What is Unequal among the Equals? Ranking Equivalent Rules from Gene Expression Data

Ruichu Cai, Anthony K. H. Tung, Zhenjie Zhang, Zhifeng Hao

Abstract—In previous studies, association rules have been proven to be useful in classification problems over high dimensional gene expression data. However, due to the nature of such datasets, it is often the case that millions of rules can be derived such that many of them are covered by exactly the same set of training tuples and thus have exactly the same support and confidence. Ranking and selecting useful rules from such equivalent rule groups remain an interesting and unexplored problem.

In this paper, we look at two interestingness measures for ranking the interestingness of rules within equivalent rule group: *Max-Subrule-Conf* and *Min-Subrule-Conf*. Based on these interestingness measures, an incremental Apriori-like algorithm is designed to select more interesting rules from the lower bound rules of the group. Moreover, we present an improved classification model to fully exploit the potential of the selected rules. Our empirical studies on our proposed methods over five gene expression datasets show that our proposals improve both the efficiency and effectiveness of the rule extraction and classifier construction over gene expression datasets.

Index Terms—Association Rules, Gene Expression Data, Incremental Mining Framework, Robust Classification.

1 Introduction

ALL ARE EQUAL, BUT SOME ARE MORE EQUAL THAN OTHERS.

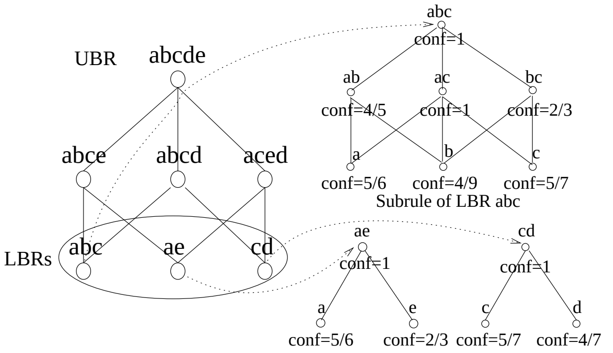
George Orwell, “Animal Farm”

Recent studies [5], [6], [7], [8], [22] have shown that association rules are helpful in the analysis of the gene expression data, especially for the reconstruction of gene expression network and disease classification. As a form of knowledge representation, association rules are also popular among biologists due to their simplicity for interpretation.

However, because of the high dimensionality and limited number of samples of gene expression data, association rules discovered from these datasets tend to suffer from combinatorial explosion. When millions of rules are discovered in the gene expression data, it is important to provide a systematic mechanism to select the most valuable ones. In our previous work [8], [7], a partial solution is provided by grouping set of rules into equivalent class called rule group. Rules within the same rule group are derived from exactly the same set of rows (or patient samples) and as a result have exactly the same support and confidence.

i	r_i	class
1	a, b, c, d, e, g, h, i	c_0
2	a, b, c, d, e, f, h, i	c_0
3	a, b, c, d, e, f, g, i	c_0
4	a, b, c, d, e, f, g, h	c_0
5	a, c, f, g, h, i	c_0
6	a, b, d, f, g, h, i	c_1
7	b, c, f, g, h, i	c_1
8	b, c, e, f, g, h, i	c_1
9	b, d, e, f, g, h, i	c_1
10	b, d, f, g, h, i	c_1

TABLE 1
Example Dataset



Lattice structure of rule group G Subrule of LBR ae Subrule of LBR cd

Fig. 1. Example Rule Group

Example 1: Consider the dataset in Table 1 which has 10 training samples each belonging to one of the two classes c_0 and c_1. Here, the alphabets “a”, ..., “i” represent the genes’ expression states that are either expressed or suppressed. Our

- *Ruichu Cai is with Faculty of Computer Science, Guangdong University of Technology and School of Computer Science and Engineering, South China University of Technology, China. Ruichu’s work was done when he was visiting National University of Singapore.*
  E-mail: cairuichu@gmail.com
- *Anthony K. H. Tung and Zhenjie Zhang are with School of Computing, National University of Singapore, Singapore.*
  E-mail: {atung, zhenjie}@comp.nus.edu.sg
- *Zhifeng Hao is with Faculty of Computer, Guangdong University of Technology, China.*
  E-mail: mazfhao@scut.edu.cn

aim is to find rules that accurately associate a group of gene states with a certain class. From Table 1, we can see that the rule $abcde \rightarrow c_0$ is applicable to row r_1, r_2, ..., r_4 (also refer to as **covered** by r_1, r_2, ..., r_4) and thus has a support of 4 and confidence of 100%. However, there are also many other rules generated from r_1,r_2,..., r_4 (example, $abce \rightarrow c_0$) have the same support and confidence as $abcde \rightarrow c_0$. We refer to this whole set of rules as a rule group and they are shown in the left hand side of Figure 1 organized as a lattice.

Given a rule group, it is proven in [7] that there is an unique upper bound rule (UBR) γ in the rule group such that its antecedent is a superset of the antecedent of all other rules γ' in the rule group. In this paper, we say γ is a super-rule of γ' or γ' is a sub-rule of γ. In Figure 1, $abcde \rightarrow c_0$ is the UBR of the example rule group since it is a super-rule of all other rules within the group.

Presentation wise, it makes sense to present an equivalent set of rules using its UBR, since it is difficult for human to interpret millions of rules that can be discovered from a gene expression dataset. However, since the UBR is the most specific rule in a rule group, it tends to contain large number of items in its antecedent, making it difficult for the biologists to identify the important genes, and also increasing the chance of over-training when the rules are used to construct a rule-based classifier [8], [7].

To overcome this problem, it is proposed in [8], [7] to use a representative set of lower bound rules (LBRs) for the purpose of classification model construction and rules interpretation. The LBRs in a rule group are a set of rules such that none of their sub-rules are in the rule group. In our running example, $abc \rightarrow c_0$, $ae \rightarrow c_0$ and $cd \rightarrow c_0$ are LBRs of the rule group G. Unlike UBRs, the number of LBRs in a rule group can be rather huge due to the high dimensionality of gene expression data and a representative set must be selected from these LBRs for constructing classification model and for users' interpretation. Since LBRs are the most general rules in a rule group, each of them tends to contain fewer genes in its antecedent. Thus, the problem of over-training can be reduced and each rule can be more easily interpreted [1].

To select representative LBRs from a rule group, Cong et al. [7] take the length of the rule into consideration and select the top-k shortest LBRs from each rule group for the construction of a classifier. This is based on the heuristic of Occam's Razor where the shortest rules represent the simplest explanations. However, questions remain on whether such an approach is the most appropriate and whether there are other more effective measures that can rank a set of equivalent rules within a rule groups.

In this paper, we give a positive answer to this question by proposing two interestingness measures that allow us to effectively and efficiently select representative LBRs from a rule group. These interestingness measures are based on two observations of gene expression data:

Observation 1: A set of genes is interesting if it gives much better prediction power than any of its subset.

Observation 1 is consistent with a widely accepted principle in the rule mining community that a rule is interesting only if it provides more prediction power than its sub-rules [14]. Biologically, this also makes sense since it is a well known biological fact that genes tend to perform a certain function in a group and the absence of even one gene will mean that the function is unlikely to be performed. Assuming that some of these functions are important in determining the class of a sample, finding rules that exhibit behavior of such nature will be important. Thus our first proposed interestingness measure called *Max-Subrule-Conf*(*MaxSC*) is geared towards finding rules that have much higher prediction accuracy than their sub-rules. In such a scheme, an LBR will be ranked high if the maximum confidence of its sub-rules is low. We note that *MaxSC* is actually equivalent to the *Minimal-Confidence-Gain* [14].

Observation 2: Gene expression datasets can contain noise and error which can invalidate Observation 1.

Gene expression datasets are well known to be noisy due to various factors like calibration of instruments, image processing, etc. In such a case, an interesting rule as stated in Observation 1 can be corrupted due to errors in the data. For example, in Table 1, if the item "d" is somehow erroneously left out from row r_6, then the confidence of the rule $abd \rightarrow c_0$ will be increased from 80% to 100%. Similarly, if the gene state "f" is somehow detected wrongly and added into row r_1, then a new rule $abcdef \rightarrow c_0$ with 100% confidence will be created. In both cases, rules with high confidence sub-rules will be created which run counter to Observation 1. In view of this, we propose a second interestingness measure *Min-Subrule-Conf*(*MinSC*) that captures this intuition. In a scheme that applies (*MinSC*) as an interestingness measure, an LBR will be ranked high if the minimum confidence of all its sub-rules is high.

While the semantic of the two interestingness measures introduced above is important in their own right, we also address other important issues that come with the new measures. These include (1) efficiency of rule extraction based on the two measures and (2) usefulness of the extracted rules in a classification model. We summarize our contribution towards these various issues as follows:

- We identify the problem of ranking equivalent rules from a rule group. To the best of our knowledge, the problem has so far been evaded by experts in the area.
- Based on our understanding on the characteristic of gene expression datasets, we propose two new rule measures, namely *Max-Subrule-Conf* and *Min-Subrule-Conf*, effective for ranking rules within a rule group.
- We propose a novel algorithm that searches for the top-k rules based on our proposed interestingness measure. Our search algorithm proceeds in an incremental "diagonal" fashion and makes use of various heuristics and pruning methods to ensure efficiency.

[1] In fact, it is not clear whether it is easier to interpret an unique UBR with large number of genes or multiple LBRs, each with a small number of genes. Most of the biologists we work with tend to prefer the later. Using LBRs always gives better classification accuracy though.

- To make full use of the extracted rules, we investigate various rule based classification schemes. Specificially, we propose a new model IRCBT to choose the best classifier among multiple rule-based classifiers for predicting the class of a sample.
- We test our proposals extensively on real life gene expression datasets. The results indicate promising improvement over the existing methods.

