Efficient Selection of Globally Optimal Rules on Large Imbalanced Data Based on Rule Coverage Relationship Analysis

Jinjiu Li * Can Wang * Longbing Cao * Philip S. Yu †

Abstract

Rule-based anomaly and fraud detection systems often suffer from massive false alerts against a huge number of enterprise transactions. A crucial and challenging problem is to effectively select a globally optimal rule set which can capture very rare anomalies dispersed in large-scale background transactions. The existing rule selection methods which suffer significantly from complex rule interactions and overlapping in large imbalanced data, often lead to very high false positive rate. In this paper, we analyze the interactions and relationships between rules and their coverage on transactions, and propose a novel metric, *Max Coverage Gain*. *Max Coverage Gain* selects the optimal rule set by evaluating the contribution of each rule in terms of overall performance to cut out those locally significant but globally redundant rules, without any negative impact on the recall. An effective algorithm, *MCGminer*, is then designed with a series of built-in mechanisms and pruning strategies to handle complex rule interactions and reduce computational complexity towards identifying the globally optimal rule set. Substantial experiments on 13 UCI data sets and a real time online banking transactional database demonstrate that *MCGminer* achieves significant improvement on both accuracy, scalability, stability and efficiency on large imbalanced data compared to several state-of-the-art rule selection techniques.

1 Introduction

Rule-based methods are crucial in real-world risk management, expert systems and decision support systems. General rule-based systems consist of two key technical stages: rule generation and rule selection. At stage one, a large number of candidate rules are generated. At stage two, the best rule set is selected for decision-making such as prediction of fraud. In practices, when engaging a large scale of business data, the quality of a rule-based system highly relies on an effective way to select the globally optimal rule set.

Let us firstly illustrate the problem of globally optimal rule selection on a highly imbalanced data set. Table 1 displays a typical task for fraud detection in online banking. The number of fraud transactions (Fraud) is far smaller than that of genuine transactions (Genuine), which is imbalanced with the ratio 1:64000. In addition, the number of transactions is rather huge (i.e. 32,000,500), and the dimension is also high (i.e. 220). We call this kind of data as a *large imbalanced data set*. The desirable fraud prediction model is expected to catch 70% of Fraud with the False Positive Rate (FPR=|False Alerts|/|Alerts|) at no more than 90%, and the minimum support of each rule in Fraud is no less than 0.02. Accordingly, the confidence (=1-FPR) needs to be larger than 0.1. It means that for every 10 detected Fraud, the number of alerts triggered by the prediction model should not exceed 100. Thus, the minimum support in Genuine needs to be set around 0.000003[1]. Further, for the real transaction data set in our experiments, the total number of candidate rules whose confidences are bigger than 0.1 is 8,500,000. A huge number of rules brings another challenging issue, i.e. rule overlapping. Lots of rules are built in a similar structure, which means that each transaction is matched by 2045 rules on average. Added to this, rule selection needs to also cater for business impact. For instance, a constraint on selecting rules is the Benefit On Investment (BOI), referring to the ratio between the total amount of money recoverable from the Fraud detected and the investigation fee caused by all the alerts. The higher the BOI, the better the prediction.

Table 1: An Example of the Task for Fraud Detection

Data Set		Prediction Objectives	
Fraud	500	Min Support in Fraud	≥ 0.02
Genuine	32,000,000	False Positive Rate	$\leq 90\%$
Total	32,000,500	Detection Rate	$\geq 70\%$
Features	220	Benefit On Investment	> 1

The above examples expose the following challenges of rule selection on the large class-imbalanced data sets:

- It is very time and memory intensive to handle a large scale of transactions during the process of optimal rule selection.
- A high dimensional rule space and a large number of rules lead to the serious rules overlapping, which requires the deep analysis of rule relationship, and

*Advanced Analytics Institute, University of Technology, Sydney. {Jinjiu.Li, CanWang613, Longbing.cao}@gmail.com.

†Computer Science Department, University of Illinois at Chicago, USA. psyug@uic.edu.

[1]The calculation details are in the Appendix "https://docs.google.com/file/d/0B7-9myg-dRYALVRrenRIV1FiQzg/edit". Note that all the followed contents in the Appendix are linked to this online document.

effective selection methods to obtain the optimal rule set that satisfies the desirable objectives.

– Rule selection on the imbalanced data is also cost sensitive. Missing a fraud, which may cause a big amount of credit loss, is more expensive than incorrectly labeling a Genuine as a Fraud. The selection method should consider two factors together: the number of rules and their significance.

– High FPR is potentially a serious problem on the large business data. It results in a large volume of false alerts that cause expensive investigation fees. Therefore, we need to find the globally optimal rule set under specific criteria, rather than the non-optimal rule set proposed by approximate methods [2, 3].

In addition, the rules need to be considered from the perspective of their global contribution rather than the individual prediction capability. In most class-imbalance scenarios, a rule with a high confidence often mistakenly alerts a large number of negative instances as well. The following example explains this issue: A rule r_0 has caught three Fraud $\{t_1, t_2, t_3\}$ but mistakenly alerted 297 Genuine. Rule r_1 has caught one Fraud $\{t_1\}$ but alerted 120 Genuine (denoted as T_1). Rule r_2 has caught one Fraud $\{t_2\}$ but alerted 120 Genuine (denoted as T_2), and rule r_3 has caught one Fraud $\{t_3\}$ but alerted 120 Genuine (denoted as T_3). Here, we have $|T_1 \cap T_2| = 90, |T_1 \cap T_3| = 80, |T_2 \cap T_3| = 0$. Then, the rule set $\{r_1, r_2, r_3\}$ catches all the three Fraud with only 190 Genuine alerted. Obviously, it outperforms $\{r_0\}$ in terms of both BOI and FPR, although $\{r_0\}$ has the highest confidence and detection rate.

