Discriminative Frequent Pattern Analysis for Effective Classification*

Hong Cheng† Xifeng Yan‡ Jiawei Han† Chih-Wei Hsu†

†University of Illinois at Urbana-Champaign
‡IBM T. J. Watson Research Center
{hcheng3, hanj, chsu}@cs.uiuc.edu, xifengyan@us.ibm.com

Abstract

The application of frequent patterns in classification appeared in sporadic studies and achieved initial success in the classification of relational data, text documents and graphs. In this paper, we conduct a systematic exploration of frequent pattern-based classification, and provide solid reasons supporting this methodology. It was well known that feature combinations (patterns) could capture more underlying semantics than single features. However, inclusion of infrequent patterns may not significantly improve the accuracy due to their limited predictive power. By building a connection between pattern frequency and discriminative measures such as information gain and Fisher score, we develop a strategy to set minimum support in frequent pattern mining for generating useful patterns. Based on this strategy, coupled with a proposed feature selection algorithm, discriminative frequent patterns can be generated for building high quality classifiers. We demonstrate that the frequent pattern-based classification framework can achieve good scalability and high accuracy in classifying large datasets. Empirical studies indicate that significant improvement in classification accuracy is achieved (up to 12% in UCI datasets) using the so-selected discriminative frequent patterns.

1. Introduction

Frequent pattern mining has been a focused theme in data mining research with a large number of scalable methods proposed for mining various kinds of patterns including itemsets [2, 10, 27], sequences [3, 16, 26] and graphs [11, 22]. Frequent patterns have found broad applications in areas like association rule mining, indexing, and clustering [1, 23, 20]. The application of frequent patterns in classification also achieved some success in the classification of relational data [14, 13, 25, 6, 19], text [15], and graphs [7].

Frequent patterns reflect strong associations between items and carry the underlying semantics of the data. They are potentially useful features for classification. In this paper, we investigate systematically the framework of frequent pattern-based classification, where a classification model is built in the feature space of single features as well as frequent patterns. The idea of frequent pattern-based classification has been exploited by previous studies in different domains, including: (1) associative classification [14, 13, 25, 6, 19], where association rules are generated and analyzed for classification; and (2) graph classification [7], text categorization [15] and protein classification [12], where subgraphs, phrases, or substrings are used as features.

All these related studies demonstrate, to some extent, the usefulness of frequent patterns in classification. Although it is known that frequent patterns are useful, there is a lack of theoretical analysis on their principles in classification. The following critical questions remain unexplored.

- Why are frequent patterns useful for classification? Why do frequent patterns provide a good substitute for the complete pattern set?
- How does frequent pattern-based classification achieve both high scalability and accuracy for the classification of large datasets?
- What is the strategy for setting the minimum support threshold?
- Given a set of frequent patterns, how should we select high quality ones for effective classification?

In this paper, we will systematically answer the above questions.

*The work was supported in part by the U.S. National Science Foundation NSF IIS-05-13678/06-42771 and NSF BDI-05-15813.

Feature combinations are shown to be useful for classification by mapping data to a higher dimensional space. For example, word phrases can improve the accuracy of document classification. Given a categorical dataset D with n features, we can explicitly enumerate all (2^n) feature combinations and use them in classification. However, there are two significant drawbacks for this approach. First, since the number of feature combinations is exponential to the number of single features, in many cases, it is computationally intractable to enumerate them when the number of single features is large (the scalability issue). Second, inclusion of combined features that appear rarely could decrease the classification accuracy due to the "overfitting" issue—features are not representative. The first problem can be partially solved by the kernel tricks which derive a subset of combined features based on parameter tuning. However, the kernel approach requires an intensive search for good parameters to avoid overfitting.

Through analysis, we found that the discriminative power of a low-support feature is bounded by a low value due to its limited coverage in the dataset; hence the contribution of low-support features in classification is limited, which justifies the usage of frequent patterns in classification. Furthermore, existing frequent pattern mining algorithms can facilitate the pattern generation, thus solving the scalability issue in the classification of large datasets.

As to the minimum support (denoted as *min_sup*) threshold setting in frequent pattern mining, a mapping is built between support threshold and discriminative measures such as information gain and Fisher score, so that features filtered by an information gain threshold cannot exceed the corresponding *min_sup* threshold either. This result can be used to set *min_sup* for generating useful patterns.

Since frequent patterns are generated solely based on frequency without considering the predictive power, the use of frequent patterns without feature selection will still result in a huge feature space. This might not only slow down the model learning process, but even worse, the classification accuracy deteriorates (another kind of overfitting issue—features are too many). In this paper, we demonstrate that feature selection is necessary to single out a small set of discriminative frequent patterns, which is essential for high quality classifiers. Coupled with feature selection, frequent pattern-based classification is able to solve the scalability issue and the overfitting issue smoothly and achieve excellent classification accuracy.

