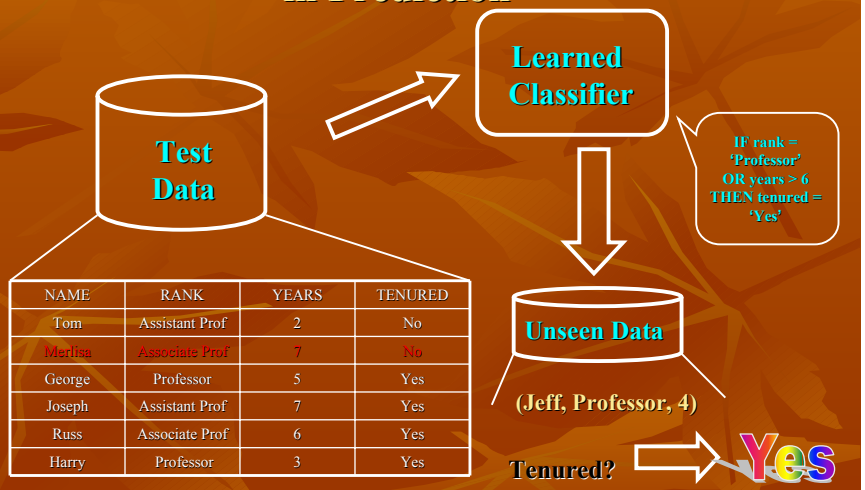
> If the accuracy is acceptable, use the classifier to classify instances whose class labels are not known.

Classification Process I – Classifier Construction



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	No
Mary	Assistant Prof	7	Yes
Bill	Professor	2	Yes
Jim	Associate Prof	7	Yes
Dave	Assistant Prof	6	No
Anne	Associate Prof	3	No

Classification Process II – Use the Classifier in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	No
Merlisa	Associate Prof	7	No
George	Professor	5	Yes
Joseph	Assistant Prof	7	Yes
Russ	Associate Prof	6	Yes
Harry	Professor	3	Yes

Association-Rule based Classification (ARC)

Classification Rule Mining: discovering a small set of classification rules that forms an accurate classifier.

> The data set E is represented by a set of items (or attribute-value pairs) $I = \{a_1, \ldots a_n\}$ and a set of class memberships $C = \{c_1, .. c_m\}$.

> Classification Rule: X => Y, where X is the body and Y is the head.

> X is a set of items (a sub set of I, denoted as $X \subset I$).

> Y is a class membership item.

> Confidence of a classification rule: conf = $S(X \cup Y) / S(X)$.

> Support S(X): the number of training instances that satisfy X.

ARC-II: Mining - Apriori

Generate all the classification rules with support and confidence larger than predefined values.

> Divide training data set into several subsets; one subset for each class membership.

> For each subset, with the help of *on-the-shelf* rule mining algorithms (e.g. Apriori), mines all item sets above the minimum support, and call them *frequent* item sets.

> Output rules by dividing frequent item sets in rule body (attribute-value pairs) and head (one class label).

> Check if the confidence of a rule is above the minimum confidence.

> Merge rules from each sub set, and sort rules according to their confidences.

ARC-III: Rule Pruning

Prune the classification rules with the goal of improving accuracy.

> Simple Strategy:

> Bound the number of rules.

> Pessimistic error-rate based pruning.

> For a rule, if we remove a single item from the rule body and the new rule decreases in error rate, we will prune this rule.

> Data set coverage approach.

> If a rule can classify at least one instance correctly, we will put it into the resulting classifier.

> Delete all covered instances from training data set.

ARC-IV: Classification

Use the resulting classification rules to classify unseen instances.

> Input:

> Pruned, sorted list of classification rules .

> Two different approaches:

> Majority vote

> Use the first rule that is applicable to the unseen instance for classification.

The Framework of Frequent-Pattern based Classification (FPC)

- Discriminative Power vs. Information Gain
- Pattern Generation
- Pattern Selection
- Classifier Construction

Discriminative Power vs. Information Gain - I

The discriminative power of a pattern is evaluated by its information gain.

- Pattern based Information Gain:
  - > Data set S has S_i training instances that belong to class C_i. Thus, S is divided into several subsets, denoted as $S = \{S_1 \ldots S_i \ldots S_m\}$.
  - > Pattern *X* divides ***S*** into two subsets: the group where pattern X is applicable and the group where pattern X is rejected. (binary splitting)
  - > The information gain of pattern X is calculated via:

$$Gain(X) = I(S_1, S_2, \ldots S_m) - E(X)$$

where $I(S_1, S_2, \ldots S_m)$ is represented by:

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s}$$

and *E(X)* is computed by:

$$E(X) = \sum_{j \in \{x, \bar{x}\}} \frac{|S_j|}{|S|} I(S_j)$$

Discriminative Power vs. Information Gain - II

The discriminative power of a pattern is evaluated by its information gain.