2 Related Works

The development of micorarray technology [37] provides the ability to measure the expression levels of tens of thousands of genes in one single experiment. The generated gene expression data helps us understand the mechanism of biological processes, thus the analysis of gene expression data has attracted a lot of attentions from the data mining researchers.

The existing gene expression data analysis algorithms can be partitioned into two categories: unsupervised and supervised. Among the unsupervised methods, clustering [36] is commonly used to find co-regulated genes or similar samples; Bayesian network is one of most popular models for gene regulatory network reconstruction [38]; association rules are also frequently used to find interesting gene expression patterns [9], [22], [8], [7], reconstruct gene regulatory network [39], [40] and discover functional modules [33].

For supervised methods, the common task is predicting the state of gene expression samples, for example, cancer v.s. normal. A lot of classification models are developed, for their huge application potentials. Traditional statistical based methods, machine learning methods and association rule based classifiers are three typical main categorical methods for this task. The statistical based methods [34], [32], [38] usually select the high ranked genes and the relation among genes is not fully explored [9]. Though machine learning methods take the relation between the genes into consideration, most of them are ”black box” and hard to interpret. For example, although SVM [4] achieves very high classification accuracy, it remains hard to interpret the results. Our work belongs to the association rule based classifier, taking the gene relation into consideration but interpretable to the biologists.

Many association rule based gene expression analysis methods have been proposed. [9] is the first work of applying association rules to find some interesting gene expression patterns. CARPENTER [22] is the first row enumeration algorithm to find closed gene expression pattern. FARMER [8] extends the work of CARPENTER by organizing the association rules in rule groups and building classifier using interesting LBRs. By using top-k pruning and a new classification model, RCBT, TOP-K [7] improves the efficiency of the mining procedure and the accuracy of the classifier. In [13], the BST algorithm is proposed. Instead of generating LBRs from the rule groups, BST maintains a list of UBRs for each training row and classifies new record by comparing them to these UBRs. The problem of equivalent rules selection is not addressed there.

Among the association rule based classifiers, TOP-K[7] is most related to our work. Our algorithm is different from TOP-K in the following aspects. First, we propose two new interestingness measures, *MaxSC* and *MinSC*. Second, an incremental LBR mining framework is developed to mine the interesting LBRs. Finally, we also propose improvement strategies for the RCBT and IRCBT is used in our work.

Our work is closely related to those works on the interestingness measure of the association rules. *MCG* [14] is a pruning strategy in the dense dataset and very similar to *MaxSC* proposed in our work. Minimal Description Length is firstly used in [17] to argue that generator is better than the closed patterns. In addition, information gain [5], [6], lift [3], significant [28], [29], entropy ranking, e.g. CPAR[30],CMAR[18], all those interestingness measures are based on the support and confidence. They do not work when all the LBRs have the same support and confidence.

Our incremental LBR mining method is related to the following association rule algorithms, Apriori [2], Depth APRIORI [2], Vertical Mining [31] and GR-Growth [17]. Different from the first three methods, our incremental framework makes full use of the top-k pruning by discovering the interesting LBRs in early stage. GR-Growth explores a fp-growth framework to discover the frequent generators in the low dimension space. Moreover it is costly to evaluate MaxSC and MinSC with them.

3 Preliminaries

A gene expression dataset D consists of n rows and m items. We use $\mathcal{R} = \{r_1, ..., r_n\}$ and $\mathcal{I} = \{I_1, I_2, ..., I_m\}$ to denote the row set and item set respectively. Here, a row r_i represents the ith sample's gene expression profile, and an item I_j is a gene expression state. Each row r_i consists of a subset of items, i.e. $r_i \subseteq \mathcal{I}$. There is also a class label set $C = \{c_1, \ldots, c_l\}$. Each row r_i is attached with one and only one label from C.

Given an itemset $\mathcal{I}' \subseteq \mathcal{I}$, the **row support** of $\mathcal{I}'$, $R(\mathcal{I}') = \{r_i \in R | \mathcal{I}' \subseteq r_i\}$, contains all rows covering the itemset $\mathcal{I}'$. Likewise, the **item support** of a rowset $\mathcal{R}' \subseteq \mathcal{R}$ is the set of itemsets contained by every row in $\mathcal{R}'$, i.e. $I(\mathcal{R}') = \cap_{r_i \in \mathcal{R}'} \{I_j \in \mathcal{I} \mid I_j \subseteq r_i\}$.

An association rule γ, or rule for short, derived from the dataset D, is represented as $A \rightarrow c_o$, where $A \subset \mathcal{I}$ is the antecedent and $c_o \in C$ is the consequent. Given an association rule γ, it can be interpreted as that the appearance of itemset A entails the class label c_o over the row. The support of γ is defined as $|R(A \cup c_o)|$, and its confidence is the ratio $|R(A \cup c_o)|/|R(A)|$. To simplify the notation, let $sup(\gamma)$ and $conf(\gamma)$ denote the support and confidence of the association rule γ respectively.

Definition 1: **Rule Group**
A rule group is a set of association rules $G = \{\gamma_1, \ldots, \gamma_r\}$ with row support $\mathcal{R}'$, iff (1)$\forall \gamma \in G$, $R(\gamma) = \mathcal{R}'$, and (2) $\forall R(\gamma) = \mathcal{R}'$, $\gamma \in G$.

Two special types of rules exists in a rule group: **upper bound rules** and **lower bound rules**:

Notation	Description
D	gene expression dataset
$\mathcal{R}$	row set
$\mathcal{I}$	item set
C	class label set of D
r_i	ith row in $\mathcal{R}$, which is a subset of $\mathcal{I}$
I_j	jth item in $\mathcal{I}$
c_o	oth class label in C
n	number of rows in $\mathcal{R}$
m	number of items in $\mathcal{I}$
$R(\mathcal{I}')$	row support of an item set $\mathcal{I}'$
$I(\mathcal{R}')$	item support of a row set $\mathcal{R}'$
γ	association rule
$sup(\gamma)$	support of rule γ
$conf(\gamma)$	confidence of rule γ
L^i	LBRs generated from $\{I_1, I_2, \cdots, I_i\}$
L_j	LBRs whose antecedent contain j items
L^i_j	$L_j \cap L^i$

TABLE 2
Table of Notations

Definition 2: **Upper Bound Rule (UBR)**
Given a rule group G, a rule $\gamma : A \rightarrow c_o$ is an upper bound rule of G if there is no rule in G, $\gamma' : A' \rightarrow c_o$ that $A' \supsetneq A$.

Definition 3: **Lower Bound Rule (LBR)**
Given a rule group G, a rule $\gamma : A \rightarrow c_o$ is a lower bound rule of G if there is no rule in G, $\gamma' : A' \rightarrow c_o$ that $A' \subsetneq A$.

We summarize all notations in Table 2.

Classifier induction is an important task performed on gene expression data. Towards this end, rule-based classifier has been proposed for gene expression data [8], [7]. While these works mostly emphasize on the efficient mining of UBRs that uniquely represent rule groups, the efficiency and effectiveness of LBRs extraction from the rule groups are almost unexplored. Our paper here aims to address the latter issue and our formal problem definition is as follows:

Given a gene expression dataset D and a set of rule groups that had been extracted from D, we aim to efficiently extract a set of LBRs from these rule groups so as to construct a classifier that can accurately predict the class of new, unseen sample.

4 Lattice Structure based Interesting Measures

4.1 Max-Subrule-Conf

The first measure introduced in this section is *Max-Subrule-Conf*, which intuitively explores the upper bound on the confidence of all sub-rules.

Definition 4: **Max-Subrule-Conf (MaxSC)**
Given an association rule $\gamma : A \rightarrow c_o$, the Max-Subrule-Conf of the rule is defined as: $MaxSC(\gamma) = max_{A' \subsetneq A} conf(A' \rightarrow c_o)$. In cases where $|A| = 1$, $MaxSC(\gamma) = 0$.

An LBR will be ranked **higher** if it has a **lower** *MaxSC*. Intuitively, such ranking helps to reduce the redundancy of the selected rules, for the following reasons. Considering the rule $\gamma : A \rightarrow c_o$, a high $MaxSC(\gamma)$ means that there is a sub-rule $\gamma' : A' \rightarrow c_o$ that has as much prediction capability as γ. This observation implies that the items in $A - A'$ are very likely to be redundant. In addition, if the LBR contains only one item on the antecedent, then it obviously contains less redundant information than other LBRs in the same rule group, thus the *MaxSC* of such LBRs is defined to be 0. Following the principle of minimizing *MaxSC* can ensure that more compact rules will be ranked higher.

MaxSC can be regarded as a special case of *Min-Confidence-Gain*(MinCG), which is proposed in [14]. Let $MinCG(\gamma)$ denote the *MinCG* of the rule γ, it is straightforward to verify that $MinCG(\gamma) = 1 - MaxSC(\gamma)$.

Example 2: Given Figure 1, the rule group with UBR $abcde \rightarrow c_0$ contains three LBRs including $abc \rightarrow c_0$, $ae \rightarrow c_0$ and $cd \rightarrow c_0$. The values of *MaxSC* on the LBRs can be summarized as $MaxSC(abc \rightarrow c_0) = conf(ac \rightarrow c_0) = 1$, $MaxSC(ae \rightarrow c_0) = conf(a \rightarrow c_0) = 5/6$ and $MaxSC(cd \rightarrow c_0) = conf(c \rightarrow c_0) = 5/7$. Since lower *MaxSC* is preferred, the LBRs will be ranked in the order $cd \rightarrow c_0 \prec ae \rightarrow c_0 \prec abc \rightarrow c_0$. Therefore, the rule $cd \rightarrow c_0$ is considered more valuable than $ae \rightarrow c_0$, according to the current measure. Similarly $ae \rightarrow c_0$ is more valuable than $abc \rightarrow c_0$.