In summary, our contributions include

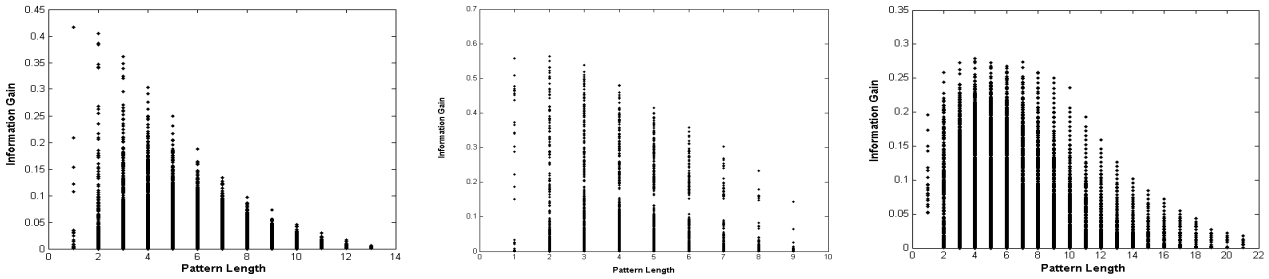
- We propose a framework of frequent pattern-based classification. By analyzing the relationship between pattern frequency and its predictive power, we demonstrate that frequent patterns provide high quality features for classification.
- Frequent pattern-based classification could exploit the state-of-the-art frequent pattern mining algorithms for feature generation, thus achieving much better scalability than the method of enumerating all feature combinations.
- We establish a formal connection between our framework with an information gain-based feature selection approach and show that the *min_sup* threshold is equivalent to an information gain threshold at filtering low quality features. Such an analysis suggests a strategy for setting *min_sup*.
- An effective and efficient feature selection algorithm is proposed to select a set of frequent and discriminative patterns for classification.

The rest of the paper is organized as follows. Section 2 gives the problem formulation. In Section 3, we provide a framework for frequent pattern-based classification. We study the usefulness of frequent patterns, figure out a connection between support and feature filtering measures, discuss the minimum support setting strategy and propose a feature selection algorithm. Extensive experimental results are presented in Section 4, and related work is discussed in Section 5, followed by conclusions in Section 6.

2 Problem Formulation

Assume a dataset has k categorical attributes, where each attribute has a set of values, and m classes $\mathcal{C} = \{c_1, ..., c_m\}$. Each $(attribute, value)$ pair is mapped to a distinct item in $\mathcal{I} = \{o_1, ..., o_d\}$. Assume a pair $(att, val) \rightarrow o_i$, where att is an attribute and val is a value. Let $\boldsymbol{x}$ be the feature vector of a data point s. Then $x_i = 1$ if $att(s) = val$; $x_i = 0$ if $att(s) \neq val$. For numerical attributes, the continuous values are discretized first. Following the mapping, the dataset is represented in $\mathbf{B}^d$ as $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbf{B}^d$ and $y_i \in \mathcal{C}$. $x_{ij} \in \mathbf{B} = \{0, 1\}$, $\forall i \in [1, n], j \in [1, d]$.

Definition 1 (Combined Feature) *A combined feature $\alpha = \{o_{\alpha_1}...o_{\alpha_k}\}$ is a subset of $\mathcal{I}$, where $o_{\alpha_i} \in \{o_1, ..., o_d\}$, $\forall 1 \leq i \leq k$. $o_i \in \mathcal{I}$ is a single feature. Given a dataset $D = \{\boldsymbol{x}_i\}$, the set of data that contains α is denoted as $D_\alpha = \{\boldsymbol{x}_i | x_{i\alpha_j} = 1, \forall o_{\alpha_j} \in \alpha\}$.*



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

Figure 1. Information Gain vs. Pattern Length on UCI data

Definition 2 (Frequent Combined Feature) *For a dataset D, a combined feature α is* frequent *if $\theta = \frac{|D_\alpha|}{|D|} \geq \theta_0$, where $\theta = \frac{|D_\alpha|}{|D|}$ is the relative support of α, and θ_0 is the min_sup threshold, $0 \leq \theta_0 \leq 1$. The set of frequent combined features is denoted as $\mathcal{F}$.*

Given a dataset $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$ and a set of frequent patterns $\mathcal{F}$, D is mapped into a higher dimensional space $\mathbf{B}^{d'}$ with d' features in $\mathcal{I} \cup \mathcal{F}$. The data is denoted as $D' = \{\boldsymbol{x}'_i, y_i\}_{i=1}^n$, where $\boldsymbol{x}'_i \in \mathbf{B}^{d'}$. Notice that $\mathcal{F}$ is parameterized with *min_sup* θ_0.

Frequent Pattern-Based Classification is learning a classification model in the feature space of single features as well as frequent patterns, where frequent patterns are generated w.r.t. *min_sup*.

3 The Framework of Frequent Pattern-based Classification

In this section, we examine the framework of frequent pattern-based classification which includes three steps: (1) feature generation, (2) feature selection, and (3) model learning.