- Information Gain is related to pattern support and pattern confidence:
  - > To simplify the analysis, assume pattern $X \in \{0,1\}$ and $C = \{0,1\}$. Let $P(x=1) = \theta$, $P(c=1) = p$ and $P(x=1|c=1) = q$.
  - > Then,

$$E(X) = \sum_{j \in \{x, \bar{x}\}} \frac{|S_j|}{|S|} I(S_j)$$

can be instantiated as:

$$E(X) = -\sum_{x \in \{0,1\}} P(x) \sum_{c \in \{0,1\}} P(c|x) \log P(c|x)$$

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

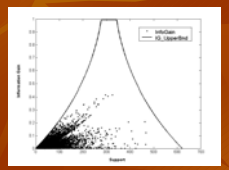
$$+ (\theta(1-q) - (1-p)) \log \frac{(1-p) - \theta(1-q)}{1-\theta}$$

where θ and q are actually the support and confidence of pattern *X*.

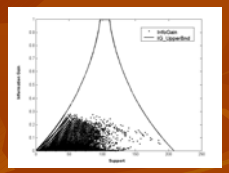
Discriminative Power vs. Information Gain - III

The discriminative power of a pattern is evaluated by its information gain.

- Information Gain is related to pattern support and pattern confidence:
  - > Given a dataset with a fixed class probability distribution, $I(S_1, S_2, \ldots S_m)$ is a constant part.
  - > E(X) is a concave function, it reaches its lower bound w.r.t. q, for fixed p and θ.
  - > After mathematical analysis, we draw the following two conclusions that:
    - > The discriminative power of a low-support pattern is poor. It will harm classification accuracy due to over-fitting.
    - > The discriminative power of a very high-support pattern is also weak. It is useless for improving classification accuracy.
  - > Experiments on UCI datasets.



Austral



Sonar

The X axis represents the support of a pattern and the Y axis represents the information gain. We can clearly see that both low-support and very high-support patterns have small values of information gain.

Pattern Selection Algorithm MMRFS

- Relevance: A relevance measure S is a function mapping a pattern X to a real value such that S(X) is the relevance w.r.t. the class label.
  - > Information gain can be used as a relevance measure.
  - > A pattern can be selected if it is relevant to the class label measured by IG.
- Redundancy: A redundancy measure R is a function mapping two patterns X and Z to a real value such that R (X, Z) is the redundancy value between them.
  - > Mapping function: $R(X,Z) = \frac{P(X,Z)}{P(X)+P(Z)-P(X,Z)} \min(S(X), S(Z))$
  - > A pattern can be chosen if it contains very low redundancy to the patterns already selected.

Pattern Selection Algorithm MMRFS-II

Algorithm 1 Feature Selection Algorithm MMRFS

Input: Frequent patterns $\mathcal{F}$, Coverage threshold δ, Relevance S, Redundancy R
Output: A selected pattern set $\mathcal{F}_s$

1: Let α be the most relevant pattern;
2: $\mathcal{F}_s = \{\alpha\}$;
3: **while** (true)
4: Find a pattern β such that the gain $g(\beta)$ is the maximum among the set of patterns in $\mathcal{F} - \mathcal{F}_s$;
5: If β can correctly cover at least one instance
6: $\mathcal{F}_s = \mathcal{F}_s \cup \{\beta\}$;
7: $\mathcal{F} = \mathcal{F} - \{\beta\}$;
8: If all instances are covered δ times or $\mathcal{F} = \phi$
9: break;
10: **return** $\mathcal{F}_s$

- The MMRFS algorithm searches over the pattern space in a greedy way.
- In the beginning, a pattern with highest relevance value (information gain value) is selected. Then the algorithm incrementally selects more patterns from F.
- A pattern is selected if it has the maximum estimated gain among the remaining patterns F-F_s. The estimated gain is calculated by:

$$g(\alpha) = S(\alpha) - \max_{\beta \in Fs} R(\alpha, \beta)$$

- The coverage parameter δ is set to ensure that each training instance is covered at least δ times by the selected patterns. In this way, the number of patterns selected is automatically determined.