An important property of *MaxSC* is **monotonicity**.

Lemma 1: Given an association rule γ, for any sub-rule γ' of γ, $MaxSC(\gamma) \geq MaxSC(\gamma')$ holds.

Proof: Assume that $\gamma : A \rightarrow c_o$ and $\gamma' : A' \rightarrow c_o$ with $A' \subset A$, then $\forall A'' \subset A'$, $A'' \subset A$ holds. Thus, any sub-rule of γ' must also be sub-rule of γ and we have $MaxSC(\gamma) \geq MaxSC(\gamma')$. □

4.2 Min-Subrule-Conf

While *Max-Subrule-Conf* can reduce the redundancy in the top-k rules, we hereby propose another measure, called *Min-Subrule-Conf*, to improve the robustness of the rules.

Definition 5: **Min-Subrule-Conf (MinSC)**
Given an association rule $\gamma : A \rightarrow c_o$, Min-Subrule-Conf of the rule is: $MinSC(\gamma) = min_{A' \subsetneq A} conf(A' \rightarrow c_o)$.

Based on the definition of *Min-Subrule-Conf*, a rule is ranked **higher** if it has a **higher** value of *MinSC*. As mentioned earlier, *MinSC* is designed to handle noisy gene expression data where rules can be corrupted in such a way that both super-rules and sub-rules can have high confidence. Furthermore, adopting *MinSC* ensures that the rules being found are robust in the sense that even if some of the items are missing from the antecedent of the rule, the particular class at the consequent of the rule will still have a high chance of being correct.

The adoption of *MinSC* is also effective in removing rules that are formed by combining some important items with trivial items. Trivial item is the item that has low prediction ability. In our example dataset, the following 5 items b, f, g,h and i are all trivial items since each of them appears in almost all the samples and thus is not useful for class prediction. For high dimensional data, the number of such trivial items is large and they may form high confidence rules despite the fact that they are trivial items.

Example 3: Take the rule group presented in Figure 1 for example, *MinSC* of the lower bounds are listed as follows: $MinSC(abc \rightarrow c_0) = conf(b \rightarrow c_0) = 4/9$, $MinSC(ae \rightarrow c_0) = conf(e \rightarrow c_0) = 2/3$, $MinSC(cd \rightarrow c_0) = conf(d \rightarrow c_0) = 1/2$. Based on the definition of *MinSC*, the preference order of the rules is $ae \rightarrow c_0 \prec cd \rightarrow c_0 \prec abc \rightarrow c_0$.

We note that **monotonicity** also holds for *MinSC*.

Lemma 2: Given an association rule γ, for any sub-rule γ' of γ, $MinSC(\gamma) \leq MinSC(\gamma')$ holds.

5 Incremental Mining of Top-K LBRs

In this section, we will focus on the efficient mining of top-k LBRs from a rule group G, with respect to *MaxSC* or *MinSC*. Our discussion here will be separated into three parts. In Section 5.1, we present an incremental framework for candidate LBR generation. In Section 5.2, we discuss how the two interestingness measures can be used to efficiently reduce the search space of the top-k rules. In Section 5.3, we propose a heuristic item ordering to accelerate the mining procedure. Finally, the complete mining algorithm is summarized in Section 5.4.

Before delving into the details of the mining algorithm, we first present an important property of LBR, which is similar to the Apriori property of frequent pattern in transaction database.

Lemma 3: A rule $\gamma : A \rightarrow c_o$ is an LBR, *iff* $\forall A' \subset A$ $(|A'| = |A| - 1)$, the following two statements hold
(1) $sup(A' \rightarrow c_o) > sup(A \rightarrow c_o)$
(2) $A' \rightarrow c_o$ is an LBR.

Proof: "$\Longrightarrow$": According to the definition of LBR, we have $sup(A \rightarrow c_o) < sup(A' \rightarrow c_o)$. If $A' \rightarrow c_o$ is not an LBR, then there must exist an item I which satisfies $sup(A' - I \rightarrow c_o) = sup(A' \rightarrow c_o)$. Let $A'' = A - I$, then $R(A'' \rightarrow c_0) = R(A \rightarrow c_o)$, which contradicts the fact that $A \rightarrow c_o$ is an LBR. So, $A' \rightarrow c_o$ is an LBR.

"$\Longleftarrow$": If $A \rightarrow c_o$ is not an LBR, there must exist a subset $A'' \subsetneq A$, which satisfies $R(A'' \rightarrow c_o) = R(A \rightarrow c_o)$. Because $A'' \subsetneq A$, there must exists an itemset A' which satisfies $|A'| = |A| - 1$ and $A'' \subseteq A' \subseteq A$. Then the rule $A' \rightarrow c_o$ $(|A'| = |A| - 1)$ satisfies $R(A' \rightarrow c_o) = R(A \rightarrow c_o)$, which contradicts with the condition $sup(A \rightarrow c_o) < sup(A' \rightarrow c_o)$. So, $A \rightarrow c_o$ is an LBR. □

The property intuitively shows that all the sub-rules of an LBR are also LBRs with a larger support. This is similar to the Apriori property used in the frequent pattern mining. As such, Lemma 3 provides an efficient way to verify an LBR by checking the support of its immediate sub-rules only. We can thus generate the LBRs of a rule group by enumerating through the lattice space consisting of its genes, as in the Apriori algorithm[2] except that we only focus on the LBRs instead of the frequent patterns.

Despite of the Apriori property on the LBRs, it remains challenging to discover all LBRs from a rule group, since traditional search strategies, such as breadth-first and depth-first search, are no longer sufficiently efficient. The underlying reason is that frequent patterns span across the whole lattice space on all levels, while LBRs of the target group typically lies at high levels of the lattice. Therefore, a breadth-first search starting at the bottom of the lattice results in large amount of processing on the intermediate levels even though they contain no LBR of the target rule group. This leads to ineffective pruning, since pruning is usually only plausible when sufficient LBRs of the target group have been discovered. On the other hand, depth-first search [31] on the lattice prevents the verification of the LBRs, since the sub-rules must be available during the Apriori pruning approach based on Lemma 3. To overcome these difficulties, we introduce a novel incremental LBR generation framework in the next section.

5.1 Incremental LBR Generation

The new incremental LBR generation framework is a mixture of depth-first and breadth first search in the LBR lattice space. In traditional lattice search strategies, the lattice space is split into levels according to the number of items involved. Consider a UBR $\mathcal{I}' \rightarrow c_o$, without loss of generality, we assume $\mathcal{I}' = \{I_1, I_2, \cdots, I_{|\mathcal{I}'|}\}$. The lattice space covered by this UBR consists of levels $\{L_1, L_2, \ldots, L_{|\mathcal{I}'|}\}$. Each L_j contains sub-rules with exactly j genes. Our new framework, however, partitions the lattice space in a diagonal way. Specifically, given the same UBR of rule group G, we use L^i to denote the set of LBRs, whose antecedents only contain first i genes, i.e. $L^i = \{A \rightarrow c_o \mid A \subseteq \{I_1, I_2, \ldots, I_i\}\}$. Each L^i can be further divided into sub-levels $\{L^i_1, L^i_2, \ldots, L^i_i\}$, with each L^i_j contains rules in L^i with exactly j genes. The new incremental generation algorithm thus iterates from L^i to L^{i+1} until $i = |\mathcal{I}'|$, and utilizes the diagonal partition for effective pruning.

In Algorithm 1, we present an incremental framework to discover all LBRs of a target rule group represented by a unique UBR $\gamma : \mathcal{I}' \rightarrow c_o$. The framework generates L^1 first, which has only one LBR $\gamma : \{I_1\} \rightarrow c_o$. The algorithm then iterates to L^{i+1} on the basis of L^i. On the generation of L^{i+1}, a traditional breadth-first strategy is adopted, by constructing from L^{i+1}_1 to L^{i+1}_{i+1} in order.

To construct L^{i+1}_1, all LBRs in L^i_1 can be directly included with a new LBR $\gamma : \{I_{i+1}\} \rightarrow c_o$. Note that the new rule γ must be an LBR by the definition. Based on the LBRs in L^{i+1}_1, higher levels L^{i+1}_j $(1 < j \leq i + 1)$ can be generated recursively by the following formula,

$$L^{i+1}_{j+1} = L^i_{j+1} \cup \{\gamma \mid \gamma : A \cup I_{i+1} \rightarrow c_o, \gamma\ is\ an\ LBR\} \quad (1)$$

where $A \rightarrow c_o \in L^i_j$.

The correctness of the Equation 1 is the key to the success of the incremental LBR generation framework, which is ensured by Lemma 4.

Lemma 4: $L^{i+1}_{j+1} = L^i_{j+1} \cup \{\gamma | \gamma : A \cup I_{i+1} \rightarrow c_o, \gamma\ is\ an\ LBR\}$, where $A \rightarrow c_o \in L^i_j$.