In the feature generation step, frequent patterns are generated with a user-specified *min_sup*. The data is partitioned according to the class label. Frequent patterns are discovered in *each partition* with *min_sup*. The collection of frequent patterns $\mathcal{F}$ is the feature candidates. In the second step, feature selection is applied on $\mathcal{F}$. The set of selected features is $\mathcal{F}_s$. Given $\mathcal{F}_s$, the dataset D is transformed to D' in $\mathbf{B}^{d'}$. The feature space includes all the single features as well as the selected frequent patterns. Finally, a classification model is built on the dataset D'.

3.1 Why Are Frequent Patterns Good Features?

Frequent patterns have two properties: (1) each pattern is a combination of single features, and (2) they are frequent. We will analyze these properties and explain why frequent patterns are useful for classification.

3.1.1 The Usefulness of Combined Features

Frequent pattern is a form of non-linear feature combination over the set of single features. With inclusion of non-linear feature combinations, the expressive power of the new feature space increases. The "Exclusive OR" is an example where the data is linearly separable in $\mathbf{B}^3 = (x, y, xy)$, but not so in the original space $\mathbf{B}^2 = (x, y)$. Non-linear mapping is widely used, *e.g.*, *string kernel* [15, 12] for text or biosequence classification. In frequent pattern-based classification, the single feature vector $\boldsymbol{x}$ is explicitly transformed from the space $\mathbf{B}^d$ where $d = |\mathcal{I}|$ to a larger space $\mathbf{B}^{d'}$ where $d' = |\mathcal{I} \cup \mathcal{F}|$. This will likely increase the chance of including important features.

In addition, the discriminative power of some frequent patterns is higher than that of single features because they capture more underlying semantics of the data. We retrieved three UCI datasets and plotted *information gain* [17] of both single features and frequent patterns in Figure 1. It is clear that some frequent patterns have higher information gain than single features.

3.1.2 Discriminative Power versus Pattern Frequency

In this subsection, we study the relationship between the discriminative power of a feature and its support and demonstrate that the discriminative power of low-support features is limited. In addition, they could harm the classification accuracy due to overfitting.

First, a classification model which uses frequent features for induction has statistical significance, thus generalizes well to the test data. If an infrequent feature is used, the model cannot generalize well to the test data since it is built based on statistically minor observations. This is referred to as overfitting.

Second, the discriminative power of a pattern is closely related to its support. Take information gain as an example. For a pattern α represented by a random variable X, the information gain is

$$IG(C|X) = H(C) - H(C|X) \tag{1}$$

where $H(C)$ is the *entropy* and $H(C|X)$ is the *conditional entropy*. Given a dataset with a fixed class distribution, $H(C)$ is a constant. The upper bound of the information gain, IG_{ub}, is

$$IG_{ub}(C|X) = H(C) - H_{lb}(C|X) \tag{2}$$

where $H_{lb}(C|X)$ is the lower bound of $H(C|X)$. Assume the support of α is θ, we will show in the following that, $IG_{ub}(C|X)$ is closely related to θ. When θ is small, $IG_{ub}(C|X)$ is low. That is, the infrequent features have a very low information gain upper bound.

To simplify the analysis, assume $X \in \{0,1\}$ and $\mathcal{C} = \{0,1\}$. Let $P(x=1) = \theta$, $P(c=1) = p$ and $P(c=1|x=1) = q$. Then

$$\begin{aligned} H(C|X) &= -\sum_{x\in\{0,1\}} P(x) \sum_{c\in\{0,1\}} P(c|x)\log P(c|x) \\ &= -\theta q \log q - \theta(1-q)\log(1-q) \\ &+ (\theta q - p)\log\frac{p-\theta q}{1-\theta} \\ &+ (\theta(1-q)-(1-p))\log\frac{(1-p)-\theta(1-q)}{1-\theta} \end{aligned}$$

$H(C|X)$ is a function of p, q and θ. Given a dataset, p is a fixed value. As $H(C|X)$ is a concave function, it reaches its lower bound w.r.t. q, for fixed p and θ at the following conditions. If $\theta \le p$, $H(C|X)$ reaches its lower bound when $q = 0$ or 1. If $\theta > p$, $H(C|X)$ reaches its lower bound when $q = p/\theta$ or $1-(1-p)/\theta$. The cases of $\theta \le p$ and $\theta \ge p$ are symmetric. Due to space limit, we only discuss the case when $\theta \le p$ and the analysis for the other is similar.