Experimental Study

- Basic learning classifiers – used to classify unseen instances.
  - > C4.5 and SVM
- For each dataset, a set of frequent patterns F is generated.
  - > A basic classifier will be built using all patterns in F. We call it Pat_All.
  - > MMRFS is applied on F and a basic classifier is built using a set of selected features F_s. We call the resulting classifier Pat_FS.
  - > For comparisons, basic classifiers which are built on single features are also tested. Item_All represents the basic classifier which is built on all single features, and Item_FS built on a set of selected single ones.
- Classification Accuracy.
- All experimental results are obtained by use of ten-fold cross validation.

Experimental Study-II

Table 1. Accuracy by SVM on Frequent Combined Features vs. Single Features

Data	Single Feature			Freq. Pattern	
	Item_All	*Item_FS*	*Item_RBF*	*Pat_All*	*Pat_FS*
anneal	**99.78**	**99.78**	99.11	99.33	99.67
austral	85.01	85.50	85.01	81.79	**91.14**
auto	83.25	84.21	78.80	74.97	**90.79**
breast	97.46	97.46	96.98	96.83	**97.78**
cleve	84.81	84.81	85.80	78.55	**95.04**
diabetes	74.41	74.41	74.55	77.73	**78.31**
glass	75.19	75.19	74.78	79.91	**81.32**
heart	84.81	84.81	84.07	82.22	**88.15**
hepatic	84.50	89.04	85.83	81.29	**96.83**
horse	83.70	84.79	82.36	82.35	**92.39**
iono	93.15	94.30	92.61	89.17	**95.44**
iris	94.00	**96.00**	94.00	95.33	**96.00**
labor	89.99	91.67	91.67	94.99	**95.00**
lymph	81.00	81.62	84.29	83.67	**96.67**
pima	74.56	74.56	76.15	76.43	**77.16**
sonar	82.71	86.55	82.71	84.60	**90.86**
vehicle	70.43	72.93	72.14	73.33	**76.34**
wine	98.33	99.44	98.33	98.30	**100**
zoo	97.09	97.09	95.09	94.18	**99.00**

- Table 1 shows the results by SVM.
- Pat_FS has significant improvement over Item_All and Item_FS. This conclusion indicates that:
  - > the discriminative power of some frequent patterns is higher than that of single features.
- The performance of Pat_All is much worse than that of Pat_FS. That confirms that redundant and non-discriminative patterns let classifiers over-fit the data and decrease the classification accuracy.
- The experimental results by C4.5 is similar to SVM's.

Experimental Study-III: Scalability Test

Table 3. Accuracy & Time on Chess Data

min_sup	#Patterns	Time (s)	SVM (%)	C4.5 (%)
1	N/A	N/A	N/A	N/A
2000	68,967	44.703	92.52	97.59
2200	28,358	19.938	91.68	97.84
2500	6,837	2.906	91.68	97.62
2800	1,031	0.469	91.84	97.37
3000	136	0.063	91.90	97.06

Table 4. Accuracy & Time on Waveform Data

min_sup	#Patterns	Time (s)	SVM (%)	C4.5 (%)
1	9,468,109	N/A	N/A	N/A
80	26,576	176.485	92.40	88.35
100	15,316	90.406	92.19	87.29
150	5,408	23.610	91.53	88.80
200	2,481	8.234	91.22	87.32

Table 5. Accuracy & Time on Letter Recognition Data

min_sup	#Patterns	Time (s)	SVM (%)	C4.5 (%)
1	5,147,030	N/A	N/A	N/A
3000	3,246	200.406	79.86	77.08
3500	2,078	103.797	80.21	77.28
4000	1,429	61.047	79.57	77.32
4500	962	35.235	79.51	77.42

- Scalability tests are performed to show the frequent pattern-based framework is very scalable with good classification accuracy.
- Three large UCI data sets are chosen.
- In each table, experiments are conducted by varying min_sup. #Patterns gives the number of frequent patterns. Time gives the sum of pattern mining and pattern selection time.
- min_sup =1 is used to enumerate all feature combinations. Pattern selection fails with such a large number of patterns.
- In contrast, the frequent pattern-based framework is very efficient and achieves good accuracy within a wide range of minimum support thresholds.

Contributions

- This paper propose a framework of frequent pattern-based classification. By analyzing the relations between pattern support and its discriminative power, the paper shows that frequent patterns are very useful for classification.
- Frequent pattern-based classification can use the state-of-the-art frequent pattern mining algorithm for pattern generation, thus achieving good scalability.
- An effective and efficient pattern selection algorithm is proposed to select a set of frequent and discriminative patterns for classification.

Thanks!

Any Question?