Algorithm 1 Incremental LBR Generation Framework

```
Input: $D$: data set; $U,\mathcal{I}' \to c_o$: upper bound rule;
Output: $LS$: LBR set in rule group of $U$
Set $LS = \phi$,$L^0 = \phi$;
for each $i$ from 1 to $|\mathcal{I}'|$ do
    Set $j = 0$ and $L^i_0 = \{\phi \to c_o\}$;
    while $L^i_j \neq \phi$ do
        $L^i_{j+1}$=UpdateLevel($L^{i-1}_j$,$I_i$, $U$,$S$);
        $j = j + 1$;
Return $LS$;
function: UpdateLevel($L^{i-1}_j$,$I_i$,$U$,$S$);
$L^i_{j+1} = L^{i-1}_{j+1}$;
for each $A \to c_o \in L^{i-1}_j$ do
    Generate rule $\gamma : A \cup I_i \to c_o$;
    if $\gamma$ is an LBR according to Lemma 3 then
        if $sup(\gamma) = sup(U)$ then
            Insert $\gamma$ into $LS$;
        else
            Insert $\gamma$ into $L^i_{j+1}$;
Return $L^i_{j+1}$;
```

Proof: : First, it's obvious that $L^i_{j+1} \subset L^{i+1}_{j+1}$ and $\{\gamma|\gamma : A \cup I_{i+1} \to c_o, \gamma\ is\ an\ LBR\} \subset L^{i+1}_{j+1}$. So $L^i_{j+1} \cup \{\gamma|\gamma : A \cup I_{i+1} \to c_o, \gamma\ is\ an\ LBR\} \subset L^{i+1}_{j+1}$ holds.

Moreover, let $\gamma : A_1 \to c_o$ denote an LBR in L^{i+1}_{j+1}. We will show $\gamma \in L^i_{j+1} \cup \{\gamma|\gamma : A \cup I_{i+1} \to c_o, \gamma\ is\ an\ LBR\}$. If $I_{i+1} \notin A_1$, according to the definition of L^i_{j+1}, we have $\gamma \in L^i_{j+1}$. Otherwise, let $A = A_1 - I_{i+1}$, then $A \to c_o$ is a sub-rule of γ. According to Lemma 3, $A \to c_o$ must be an LBR and $A \to c_o \in L^i_j$ holds. So $\forall \gamma \in L^{i+1}_{j+1}$, $\gamma \in L^i_{j+1} \cup \{\gamma|\gamma : A \cup I_{i+1} \to c_o, \gamma\ is\ an\ LBR\}$ holds.

So, we have $L^{i+1}_{j+1} = L^i_{j+1} \cup \{\gamma|\gamma : A \cup I_{i+1} \to c_o, \gamma\ is\ an\ LBR\}$, which finishes the proof. □

The above lemma shows that L^{i+1}_{j+1} consists of two parts, L^i_{j+1} and $\{\gamma|\gamma : A \cup I_{i+1} \to c_o, \gamma\ is\ an\ LBR\}$. The first part consists of the LBRs that are of length $j + 1$ generated from the first i items. The second part is the newly generated LBRs by taking the new items I_{j+1} into consideration. In the incremental framework, we only need to focus on the newly generated LBRs from the second part.

The new candidate $\gamma : A \to c_o$ needs to be confirmed as a new LBR. The level-wise structure of the LBRs provides an efficient way to determine whether a candidate is an LBR by checking the conditions as Lemma 3. With the help of a hash table of L_j, the testing can be done in constant time. If γ' is an LBR of the target rule group G, it is unnecessary to further iterate from γ', since any super-rule of γ' cannot be an LBR of G any more.

Figure 2 gives a running example illustrating the incremental mining of LBRs from the rule group presented with UBR $abcde \to c_o$. The LBR levels $\{L^1, L^2, \dots, L^5\}$ are generated in order from Figure 2(a) to Figure 2(e), to find the LBRs of the target rule group. To facilitate a simpler presentation without ambiguity, we use the antecedents to denote the rules in the following descriptions.

Figure 2(a) shows initialization of the level L^1, which consists of only one rule, i.e. $L^1 = \{a\}$, by definition. In Figure 2(b), L^2 is generated on the basis of L^1. First, for $j = 1$, a new LBR b is generated and added to L^2_1. When $j = 2$, the new candidate ab is generated by adding the new item b to the rule a. The algorithm verifies the validity of ab as an LBR through Lemma 3, by comparing its support with those of a and b. The rule ab is thus inserted into L^2_2. In Figure 2(b), two generated nodes, b and ab, are marked in the shaded areas in the figures. Figure 2(c) shows the construction of L^3. Similarly, when $j = 1$, c is automatically generated and added to L^3_1; for $j = 2$, the rule ac (resp. bc) is extended from rule a (resp. b) by adding another item c. For $j = 3$, the valid LBR abc is constructed by adding c to the ab. Again, the new nodes for L^3 are marked in the shaded area of Figure 2(c). Since the support of the rule abc is exactly the same as that of $abcde$, the rule abc is an LBR of the target group and thus marked in dark color. By continuing the algorithm to L^4 and L^5, another two LBRs of the target group $\{cd, ae\}$ are discovered in Figure 2(d) and Figure 2(e) respectively.

Note that this algorithm generates all LBRs without performing any ranking. A naive implementation of top-k rule selection is to rank the LBRs of the target group after the iteration process discovers all of them. In the next section, we discuss some pruning strategies to improve the efficiency of top-k LBR mining.

5.2 Pruning With Interestingness Measures

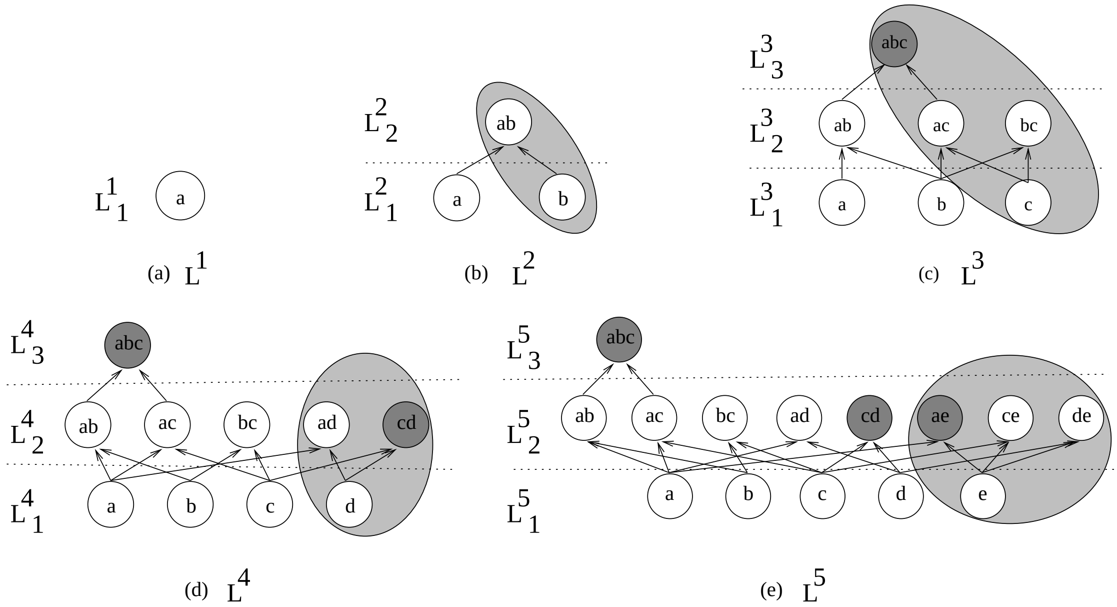
In Algorithm 1, all LBRs of a specified rule group are generated. When only the top-k LBRs with respect to the interestingness measures are required, most of the CPU cycles of Algorithm 1 are wasted on the computation of useless LBRs. In this subsection, we present some pruning strategies in the incremental generation framework, generating top-k LBRs of the target group without iterating all LBRs. Generally speaking, an intermediate rule discovered in Algorithm 1 is useful only when the extensions from it can possibly lead to some rules that are ranked high with respect to *MaxSC* or (*MinSC*).

Assume we have already discovered k LBRs of the rule group, denoted as LS. If all super-rule γ' generated from γ is no more interesting than the current LBRs in LS, then γ is not interesting anymore and can be pruned safely. This motivates our pruning strategies introduced below.

5.2.1 *Pruning With Max-Subrule-Conf*

For *MaxSC*, we define the interestingness threshold as below: $\theta = \max_{\gamma \in LS} MaxSC(\gamma)$. Given an LBR γ, if all super-rules γ' generated from γ satisfy $MaxSC(\gamma') \geq \theta$, γ can be pruned. Thus, we are interested in the **lower bound** of $MaxSC(\gamma')$, in the pruning strategy.

Consider the following sub-rules of γ', $A' - \mathcal{I} \to c_o$, where $\mathcal{I} \in A$. According to the definition of the confidence, we have



(a) L^1

(b) L^2

(c) L^3

(d) L^4

(e) L^5

Fig. 2. A Running Example of the Incremental LBR Mining Method

$conf(A' - \mathcal{I} \rightarrow c_o) = |R((A' - \mathcal{I}) \cup c_o)|/|R(A' - \mathcal{I})|$. Let $x_1 = |R((A' - \mathcal{I}) \cup c_o) - R(A' \cup c_o)|$ and $x_2 = |R(A' - \mathcal{I}) - R(A') - R(c_o)|$, we have,

$$conf(A' - \mathcal{I} \rightarrow c_o) = \frac{|R(A' \cup c_o)| + x_1}{|R(A')| + x_1 + x_2} \geq \frac{|R(A' \cup c_o)|}{|R(A')| + x_2}$$

Because $\mathcal{I} \subset A$ and $A \subset A'$, $x_2 \leq |R(A - \mathcal{I}) - R(A) - R(c_o)|$ holds. So we have

$$\begin{aligned} MaxSC(\gamma) &\geq conf(A' - \mathcal{I} \rightarrow c_o) \\ &\geq \frac{|R(A' \cup c_o)|}{|R(A')| + |R(A - \mathcal{I}) - R(A) - R(c_o)|} \end{aligned} \quad (2)$$

Moreover, due to the monotonicity of *MaxSC*, we have

$$MaxSC(\gamma') \geq MaxSC(\gamma) \quad (3)$$

Combining Equation 2 and Equation 3, we obtain a lower bound of MaxSC of the rule γ' as below:

Lemma 5: Given an LBR $\gamma : A \rightarrow c_o$ and an LBR of target group G generated from γ, $\gamma' : A' \rightarrow c_o$, the lower bound of $MaxSC(\gamma')$ is as below.

$$\max\{\max_{\mathcal{I} \in A} \frac{|R(A' \cup c_o)|}{|R(A')| + |R(A – \mathcal{I}) - R(A) - R(c_o)|}, MaxSC(\gamma)\} \quad (4)$$

Intuitively, the lower bound of $MaxSC(\gamma')$ is determined by two portions: the first portion is the lower bound on the confidence of $A' - I \rightarrow c_o$ and the second portion is the *MaxSC* of the current sub-rule being processed. The above lemma shows that the *MaxSC* of all the LBRs in a target rule group generated from the rule is bounded by Equation 4 and we can use this bound to prune off LBRs in the early stage if we can evaluate it in an efficient way.