Since $q = 0$ and $q = 1$ are symmetric for the case $\theta \le p$, we only discuss the case $q = 1$. In that case, the lower bound $H_{lb}(C|X)$ is

$$H_{lb}(C|X)_{|q=1} = (\theta-1)\left(\frac{p-\theta}{1-\theta}\log\frac{p-\theta}{1-\theta} + \frac{1-p}{1-\theta}\log\frac{1-p}{1-\theta}\right) \tag{3}$$

The partial derivative of $H_{lb}(C|X)_{|q=1}$ w.r.t. θ is

$$\begin{aligned} \frac{\partial H_{lb}(C|X)_{|q=1}}{\partial\theta} &= \log\frac{p-\theta}{1-\theta} - \frac{p-1}{1-\theta} - \frac{1-p}{1-\theta} \\ &= \log\frac{p-\theta}{1-\theta} \\ &\le \log 1 \\ &\le 0 \end{aligned}$$

The above analysis demonstrates that the information gain upper bound $IG_{ub}(C|X)$ is a function of support θ. $H_{lb}(C|X)_{|q=1}$ is monotonically decreasing with θ, *i.e.*, the smaller θ is, the larger $H_{lb}(C|X)$, and the smaller $IG_{ub}(C|X)$. When θ is small, $IG_{ub}(C|X)$ is small. Therefore, the discriminative power of low-frequency patterns is bounded by a small value. For the symmetric case $\theta \ge p$, a similar conclusion could be drawn: The discriminative power of very high-frequency patterns is bounded by a small value, according to the similar rationale.

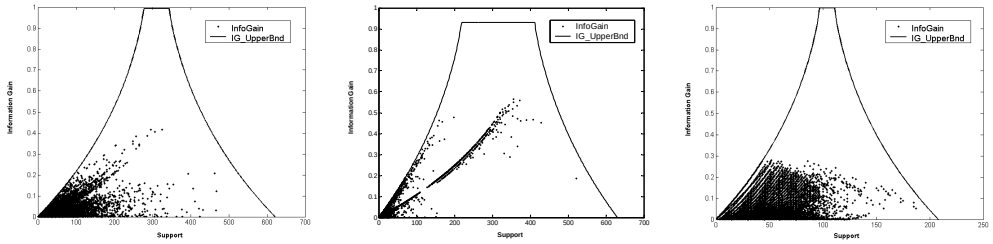
To support the analysis above, we depict empirical results on three UCI datasets in Figure 2. The x axis represents the (absolute) support of a pattern and the y axis represents the information gain. We can clearly see that the information gain of a low-support pattern is bounded by a small value. In addition, for each absolute support, we also plot the theoretical upper bound $IG_{ub}(C|X)_{|q=1}$ if $\theta \le p$ or $IG_{ub}(C|X)_{|q=p/\theta}$ if $\theta > p$, given the fixed $p = P(c=1)$ from the real dataset. We can see that the upper bound of information gain at very low support (and very high support) is small, which confirms our analysis. For example, for a support count of 31 (*i.e.*, $\theta = 5\%$) in Figure 2 (a), the information gain upper bound is as low as 0.06.

Another interesting observation is, at a medium large support (*e.g.*, support = 300 in Figure 2 (a)) where the upper bound reaches the maximum possible value $IG_{ub} = H(C)$, there is a big margin between the information gain of frequent patterns and the upper bound. However, it does not necessarily demonstrate that frequent patterns cannot have very high discriminative power. As a matter of fact, the set of available frequent patterns and their predictive power is closely related to the dataset and the class distribution.

Besides information gain, *Fisher score* [8] is also popularly used to measure the discriminative power of a feature. We analyze the relationship between Fisher score and pattern support. Fisher score is defined as

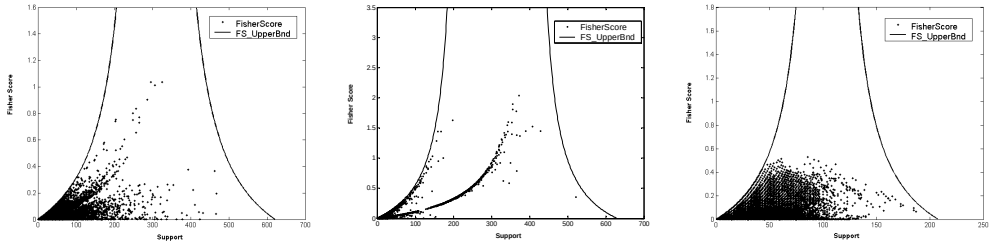
$$Fr = \frac{\sum_{i=1}^{c} n_i(\mu_i - \mu)^2}{\sum_{i=1}^{c} n_i\sigma_i^2} \tag{4}$$

where n_i is the number of data samples in class i, μ_i is the average feature value in class i, σ_i is the standard



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

Figure 2. Information Gain and the Theoretical Upper Bound vs. Support on UCI data



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

Figure 3. Fisher Score and the Theoretical Upper Bound vs. Support on UCI data

deviation of the feature value in class i, and μ is the average feature value in the whole dataset.