For any two itemsets $\mathcal{I}_1 \subset \mathcal{I}_2 \subset A$, we have $R(A – \mathcal{I}_1) \subset R(A – \mathcal{I}_2)$, which means that the lower bound generated from the $\mathcal{I}_1$ is tighter than that of $\mathcal{I}_2$. Thus, we only need to focus on the 1-item itemset $\mathcal{I} \subset A$. When $|\mathcal{I}| = 1$, $|R(A - \mathcal{I})|$ can be efficiently obtained by looking up the corresponding candidate in $L_{|A|-1}$. Moreover, $|R(A')|$ and $|R(A' \cup c_o)|$ are constants according to the definition of rule group. So, the lower bound of $MaxSC(\gamma')$ can be efficiently estimated during our level-wise LBR generation.

To evaluate the second portion of the lower bound, $MaxSC(\gamma)$ needs to be evaluated. In the level-wise enumeration of LBRs, this can be efficiently evaluated as in Equation 5, as the maximal over the confidence of all immediate sub-rules and their *MaxSC*.

$$MaxSC(\gamma) = \max_{\gamma'}\{\max\{conf(\gamma'), MaxSC(\gamma')\}\} \quad (5)$$

where γ' is $A' \rightarrow c_o$, $A' \subset A$ and $|A'| = |A| - 1$.

5.2.2 Pruning With Min-Subrule-Conf

The interestingness threshold for *MinSC* is defined as below: $\theta = \min_{\gamma \in LS} MinSC(\gamma)$. Given a candidate LBR γ, if it is guaranteed that all super-rule γ' generated from γ satisfy $MinSC(\gamma') \leq \theta$, then the LBR γ can be pruned since we aim to find the top-k LBRs with the highest *MinSC*. Thus, we are interested in the **upper bound** of $MinSC(\gamma')$.

For *MinSC*, a rule often has a minimal confidence on the one-item sub-rules. As such, we pay more attention to the confidence of the single item for bounding $MinSC(\gamma')$. Let $\mathcal{R}' = R(\gamma) - R(\gamma') = \{r_1, r_2, \cdots r_k\}$, any row $r \in R'$ must contain one of the following items $I(r) = \{I | I \notin r\}$, and $MinSC(\gamma')$ is bounded by the minimal confidence of those items, $MinSC(\gamma') \leq \max_{I \in I'(r)} conf(I)$. Since the above inequality holds for all the rows $r \in R'$, so we have a tighter upper bound on $MinSC(\gamma')$ as follows:

$$MinSC(\gamma') \leq \min_{r \in R'} \max_{I \in I(r)} conf(I) \tag{6}$$

Moreover, due to the monotonicity of *MinSC*, we have

$$MinSC(\gamma') \leq MinSC(\gamma) \tag{7}$$

Combining Equation 6 and Equation 7, we have the upper bound of $MinSC(\gamma')$ as follows:

Lemma 6: Given an LBR $\gamma : A \rightarrow c_o$ and an LBR of the target rule group generated from this LBR, $\gamma' : A' \rightarrow c_o$, the upper bound of $MinSC(\gamma')$ is:

$$\min\{\min_{r \in R'} \max_{I \in I(r)} conf(I), MinSC(\gamma)\} \tag{8}$$

where $R' = R(\gamma) - R(\gamma')$ and $I(r) = \{I | I \notin r\}$.

The first part shows that the upper bound of *MinSC* is determined by some single items in the rule. The second part shows that the LBRs of the target rule group are bounded by the *MinSC* of its sub-rules.

We next discuss how Equation 8 can be evaluated efficiently. The value of $\max_{I \in I(r)} conf(I)$ can be calculated in a preprocessing procedure and thus the first part of the bound can be estimated very efficiently especially for gene expression data which have small number of rows.

For the second part, *MinSC* of the candidate can be efficiently updated in the level-wise structure, similar to that of *MaxSC*.

$$MinSC(\gamma) = \min_{\gamma'}\{\min\{conf(\gamma'), MinSC(\gamma')\}\} \tag{9}$$

where γ' is $A' \rightarrow c_o$, $A' \subset A$ and $|A'| = |A| - 1$.

5.3 Heuristic Item Ordering

In the above incremental LBR mining framework, the efficiency of the algorithm greatly depends on the order of items: a proper ordering of the items ensures that high ranked LBRs are found in the early stage and makes the process more efficient through better pruning. Here, a heuristic item ranking method is developed by exploring the importance and the interestingness of the items.

Our heuristic item ordering is inspired by the following observations: given a rule group G with upper bound $\mathcal{I}' \rightarrow c_o$, if $\exists r \in R - R(\mathcal{I}')$ (R is the universal row set) which satisfies $r \cap \mathcal{I}' = \mathcal{I}' - \{I\}$, then this item I must be found in each LBR of the rule group G. Because $\forall S \subset \mathcal{I}' - \{I\}$, $R(S) \supset R(A) \cup r$ and $S \rightarrow c_o$ can't be LBR. Considering a generalized case, if $r \cap \mathcal{I}' = \mathcal{I}' - \{I_1, I_2, \cdots, I_k\}$, the LBR of G must contain one of these k items, and we can define the importance of an item based on these observations.

Definition 6: **Item's replaceability and importance** Given a rule group G with UBR $\mathcal{I}' \rightarrow c_o$, the replaceability of an item, $I \in \mathcal{I}'$, is defined as $REP(I) = \min\{|\mathcal{I}' - r \cap \mathcal{I}'| \mid I \in r, r \in R - R(\mathcal{I}')\}$, and the importance is the reciprocal of the replaceability $IMP(I) = \frac{1}{REP(I)}$.

Intuitively, an item with low replaceability is more important for that is contained in the LBRs of the target rule group with high probability. Thus, the items are heuristically ranked in descending order according to the importance. In addition, if two items have the same importance, the item with higher confidence is given higher priority.

5.4 Algorithm

By integrating the top-k pruning strategies and the heuristic item ordering with the incremental LBR generation framework, we give the complete algorithm in Algorithm 2.

The algorithm takes three input parameters: D is the discretized gene expression dataset; U is the UBR of rule group G; k is the number of the LBRs need to be discovered. First, the items are heuristically ordered according to their importance; then algorithm incrementally generates the L^i for the first i items of the UBR by calling subfunction *UpdateLevel*.

During the generation of L^i_j, the subfunction *CheckCandidate* is called to check the state of new candidate: the interesting LBRs are added to L^i_j; LBRs that is more interesting that previously found LBRs is added to the lower bound set LS and the interestingness threshold is updated accordingly.

6 Classification Schemes

We will describe two classifier induction methods: RCBT [7] and IRCBT. RCBT is the state of art rule based classifier induction method, which has been proposed in previous works [7]. IRCBT is an improved version of RCBT by reducing the risk of using trivial classifier to classify the new coming samples, and we will give more details.

6.1 RCBT

RCBT is a rule-based classification model, first proposed in [7]. By using a set of LBRs to make collective decision and building standby classifiers, RCBT reduces the chance that a sample is classified by the default classifier.

RCBT has l classifiers $CL_1, CL_2, ..., CL_l$. The classifier CL_j is built from the rule group set RG_j, where RG_j is the union set of the jth covering rule group for each sample. For each rule group, RCBT finds the k shortest LBRs, those lower bounds will make collective decision to form CL_j.

For a test sample t, the classifiers are run in a certain order to predict the class of t. Given a classifier CL_j, it evaluates the score of the sample with respect to each class, and the class with highest score will be the prediction result for t. The scoring function of the sample for label c_o is as below:

$$Score(t \in c_o) = \frac{\sum_{\gamma \in \Gamma(c_o, r)} conf(\gamma) sup(\gamma)}{\sum_{\gamma \in \Gamma(c_o)} conf(\gamma) sup(\gamma)} \tag{10}$$

where $\Gamma(c_o)$ is the rule set with consequent c_o, and $\Gamma(c_o, t)$ is a subset of $\Gamma(c_o)$ which are covered by the sample t.

If any class label claims better score than any other labels, this label is returned as final result and the prediction process is terminated. Otherwise, next classifier C_{j+1} will be invoked. If no deterministic result is available after CL_l is consumed, the sample is assigned as the default label which is the major class of the training samples.