We use the notation of p, q and θ as defined before and assume we only have two classes. Assume $\theta \leq p$ (the analysis for $\theta > p$ is symmetric), then Fr is,

$$Fr = \frac{\theta(p-q)^2}{p(1-p)(1-\theta) - \theta(p-q)^2} \tag{5}$$

In Eq. (5), let $Y = p(1-p)(1-\theta)$ and $Z = \theta(p-q)^2$. Then $Y \geq 0$ and $Z \geq 0$. If $Y = 0$, we can verify that $Z = 0$ too. Then Fr is undefined in Eq. (5). In this case, $Fr = 0$ according to Eq. (4). For the case when $Y > 0$ and $Z \geq 0$, Eq. (5) is equivalent to

$$Fr = \frac{Z}{Y - Z}$$

For fixed p and θ, Y is a positive constant. Then Fr monotonically increases with $Z = \theta(p-q)^2$. Assume $p \in (0, 0.5]$ ($p \in [0.5, 1)$ is symmetric), then when $q = 1$, Fr reaches its maximum value w.r.t. q, for fixed p and θ. We denote this maximum value as Fr_{ub}. Put $q = 1$ into Eq. (5), we have

$$Fr_{ub|q=1} = \frac{\theta(1-p)}{p-\theta} \tag{6}$$

According to Eq. (6), as θ increases, $Fr_{ub|q=1}$ increases monotonically, for a fixed p. For $\theta \leq p$, Fisher score upper bound of a low-frequency pattern is smaller than that of a high-frequency one. Note, as θ increases, $Fr_{ub|q=1}$ will have a very large value. When $\theta \to p$, $Fr_{ub|q=1} \to \infty$.

Another interesting evidence to show the relationship between Fr and θ is the sign of $\frac{\partial Fr}{\partial \theta}$. For Eq. (5), the partial derivative of Fr w.r.t. θ is

$$\frac{\partial Fr}{\partial \theta} = \frac{(p-q)^2 p(1-p)}{(p - p^2 - \theta q^2 - \theta p + 2\theta pq)^2} \geq 0 \tag{7}$$

The inequality holds because $p \in [0, 1]$. Therefore, when $\theta \leq p$, Fr monotonically increases with θ, for fixed p and q. The result shows that, Fisher score of a high-frequency feature is larger than that of a low-frequency one, if p and q are fixed.

Figure 3 shows the Fisher score of each pattern vs. its (absolute) support. We also plot the Fisher score upper bound Fr_{ub} w.r.t. support. As mentioned above, for $\theta \leq p$, as θ increases, Fr_{ub} will have very large values. $Fr_{ub} \to \infty$ as θ approaches p. Hence, we only plot a portion of the curve which shows the trend very clearly. The result is similar to Figure 2. These empir-

ical results demonstrate that, features of low support have very limited discriminative power, which is due to their limited coverage in the dataset. Features of very high support have very limited discriminative power too, which is due to their commonness in the data.

3.1.3 The Justification of Frequent Pattern-Based Classification

Based on the above analysis, we will demonstrate that the frequent pattern-based classification is a scalable and effective methodology. The justification is done by building a connection between a well-established information gain-based feature selection approach and our frequent pattern-based method.

Assume the problem context is using combined features for classification. In a commonly used feature selection approach, assume all feature combinations are generated as feature candidates. A subset of high quality features are selected for classification, with an information gain threshold IG_0 (or a Fisher score threshold). According to the analysis in Section 3.1.2, one can always find a *min_sup* threshold θ^*, which satisfies:

$$\theta^* = \arg\max_{\theta} (IG_{ub}(\theta) \leq IG_0) \tag{8}$$

where $IG_{ub}(\theta)$ is the information gain upper bound at support θ. That is, θ^* is the maximum support threshold where the information gain upper bound at this point is no greater than IG_0.

The feature selection approach filters all the combined features whose information gain is less than IG_0; accordingly, in the frequent pattern-based method, features with support $\theta \leq \theta^*$ can be safely skipped because $IG(\theta) \leq IG_{ub}(\theta) \leq IG_{ub}(\theta^*) \leq IG_0$. Compared with the information gain-based approach, it is equivalent to generate the feature with $min\_sup = \theta^*$, then apply feature selection on the frequent patterns only. The latter is our frequent pattern-based approach. Since the number of all the feature combinations is usually very large, the enumeration and feature selection over such a huge feature space is computationally intractable. In contrast, frequent pattern-based method achieves the same result but in a much more efficient way. Obviously it can benefit from the state-of-the-art frequent pattern mining algorithms. The choice of the information gain threshold IG_0 in the first approach corresponds to the setting of the *min_sup* parameter in our framework. If IG_0 is large, the corresponding θ^* is large and vice versa. As it is important to determine the information gain threshold in most feature selection algorithms, the strategy of setting an appropriate *min_sup* is equally crucial. We will discuss this issue in Section 3.2.

3.2 The Minimum Support Effect

Since the set of frequent patterns $\mathcal{F}$ is generated according to *min_sup*, we study the impact of *min_sup* on the classification accuracy and propose a strategy to set *min_sup*.

If *min_sup* is set with a large value, the patterns in $\mathcal{F}$ correspond to very frequent ones. In the context of classification, they may not be the best feature candidates, since they appear in a large portion of the dataset, in different classes. We can clearly observe in Figures 2 and 3 that at a very large *min_sup* value, the theoretical upper bound decreases, due to the "overwhelming" occurrences of the high-support patterns. This is analogous to the *stop word* in text retrieval where those highly frequent words are removed before document retrieval or text categorization.