Algorithm 2 Incremental Top-k LBRs Mining Algorithm

```
Input: $D$: data set; $U$,$\mathcal{I}' \rightarrow c_o$: upper bound rule;
Output: $LS$: LBR set in rule group of $U$
Heuristic item ordering ;
Set $LS = \phi$,$L^0 = \phi$;
Initialize Interestingness Threshold $\theta$;
for each $i$ from 1 to $\mathcal{I}'$ do
    Set $j = 0$ and $L^i_0 = \{\phi \rightarrow c_o\}$;
    while $L^i_j \neq \phi$ do
        $L^i_{j+1}$=UpdateLevel($L^{i-1}_j$,$I_i$, $U$,$S$);
        $j = j + 1$;
Return: $LS$;
function: UpdateLevel($L^{i-1}_j$,$I_i$,$U$,$LS$);
$L^i_{j+1} = L^{i-1}_{j+1}$;
for each $A' \rightarrow c_o \in L^{i-1}_j$ do
    Generate rule $\gamma : A' \cup I_i \rightarrow c_o$;
    if CheckCandidate($\gamma$,$L_{j-1}$,$\theta$,$U$) then
        if $R(\gamma) = R(U)$ then
            Update $LS$ with $\gamma$;
            Update $\theta$;
        else
            Insert $\gamma$ into $L^i_{j+1}$;
Return: $L^i_{j+1}$;
function: CheckCandidate($A \rightarrow c_o$, $L_{j-1}$, $\theta$, $U$);
for each $A' \subset A$ ($|A'| = |A| - 1$) do
    if $A' \rightarrow c_o \notin L_{j-1}$ then
        Return: FALSE;
    if $sup(A' \rightarrow c_o) = sup(A \rightarrow c_o)$ then
        Return: FALSE;
    if $A' \rightarrow c_o$ is prunable with $\theta$ then
        Return: FALSE;
Estimate lower bound of $MaxSC(A \rightarrow c_o)$;
/*or Estimate lower bound of $MinSC(A \rightarrow c_o)$)*/
if Prunable with interestingness bound then
    Return: FALSE;
else
    Return: TRUE;
```

6.2 Improved RCBT

Although RCBT shows promising results in practice, it always employs the first successful classifier to classify the sample, possibly wasting the information contained in other classifiers. Furthermore, although the terminating classifier is able to distinguish between the best label from other labels, the confidence of the classifier may not be sufficiently high. More often than not, we observe that classifiers that are ranked lower can often predict the classification result with higher confidence.

Example 4: Table 3 shows the score for a test sample on 5 classifiers of the DLBCL [25] dataset. When using RCBT to classify the sample, the algorithm stops at CL_1 and it is classified as class c_1. But we can see that the score difference between c_0 and c_1 is trivial. Considering this test sample in all five classifiers, it's more reasonable to classify it as class c_0 based on CL_2, CL_3 and CL_4 since the sample has much higher score on class c_0 than c_1.

TABLE 3
A test sample's score on 5 classifiers

Classifier	CL_1	CL_2	CL_3	CL_4	CL_5
c_0	0.19	0.68	0.8	0.51	0.45
c_1	0.21	0.53	0.31	0.42	0.53

The example shows that we should avoid using classifiers with trivial score difference to classify the test sample, for the result of such classifier is not statistically significant. As such, in the improved RCBT (IRCBT) scheme we propose to employ the classifier with the highest confidence to predict the class label for the given sample. For a given test sample t and the l classifiers constructed with the same rule-based technique, all of the l classifiers are run to evaluate the classification score on the sample. The result of the most significant classifier is selected as the final result.

Definition 7: Given a classifier CL and a sample t, assuming CL's score of the sample is the order: $S(t \in c_1) > S(t \in c_2) \cdots S(t \in c_{|\mathcal{B}|})$, then the **significance** of the classifier is $S(t \in c_1) - S(t \in c_2)$.

The significance of a classifier's result on sample t is given in Definition 7, which is the score difference between the top two classes as predicted by the classifier. The definition is based on the observation that most of the misclassified cases take place between the top two classes with the highest score.

As an example, in Table 3, the significance of classifier CL_3 is 0.49, and the sample is classified as class 0 according to classifier CL_3.

7 EXPERIMENTS

In this section, the efficiency of the incremental LBR mining algorithm and the usefulness of *MaxSC* and *MinSC* are studied. All Algorithms are developed in the Visual C++ 6.0 environment. The binary code of MaxSC, MinSC and the IRCBT classification model is available at [41]. The experiments are run on a server with a Quad-Core AMD Opteron(tm) Processor 8356 (2.29GHz*16), and 127GB of RAM. In the experiments, only one CPU is used.

The following five datasets are used in the experiments: Bortezomib [21], DLBCL [25], Leukemia [11], Lung Cancer [12], and Prostate [26]. The general information of the five datasets is summarized in Table 4.

TABLE 4
General Information of Gene Expression Datasets

Dataset	#Gene	Class	Num.
Bortezomib	44928	R: NR	108
DLBCL	7129	DLBCL:FL	77
Leukemia	7129	ALL:AML	72
Lung cancer	12533	MPM:ADCA	181
Prostate	12600	tumor:normal	136

Discretization is a necessary preprocess before conducting association rule mining on gene expression data, which converts the continuous gene expression data to discretized item set. Same as [7], the entropy based discretion method is used in our work. The entropy based discretization benefits our work from the following two aspects: 1) discretization performs some kind of noise reduction, which is important in the noisy data; 2) most part of the label's information are maintain in the entrop based discretization procedure.

All experiments below are done by repeating the 3-fold cross validation procedure [16] is repeated multiple times to obtain average readings. In 3-fold cross validation approach, the dataset is randomly partitioned into three sets of equal size and the algorithm is executed three times, each time using one of the folds as the test set and the remaining two as the training set.

7.1 Efficiency

We first look at the efficiency of rule extraction. Two issues will be studied here, first the effectiveness of the pruning strategies and heuristic item ordering, then a comparison with the short rule first approach (we shall refer to this algorithm as *Short*)in [7]. The experimental results are based on repeating the 3-fold cross validation 50 times.

Efficiency of the Pruning Strategies: Four different variants of the incremental LBR mining algorithm are compared: **B** (basic method without pruning and heuristic item ordering), **H** (only with heuristic item ordering), **P** (only with pruning) and **HP** (heuristic + pruning). In the experiments, the algorithm is stopped if it can't finish in 10000 seconds.

Figure 3 shows the running time of the four variants with varying k when we want to find the top-k LBRs based on *MaxSC*. For the Prostate dataset only the **HP** variant can mine out the top-k LBRs in reasonable time (10000seconds) and thus we skip the comparison on this dataset. For the other four datasets, the time is reported if it can complete in 10000 seconds. From the graphs, we can see that both the pruning strategy and the heuristic item ordering strategy improve the efficiency of the algorithm dramatically. **HP** improves the efficiency of the basic incremental LBR mining algorithm by about 2 orders of magnitude. Moreover, the **H** variant generally spends more time than **P**. This is because, though **H** can find some interesting LBRs in the early stage, those LBRs can't be used to prune non-interesting candidates and the search space is still very large.

Figure 4 shows the running time of the four variants with varying k when we want to find the top-k LBRs based on *MinSC*. Similar to the experimental results of *MaxSC*, only the **HP** variant can complete in reasonable time on the Prostate dataset, and only the experimental results on the other four datasets are presented. Figure 4 shows that **HP** improves the efficiency of the basic incremental LBR mining algorithm by about 2 orders of magnitude in all datasets.

Comparisons with other methods: We next compare the running time of our algorithm with that of *Short* [7]. The computational time on the five datasets is presented in Table 5. In all the experiments here, we try to find the top 20 LBRs from 10 UBRs. The UBRs are extract using the algorithm in [7].

On Bortezomib, Leukemia and Prostate, *MinSC* is the fastest among the three algorithms. Though *Short* works efficiently on the DLBCL and Lung Cancer dataset, it is extremely inefficient on other two datasets. For example, for Prostate, it takes almost a day to finish. This is due to the fact that the LBRs on Prostate are actually much longer than that of other datasets and the breath-first search of *Short* becomes very inefficient. For example, one of the top-1 UBR of Prostate dataset contains 392 items, and the shortest LBR of this rule group contains 9 items. This means that the breadth-first search adopted by *Short* needs to generate $O(392^8)$ candidates before discovering the first LBR of the target rule group.

Though *MaxSC* works slower than *Short* in several datasets, it can mine out the LBRs within 10 minutes for all of them. This is still acceptable in real application. Among the algorithms, *MinSC* is the most efficient one.

TABLE 5
Running Time (seconds)

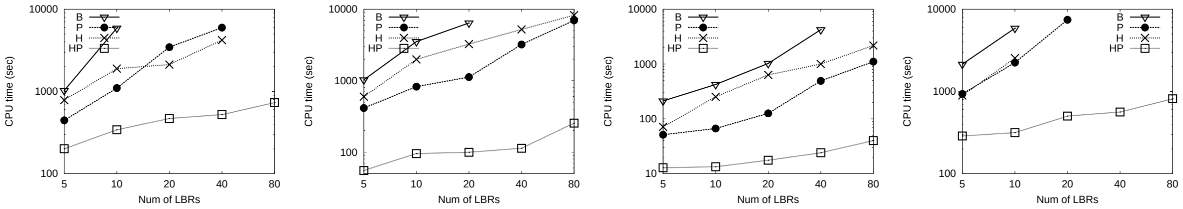
Dataset	*Short*	*MaxSC*	*MinSC*
Bortezomib	669.10	469.14	**2.33**
DLBCL	**2.78**	99.73	6.77
Leukemia	17.24	17.58	**0.80**
Lung Cancer	**13.53**	501.99	84.61
Prostate	90459.11	501.72	**27.84**
Average	18232.35	318.03	**24.47**

7.2 Classification Accuracy and Complexity

Next, we compare the classification accuracy of the three rule selection criteria: *Short*[7], *MinSC* and *MaxSC*. For the classification model, we compare IRCBT with the state of art RCBT.

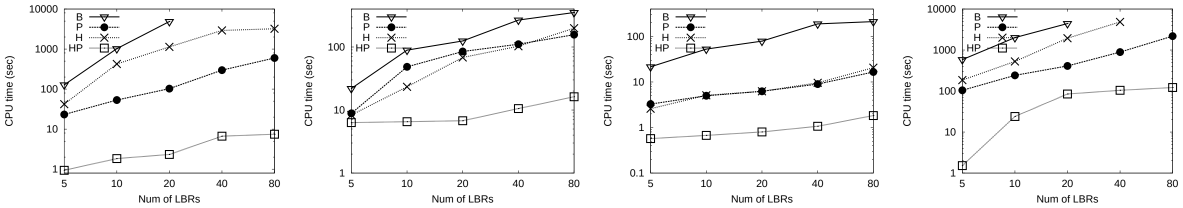
Note that the combination, *Short+RCBT*, corresponds to the *Top-K* method proposed in [7]. For the purpose of fair comparison, the top-10 covering rule groups of each sample are discovered and top-20 LBRs are generated for each rule group. These are the optimal parameter settings of *Short+RCBT* [7]. In addition, our method is compared with the state of art classifier SVM which is implemented based on *lib-SVM* version 2.87 [1]. To keep the comparisons fair, SVM is run using the same genes selected by entropy discretization, but with normalized real values of the gene expression data. The parameter of SVM is tuned with 3-fold cross validation on the training data.

Table 6 shows the classification accuracy of the 7 variants on the five datasets. All results are based on the average of 50 3-fold cross validation. The deviation from the average are also recorded in Table 6. The best classification accuracy of each dataset is presented in bold. Generally, *MinSC* +IRCBT achieves the best performance among all variants with the highest average accuracy and is also the top performance for four out of five datasets. Furthermore, its deviation from the average is also comparable to all the other variants. In fact, for



(a) Bortezomib (b) DLBCL (c) Leukemia (d) Lung Cancer

Fig. 3. Efficiency of The Pruning Strategies for *MaxSC*



(a) Bortezomib (b) DLBCL (c) Leukemia (d) Lung Cancer

Fig. 4. Efficiency of the Pruning Strategies for *MinSC*

TABLE 6
Classification Accuracy (%)

Interestingness Measure		*Short* [7]		*MaxSC*		*MinSC*	
Classifier	SVM	RCBT	IRCBT	RCBT	IRCBT	RCBT	IRCBT
Bortezomib	66.10±4.22	64.37±3.94	66.46±4.16	63.81±4.48	65.91±3.55	65.19±4.10	**66.52±3.81**
DLBCL	**88.42±3.52**	84.42±4.70	86.91±4.85	81.79±3.87	84.81±3.63	84.75±4.30	86.88±3.33
Leukemia	91.41±4.11	83.53±4.26	87.08±3.64	83.47±4.21	87.06±3.61	93.69±2.38	**95.28±2.04**
Lung Cancer	95.92±0.33	97.86±0.95	98.50±0.72	92.00±1.85	92.88±0.42	98.84±0.56	**99.13±0.42**
Prostate	76.41±5.21	73.38±3.15	75.00±3.92	73.97±3.57	76.69±3.95	76.18±3.94	**77.28±3.82**
Average	83.65	80.71	82.79	79.01	81.47	83.73	**85.02**

TABLE 7
Complexity of Classifiers

Interestingness Measure	*Short* [7]		*MaxSC*		*MinSC*	
Length	Num. of Genes	Leng. of Rules	Num. of Genes	Leng. of Rules	Num. of Genes	Leng. of Rules
Bortezomib	212.20±30.74	4.50±0.29	80.85±12.80	7.10±0.50	55.15±4.10	6.06±0.40
DLBCL	145.67±20.67	2.80±0.26	148.52±28.76	3.83±0.52	70.71±10.50	3.50±0.29
Leukemia	144.00±28.26	2.91±0.73	144.44±27.80	4.08±0.36	63.53±7.07	3.05±0.18
Lung Cancer	121.07±19.58	2.70±0.28	125.23±23.07	5.91±0.63	58.95±6.52	3.17±0.24
Prostate	192.6±13.82	7.83±0.63	97.90±24.27	8.22±0.64	84.70±15.71	8.14±0.80

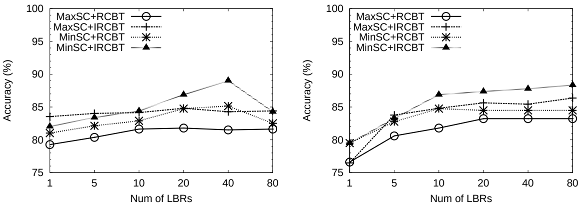
the Lung Cancer dataset where it has always 99% prediction accuracy, its deviation from the mean accuracy is the lowest.

Surprisingly, despite the fact that *MaxSC* is the most widely accepted interestingness measure [14], none of the variants involving it came out top in the performance. Our explanation for this interesting result is that the noisy nature of gene expression data in fact renders the statistical and biological reasoning behind *MaxSC* useless and that more work must be done to take noise into account when designing interestingness measure for ranking rules.

Sensitivity to parameter settings: Figure 5(a) shows the effect of varying number of LBRs and UBRs on the classification accuracy for dataset DLBCL. Studies on other datasets give similar result. Generally, the algorithms are robust to the number of LBRs and the number of UBRs.

In detail, Figure 5(a) shows that when the number of LBRs is very small, increasing the number of LBRs improves the accuracy of all methods. This is because when there are too few LBRs, the important information of the training dataset cannot be captured. When the number of LBRs is larger than 10, the classifier has already captured the main information of the dataset and adding more LBRs won't improve the accuracy of the algorithms. Moreover, when too many LBRs are selected, some less interesting LBRs will be selected which will have negative effect to the accuracy of the classifier. The decreasing of *MinSC*+IRCBT's accuracy when 80 LBRs are selected, verifies such property. This pheromone also shows the necessary of LBR selection.

Figure 5(b) shows the change of the accuracy with the number of UBRs. When the number of UBRs is very small, increasing the number of UBRs improves the accuracy. When the number of UBRs is large, the accuracies of IRCBT increase with the number of UBRs. This doesn't happen to RCBT because RCBT only uses the first covering classifier to classify the testing samples. If a testing sample can be handled by a small set of classifiers, the increasing of standby classifiers has no effect on the classification accuracy. IRCBT on the other hand uses the most significant classifier to classify the testing sample, and all the classifiers' information is explored before making prediction.



(a) Varying Num. of LBRs (b) Varying Num. of UBRs

Fig. 5. Sensitivity to the Parameters

Complexity of the Classifiers: We next look at the complexity of the classifiers that are being built. Table 7 shows the average number of genes and the average length of the rules that are involved in the classifiers for each of the interestingness measure. Note that since RCBT and IRCBT use exactly the same set of rules, there is no need to distinguish between these two classification schemes.

As expected, since *Short* always chooses the shortest rules, the average length of the rules selected by *Short* is always lower than *MaxSC* and *MinSC*. *Short* follows the belief of Occam's Razor that using the shortest rule in a rule-based classifier results in a much simpler classification model [19].

However, when we measure the complexity of the model using the average number of genes involved in the classifier, this conventional belief does not hold anymore. As can be seen from Table 7, when *MinSC* is used as the interestingness measure, the number of genes that involved in a classifier is substantially lower when compared to *Short*. For example, given the Leukemia dataset, the classifier built using *MinSC* involved on average 63.53 genes. While *Short* used around 144.00 genes to build a classifier that lost out substantially to *MinSC* in term of prediction accuracy. Compared to *MaxSC*, *MinSC*'s advantage is slightly reduced but is still substantial.

Note that involving a small number of genes in the classifier is also important in this case to biologists as they can focus on a smaller set of genes for investigation.

7.3 Biological Interpretation

One motivation for extracting rules from gene expression datasets is the ease of interpretation by the biologists. Here, we show some interesting results on the DLBCL dataset to illustrate this fact. The task on DLBCL is to classify two subclass of lymphoma (immune system cancer), Diffuse Large B-Cell lymphoma (DLBCL) and Follicular Lymphoma (FL).

We would like to emphasize that our problem is different from traditional ranking based methods. Instead of discovering the strongest rules, our method tries to find the optimal combinations of rules to maximize the prediction quality. Therefore, some rules with weak individual prediction ability, which may be ranked low in most rule analysis, can play important role on performance improvements. In this group of the sections, we will illustrate some example rules with low rank but high biological interestingness.

The mutual information based gene ranking [20] is used as the benchmark of the ranking based algorithms. The high ranked genes are generally considered more interesting than others and attract a lot of attentions, such as the genes MCM7, RCH1, CIP2 and CD69 in the DLBCL dataset [25]. However, as mentioned in Observation 1 in the introduction section, genes often perform certain function as a group and might not exhibit correlation with the class attribute in an isolated manner. Here, we will illustrate how such genes can be derived by adopting our interestingness measures and mining methodology.

We focus on genes that occur very often in the rules that we extracted based on *MaxSC* and *MinSC*. The genes are listed (heuristically, occur in more than 50 rules for both the measures) are listed in Table 8 together with their ranking based on mutual information. As can be seen, except for the first gene, all the other three genes are ranked low. Furthermore, these genes do not occur that frequently when the shortest rule measure is being used. We will next look at the biological significance of these genes.