As *min_sup* lowers down, it is expected that the trend of classification accuracy increases, as more discriminative patterns with medium frequency are discovered. However, as *min_sup* decreases to a very low value, the classification accuracy stops increasing, or even starts dropping due to overfitting. As we analyzed in Section 3.1, features with low support have low discriminative power. They could even harm the classification accuracy if they are included for classification, due to the overfitting effect. In addition, the costs of time and space at both the frequent pattern mining and the feature selection step become very high with a low *min_sup*.

We propose a strategy to set *min_sup*, the major steps of which are outlined below.

- Compute the theoretical information gain (or Fisher score) upper bound as a function of support θ;
- Choose an information gain threshold IG_0 for feature filtering purpose;
- Find $\theta^* = \arg\max_{\theta} (IG_{ub}(\theta) \leq IG_0)$;
- Mine frequent patterns with $min\_sup = \theta^*$.

First, compute the theoretical information gain upper bound as a function of support θ. This only involves with the class distribution p, without generating frequent patterns. Then decide an information gain threshold IG_0 and find the corresponding θ^*. Then for $\theta \leq \theta^*$, $IG_{ub}(\theta) \leq IG_{ub}(\theta^*) \leq IG_0$. In this way, frequent patterns are generated efficiently without missing any feature candidates w.r.t. IG_0. As there are more mature studies on how to set the information gain threshold in feature selection methods [24], we can borrow their strategy and map the selected information gain threshold to a *min_sup* threshold in our method.

3.3 Feature Selection Algorithm MMRFS

Although frequent patterns are shown to be useful for classification, not every frequent pattern is equally useful. It is necessary to perform feature selection to single out a subset of discriminative features and remove non-discriminative ones. In this section, we propose an algorithm MMRFS. The notion is borrowed from the Maximal Marginal Relevance (MMR) [4] heuristic in information retrieval, where a document has high marginal relevance if it is both relevant to the query and contains minimal marginal similarity to previously selected documents. We first define *relevance* and *redundancy* of a frequent pattern in the context of classification.

Definition 3 (Relevance) *A relevance measure S is a function mapping a pattern α to a real value such that $S(\alpha)$ is the relevance w.r.t. the class label.*

Relevance models the discriminative power of a frequent pattern w.r.t. the class label. Measures like information gain and Fisher score can be used as a relevance measure.

Definition 4 (Redundancy) *A redundancy measure R is a function mapping two patterns α and β to a real value such that $R(\alpha, \beta)$ is the redundancy between them.*

Redundancy measures the extent by which two patterns are similar. In this paper, we use a variant of the Jaccard measure [18] to measure the redundancy between different features.

$$R(\alpha, \beta) = \frac{P(\alpha, \beta)}{P(\alpha) + P(\beta) - P(\alpha, \beta)} \times \min(S(\alpha), S(\beta)) \quad (9)$$

According to the redundancy definition, we use the *closed frequent patterns* [27] as features instead of frequent ones in our framework, since for a closed pattern α and its non-closed sub-pattern β, β is completely redundant w.r.t. α.

The MMRFS algorithm searches over the feature space in a heuristic way. A feature is selected if it is relevant to the class label and contains very low redundancy to the features already selected. Initially, the feature with the highest relevance measure is selected. Then the algorithm incrementally selects more patterns from $\mathcal{F}$ with an estimated gain g. A pattern is selected if it has the maximum gain among the remaining patterns. The gain of a pattern α given a set of already selected patterns $\mathcal{F}_s$ is

$$g(\alpha) = S(\alpha) - \max_{\beta \in \mathcal{F}_s} R(\alpha, \beta) \quad (10)$$

Algorithm 1 Feature Selection Algorithm MMRFS

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

```
Let α be the most relevant pattern;
F_s = {α};
while (true)
    Find a pattern β such that the gain g(β) is the maximum among the set of patterns in F − F_s;
    If β can correctly cover at least one instance
        F_s = F_s ∪ {β};
    F = F − {β};
    If all instances are covered δ times or F = φ
        break;
return F_s
```

An interesting question arises: How many frequent patterns should be selected for effective classification? A promising method is to add a database coverage constraint δ, as in [13]. The coverage parameter δ is set to ensure that each training instance is covered at least δ times by the selected features. In this way, the number of features selected is automatically determined, given a user-specified parameter δ. The algorithm is described in Algorithm 1.

4 Experimental Results

In this section, we report a systematic experimental study for the evaluation of our frequent pattern-based classification framework and our proposed feature selection algorithm MMRFS.

A series of datasets from UCI Machine Learning Repository are tested. Continuous attributes are discretized. We use FPClose [9] to generate closed patterns and MMRFS algorithm to do the feature selection. LIBSVM [5] and C4.5 in Weka [21] are chosen as two classification models. Each dataset is partitioned into ten parts evenly. Each time, one part is used for test and the other nine are used for training. We did 10-fold cross validation on each training set and picked the best model for test. The classification accuracies on the ten test datasets are averaged and reported.