TABLE 8
Common Frequent Genes of *MaxSC* and *MinSC*

Gene	Rank	Frequency in *Short*	Frequency in *MaxSC*	Frequency in *MinSC*
MCM7	1	64	64	64
RPL26	89	27	56	56
STRA13	184	18	89	122
NR1D2	494	11	94	56

Among all the genes, MCM7 is ranked the most significant based on both mutual information and our proposed interestingness measures. This is hardly surprising since MCM7 is homologeous to the DNA replication licensing factor CDC47 and is known to be highly associated with cellular proliferation and related to DLBCL[25].

The other three genes are RPL26, STRA13 and NR1D2 which are all ranked low based on mutual information. Those genes are not well known in the research on DLBCL and FL. However, there are extensive biological evidences showing that these three genes are related to DLBCL or FL. RPL26, Ribosomal Protein L26, is found to control p53 translation and induction after DNA damage[27] and DNA damage is considered as the most possible inducement of B-cell lymphoma [10]. As stated in [23], [24], STRA13 expression is developmentally regulated during B cell differentiation procedure, highly related to DLBCL and FL. NR1D2 is a member of nuclear receptor subfamily 1, which is found to be a new immune regulatory gene[15] and highly related to immune system cancer.

Finally, we see the three genes discovered by our algorithm are with low rank by traditional method, but with high biological interestingness. This verifies our motivation and implies the advantage of our rule based method.

8 Conclusions

In this paper, we propose two interestingness measures, *MaxSC* and *MinSC*, to rank LBRs within the same rule group. Considering the lattice structure of the LBRs, these two interestingness measures provide more information of the LBRs than traditional measures. An incremental top-k LBR mining framework is also developed to find the most interesting LBRs with respect to *MaxSC* or *MinSC*. This framework discovers interesting LBRs in the early stage of enumeration which maximizes the effectiveness of top-k pruning. Although the framework focuses on efficient mining of the top-k LBRs with the proposed measures, it can be easily extended to mining top-k patterns and rules with different interestingness measures. To make full use of the extracted rules, we introduce an additional classification scheme called IRCBT on the basis of traditional RCBT scheme. Experiments on various gene expression datasets show the efficiency and effectiveness of our proposals.

Acknowledgment

Support in part by NUS ARF Grant R-252-000-277-112 as part of the research project entitled GEMINI: Gene Expression Mining (http://nusdm.comp.nus.edu.sg/).

References

[1] www.csie.ntu.edu.tw/ cjlin/libsvm.

[2] R. Agrawal, et al. Mining association rules between sets of items in large databases. In *SIGMOD*, 1993.

[3] S. Brin, et al. Dynamic itemset counting and implication rules for market basket data. In *SIGMOD*, 1997.

[4] M. P. S. Brown, et al. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America*, 97(1):262–267, 2000.

[5] H. Cheng, et al. Discriminative frequent pattern analysis for effective classification. In *ICDE*, 2007.

[6] H. Cheng, et al. Direct discriminative pattern mining for effective classification. In *ICDE*, 2008.

[7] G. Cong, et al. Mining top-k covering rule groups for gene expression data. In *SIGMOD*, 2005.

[8] G. Cong, et al. Farmer: Finding interesting rule groups in microarray datasets. In *SIGMOD*, 2004.

[9] C. Creighton and S. Hanash. Mining gene expression databases for association rules. *Bioinformatics*, 19(1):79–86, 2003.

[10] A. Dent. B-cell lymphoma: suppressing a tumor suppressor. *Nature Medicine*, 11(22):22, 2005.

[11] T. Golub, et al. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, 286(5439):531, 1999.

[12] G. J. Gordon, et al. Translation of Microarray Data into Clinically Relevant Cancer Diagnostic Tests Using Gene Expression Ratios in Lung Cancer and Mesothelioma. *Cancer Research*, 62(17):4963–4967, 2002.

[13] M. Iwen, et al. Scalable Rule-Based Gene Expression Data Classification. In *ICDE*, 2008.

[14] R. Bayardo, R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense databases. In *ICDE*, pages 188–197, 1999.

[15] D. Koczan, et al. Gene expression profiling of peripheral blood mononuclear leukocytes from psoriasis patients identifies new immune regulatory molecules. *European Journal of Dermatology*, 15(4):251–258, 2005.

[16] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, 1995.

[17] J. Li, et al. Minimum description length principle: Generators are preferable to closed patterns. In *AAAI*, 2006.

[18] W. Li, et al. Cmar: Accurate and efficient classification based on multiple class-association rules. In *ICDM*, 2001.

[19] B. Liu, et al. Integrating classification and association rule mining. In *SIGKDD*, 1998.

[20] X. Liu, et al. An entropy-based gene selection method for cancer classification using microarray data. *BMC Bioinformatics*, 6(1):76, 2005.

[21] G. Mulligan, et al. Gene expression profiling and correlation with outcome in clinical trials of the proteasome inhibitor bortezomib. *Blood*, 109(8):3177–3188, 2007.

[22] F. Pan, et al. Carpenter: finding closed patterns in long biological datasets. In *SIGKDD*, 2003.

[23] M. Seimiya, et al. Clast5/stra13 is a negative regulator of b lymphocyte activation. *Biochemical and Biophysical Research Communications*, 292(1):121–127, 2002.

[24] M. Seimiya, et al. Impaired lymphocyte development and function in clast5/stra13/dec1-transgenic mice. *European Journal of Immunology*, 34(5):1322–1332, 2004.

[25] M. Shipp, et al. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8(1):68–74, 2002.

[26] D. Singh, et al. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203 – 209, 2002.

[27] M. Takagi, et al. Regulation of p53 translation and induction after dna damage by ribosomal protein l26 and nucleolin. *Cell*, 123(1):49 – 63, 2005.

[28] G. I. Webb. Discovering significant rules. In *SIGKDD*, 2006.

[29] G. I. Webb. Discovering significant patterns. *Machine Learning*, 71(1):131, 2008.

[30] X. Yin and J. Han. Cpar: Classification based on predictive association rules. In *SDM*, 2003.

[31] M. J. Zaki and K. Gouda. Fast vertical mining using diffsets. In *SIGKDD*, 2003.

[32] A. Reiner, D. Yekutieli and Y. Benjamini. Identifying differentially expressed genes using false discovery rate controlling procedures. Bioinformatics 19(3):368–375, 2003.

[33] G. Pandey, G. Atluri, M. Steinbach1 and V. Kumar Association Analysis Techniques for Discovering Functional Modules from Microarray Data. Nautre Precedings,2008.

[34] S. Dudoit, Y.H. Yang, M.J. Callow, and T.P. Speed Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. Stat. Sinica, 12, 111C-139, 2002

[35] B. Efron, R. Tibshirani, J.D. Storey and V. Tusher. Empirical bayes analysis of a microarray experiment. J. Am. Stat. Assoc.,96,1151C-1160.

[36] D. Jiang,C. Tang and A. Zhang. Cluster Analysis for Gene Expression Data: A Survey. TKDE, 16(11):1370–1386, 2004.

[37] M. Schena, D. Shalon, R.W. Davis, et al. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science, 270(5235): 467-470,1995.

[38] N. Friedman, I. Nachman and D. Pe'er. Using Bayesian Networks to Analyze Expression Data. In *RECOMB*, 2000.

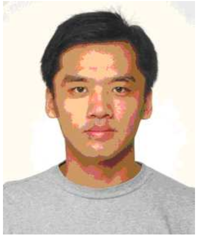
[39] H.C. Wang and Y.S. Lee Gene Network Prediction from Microarray Data by Association Rule and Dynamic Bayesian Network. LNCS, 3482:309–317, 2005.

[40] X.Q. Shang, Q. Zhao and Z.H. Li Mining High-Correlation Association Rules for Inferring Gene Regulation Networks. LNCS, 5691:244–255,2009.

[41] http://www.comp.nus.edu.sg/∼atung/lbGene.zip

Ruichu Cai Ruichu Cai received his B.S. in Applied Mathematics and PhD in Computer Science from South China University of Technology in 2005 and 2010, respectively. He is currently a Lecture in the Faculty of Computer Science, Guangdong University of Technology. He was visiting student of National University of Singapore in 2007/09-2009/09. His research interests cover a variety of different topics including association rule mining, clustering algorithms, feature selection.

Anthony K.H. Tung Anthony K.H. Tung received the BSc (Second Class Honor) and MSc degrees in computer science from the National University of Singapore (NUS) in 1997 and 1998, respectively, and the PhD degree in computer sciences from Simon Fraser University (SFU) in 2001. He is currently an Associate Professor in the Department of Computer Science, National University of Singapore. His research interests involve various aspects of databases and data mining (KDD) including buffer management, frequent pattern discovery, spatial clustering, outlier detection, and classification analysis. His recent interest also includes data mining for microarray data, sequences searches, social network and graph data processing, and dominant relationship analysis.



Zhenjie Zhang Zhenjie Zhang received his B.S. from Department of Computer Science and Engineering, Fudan University in 2004. He is currently a Ph.D. student and Research Fellow in the School of Computing, National University of Singapore. He was visiting student of Hong Kong University of Science and Technology in 2008 and visiting scholar of AT&T Shannon Lab in 2009. His research interests cover a variety of different topics including clustering analysis, skyline query, non-metric indexing and game theory. He has served as Program Committee members in VLDB 2010 and KDD 2010.

Zhifeng Hao Zhifeng Hao received the BSc degrees in Mathematics from the Sun Yat-Sen University in 1990, and the PhD degree in Mathematics from Nanjing University in 1995. He is currently a Professor in the Faculty of Computer Science, Guangdong University of Technology, and School of Computer Science and Engineering, South China University of Technology. His research interests involve various aspects of Algebra, Machine Learning, Data Mining, Evolutionary Algorithms.