4.1 Frequent Pattern-based Classification

We test the performance of the frequent pattern-based classification. For each dataset, a set of frequent patterns $\mathcal{F}$ is generated. A classification model is built using features in $\mathcal{I} \cup \mathcal{F}$, denoted as *Pat_All*. MMRFS is

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 2. Accuracy by C4.5 on Frequent Combined Features vs. Single Features

Dataset	Single Features		Frequent Patterns	
	Item_All	*Item_FS*	*Pat_All*	*Pat_FS*
anneal	98.33	98.33	97.22	**98.44**
austral	84.53	84.53	84.21	**88.24**
auto	71.70	77.63	71.14	**78.77**
breast	95.56	95.56	95.40	**96.35**
cleve	80.87	80.87	80.84	**91.42**
diabetes	**77.02**	**77.02**	76.00	76.58
glass	75.24	75.24	76.62	**79.89**
heart	81.85	81.85	80.00	**86.30**
hepatic	78.79	85.21	80.71	**93.04**
horse	83.71	83.71	84.50	**87.77**
iono	92.30	92.30	92.89	**94.87**
iris	**94.00**	**94.00**	93.33	93.33
labor	86.67	86.67	**95.00**	91.67
lymph	76.95	77.62	74.90	**83.67**
pima	75.86	75.86	76.28	**76.72**
sonar	80.83	81.19	**83.67**	**83.67**
vehicle	70.70	71.49	**74.24**	73.06
wine	95.52	93.82	96.63	**99.44**
zoo	91.18	91.18	95.09	**97.09**

applied on $\mathcal{F}$ and a classifier is built using features in $\mathcal{I} \cup \mathcal{F}_s$, denoted as *Pat_FS*. For comparison, we test the classifiers built on single features, denoted as *Item_All* (using all single features) and *Item_FS* (selected single features), respectively. Table 1 shows the results by SVM and Table 2 shows the results by C4.5. In LIBSVM, all the above four models use linear kernel. In addition, an SVM model is built using RBF kernel on single features, denoted as *Item_RBF*.

From Table 1, it is clear that *Pat_FS* achieves the best classification accuracy in most cases. It has significant improvement over *Item_All* and *Item_FS*. This result is consistent with our theoretical analysis that (1) frequent patterns are useful by mapping the data to a higher dimensional space; and (2) the discriminative power of some frequent patterns is higher than that of single features.

Another interesting observation is that the performance of *Item_RBF* is inferior to that of *Pat_FS*. The reason is, RBF kernel has a different mechanism for feature generation from our approach. In our approach, *min_sup* is used to filter out low-frequency features and MMRFS is applied to select highly discriminative features. On the other hand, the RBF kernel maps the original feature vector to a possibly infinite dimension. The degree (*i.e.*, the maximum length) of combined features depends on the value of γ where γ is the factor in $\mathcal{K}(\mathbf{x}, \mathbf{y}) = e^{-\gamma \|\mathbf{x}-\mathbf{y}\|^2}$, *i.e.*, the degree increases as γ grows. Given a particular γ, the combined features $\mathcal{F}^p$ of length $\leq p$ are used without discriminating their frequency or predictive power, while the combined features of length $> p$ are filtered out.

We also observe that the performance of *Pat_All* is much worse than that of *Pat_FS*, which confirms our reasoning that, redundant and non-discriminative patterns often overfit the model and deteriorate the classification accuracy. In addition, MMRFS is shown to be effective. Generally, any effective feature selection algorithm can be used in our framework. The emphasis is that feature selection is an important step in frequent pattern-based classification.

The above results are also observed in Table 2 for decision tree-based classification.

4.2 Scalability Tests

Scalability tests are performed to show our frequent pattern-based framework is very scalable with good classification accuracy. Three dense datasets, Chess,

Waveform and Letter Recognition[1] from UCI repository are used. On each data, $min\_sup = 1$ is used to enumerate all feature combinations and feature selection is applied over them. In comparison, the frequent pattern-based classification method is tested with variant support threshold settings.

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

In Table 3, we show the result by varying *min_sup* on the Chess data which contains 3,196 instances, 2 classes and 73 items. *#Patterns* gives the number of closed patterns. *Time* gives the sum of pattern mining and feature selection time. We do not include the classification time in the table because our goal is to show that the proposed framework has good scalability in feature generation and selection. The last two columns give the classification accuracy by SVM and C4.5. When $min\_sup = 1$, the enumeration of all the patterns cannot complete in days, thus blocking model construction. Our framework, benefiting from higher support threshold, can accomplish the mining of frequent patterns in seconds and achieve satisfactory classification accuracy.

Tables 4 and 5 show similar results on the other two datasets. When $min\_sup = 1$, millions of patterns are enumerated. Feature selection fails with such a large number of patterns. In contrast, our frequent pattern-based method is very efficient and achieves good accuracy within a wide range of minimum support thresholds.

5 Related Work

The frequent pattern-based classification is related to associative classification. In associative classification, a classifier is built based on high-confidence, high-support association rules [14, 13, 25, 6, 19]. The association between frequent patterns and class labels is used for prediction.

A recent work on top-k rule mining [6] discovers top-k covering rule groups for each row of gene expression profiles. Prediction is then performed based on a classification score which combines the support and confidence measures of the rules.

HARMONY [19] is another rule-based classifier which directly mines classification rules. It uses an instance-centric rule-generation approach and assures for each training instance, that one of the highest-confidence rules covering the instance is included in the rule set. HARMONY is shown to be more efficient and scalable than previous rule-based classifiers. On several datasets that were tested by both our method and HARMONY, our classification accuracy is significantly higher, e.g., the improvement is up to 11.94% on Waveform and 3.40% on Letter Recognition.

Our work is different from associative classification in the following aspects: (1) We use frequent patterns to represent the data in a different feature space, in which any learning algorithm can be used, whereas associative classification builds a classification model using rules only; (2) in associative classification, the prediction process is to find one or several top ranked rule(s) for prediction, whereas in our case, the prediction is made by the classification model; and (3) more importantly, we provide in-depth analysis on why frequent patterns provide a good solution for classification, by studying the relationship between the discriminative power and pattern support. By establishing a connection with an information gain-based feature selection approach, we propose a strategy for setting

[1]The discretized Letter Recognition data is obtained from www.csc.liv.ac.uk/~frans/KDD/Software/LUCS-KDD-DN/DataSets

min_sup as well. In addition, we demonstrate the importance of feature selection on the frequent pattern features and propose a feature selection algorithm.

Other related work includes classification which uses string kernels [15, 12], or word combinations in NLP or structural features in graph classification [7]. In all these studies, frequent patterns are generated and the data is mapped to a higher dimensional feature space. Data which are not linearly separable in the original space become linearly separable in the mapped space.

6 Conclusions

In this paper, we propose a systematic framework for frequent pattern-based classification and give theoretical answers to several critical questions raised by this framework. Our study shows frequent patterns are high quality features and have good model generalization ability. Connected with a commonly used feature selection approach, our method is able to overcome two kinds of overfitting problems and shown to be scalable. A strategy for setting *min_sup* is also suggested. In addition, we propose a feature selection algorithm to select discriminative frequent patterns. Experimental studies demonstrate that significant improvement is achieved in classification accuracy using the frequent pattern-based classification framework.

The framework is also applicable to more complex patterns, including sequences and graphs. In the future, we will conduct research in this direction.

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. of SIGMOD*, pages 207–216, 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of VLDB*, pages 487–499, 1994.

[3] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. of ICDE*, pages 3–14, 1995.

[4] J. Carbonell and J. Coldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proc. of SIGIR*, pages 335–336, 1998.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[6] G. Cong, K. Tan, A. Tung, and X. Xu. Mining top-k covering rule groups for gene expression data. In *Proc. of SIGMOD*, pages 670–681, 2005.

[7] M. Deshpande, M. Kuramochi, and G. Karypis. Frequent sub-structure-based approaches for classifying chemical compounds. In *Proc. of ICDM*, pages 35–42, 2003.

[8] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley Interscience, 2nd edition, 2000.

[9] G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'03)*, 2003.

[10] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proc. of SIGMOD*, pages 1–12, 2000.

[11] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *Proc. of ICDM*, pages 313–320, 2001.

[12] C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for svm protein classification. In *Proc. of PSB*, pages 564–575, 2002.

[13] W. Li, J. Han, and J. Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proc. of ICDM*, pages 369–376, 2001.

[14] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proc. of KDD*, pages 80–86, 1998.

[15] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

[16] 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. In *Proc. of ICDE*, pages 215–226, 2001.

[17] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[18] P. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proc. of KDD*, pages 32–41, 2002.

[19] J. Wang and G. Karypis. HARMONY: Efficiently mining the best rules for classification. In *Proc. of SDM*, pages 205–216, 2005.

[20] K. Wang, C. Xu, and B. Liu. Clustering transactions using large items. In *Proc. of CIKM*, pages 483–490, 1999.

[21] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2nd edition, 2005.

[22] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. In *Proc. of ICDM*, pages 721–724, 2002.

[23] X. Yan, P. S. Yu, and J. Han. Graph Indexing: A frequent structure-based approach. In *Proc. of SIGMOD*, pages 335–346, 2004.

[24] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proc. of ICML*, pages 412–420, 1997.

[25] X. Yin and J. Han. CPAR: Classification based on predictive association rules. In *Proc. of SDM*, pages 331–335, 2003.

[26] M. J. Zaki. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.

[27] M. J. Zaki and C. Hsiao. CHARM: An efficient algorithm for closed itemset mining. In *Proc. of SDM*, pages 457–473, 2002.