The existing rule selection methods cannot be applied on a large class-imbalance data set and are not able to capture the globally optimal rule set effectively. Therefore, this paper focuses on extracting the most effective rule combinations on the large and highly imbalanced data by analyzing the intrinsic rule relationships [7] between rules and transactions. The key contributions are as follows:

– We formalize the interactions and relationships between rules and their coverage on transactions, by defining a series of effective mechanisms to identify the optimal rule combinations effectively.

– We propose a novel metric, *Maximal Coverage Gain (MCG)*, to evaluate the quality of the extracted rule set. *MCG* has the following advantages against the existing methods: 1) Minimize the rules overlapping; 2) Target on the globally optimal performance rather than the local optimization; 3) Minimize the FPR; 4) Maximize the business utility; and 5) Solve the cost-sensitive problem by attaching gain factors to instances.

– We develop an effective algorithm *MCGminer* to quickly find the optimal rule set by a divide-and-conquer process. Furthermore, a series of heuristic methods and pruning strategies are introduced to cut the computational and memory costs sharply.

– We evaluate our proposed method with the state-of-the-art techniques on a variety of large transactional data in terms of prediction accuracy and stability against the increase of imbalance rate, effectiveness of our pruning strategies, and scalability[2].

The paper is organized as follows. In Section 2, we review the related work. Preliminary definitions are specified in Section 3. Section 4 describes the defined problem and pruning strategies, and presents algorithms to efficiently search an optimal coverage rule set. Section 4 shows the experimental evaluations. We conclude in Section 6.

2 Related Work

While extensive research efforts have mainly focused on generating rules, the existing methods for selecting rules are generally categorized as below: coverage based methods [4, 5], which tend to search the optimal rule set by a greedy algorithm; and multi-criterion based methods, which target the non-dominated rule set [9] or apply the integrated ranking over multiple measures to choose a final rule set [6]. The existing methods cannot be applied to the large imbalanced data directly due to the following disadvantages: Implementing a sequential covering test [5] generates too many redundancies, which negatively impact the prediction accuracy; Severe rules overlapping inevitably leads to a high FPR; Multi-criterion based methods tend to judge a rule from a local perspective. Further, the disjoint rules are incomparable, they cannot prune each other by any mutually exclusive criteria. Thus, even after pruning, the number of survival rules is still very large since a huge number of candidate rules have been mined. The concrete examples for such limitations are detailed in the attached Appendix (a). Alternatively, some researchers use the Genetic Algorithm to find the non-dominant rule set [9]. Those methods output approximate results, which potentially cause a high FPR, as illustrated in Section 1. Different from their approaches, our method provides an accurate globally optimal rule set.

In summary, the existing rule selection methods cannot deliver the best outcomes without compromise of prediction quality on the large imbalanced data. To the best of our knowledge, no existing research work has been reported on finding the globally optimal rule set from the perspective

[2]The mechanisms have been tested and deployed into an online banking risk management system i-Alertor [1] installed in a major Australian bank. The system is much more effective and efficient than the current expert system used by major Australian banks.

Table 2: Transactions of Online Banking

TID	*Credit*	*Newpayee*	*Channel*	*Benefit*	*Class*
t_1	\$1000	0	Bpay	\$-100	$Genuine$
t_2	\$800	0	PayAnyone	\$-100	$Genuine$
t_3	\$5000	1	PayAnyone	\$+5000	$Fraud$
t_4	\$500	0	Bpay	\$+500	$Fraud$
t_5	\$30	1	PayAnyone	\$+30	$Fraud$
t_6	\$800	1	PayAnyone	\$-100	$Genuine$
t_7	\$3000	1	Bpay	\$+3000	$Fraud$

of maximal coverage gain on the large imbalanced data set. This paper is motivated by this challenging problem.

3 Problem Statement

A *transaction* is formalized as a vector $t = \{v_1, \cdots, v_n, w\}$, where v_i is the corresponding value of attribute a_j ($1 \leq j \leq n$) and w is a *gain factor* of t. As a signed real number, gain factor w represents the business benefit of classifying t as targeting class. w can be positive or negative. The length of transaction t is defined as the number of involved attributes, i.e., $|t| = n$. A set of transactions consist of the *transaction base* T. Table 2 presents an example of fraud detection in online banking, where *TID* is the serial number of a transaction and Benefit is the profit recovered from alerting the current transaction as a Fraud. Then, the transaction base is $T = \{t_1, \cdots .t_7\}$, each transaction t_i ($1 \leq i \leq 7$) is described by attributes {*TID, Credit, Newpayee, Channel*}, gain factor {*Benefit*}, and label {*Class*}. The length of each transaction is 4.

Based on the tuple representation of a transaction, a *literal* is defined as an attribute-value pair, i.e., $l = (a_k, v_k^l)$, in which a_k is an attribute and v_k^l is a value of a_k. A transaction t is regarded to satisfy a literal $l = (a_k, v_k^l)$ if and only if $v_k = v_k^l$, where v_k is the value of attribute a_k in transaction t. For example, we have (*Newpayee*, 1) to be a valid literal in Table 2, and transaction set $T' = \{t_3, t_5, t_6, t_7\}$ satisfies this literal. Accordingly, we define a *rule* in the following.

DEFINITION 1. *A **rule** is a conjunction of multiple literals with an associated class label, formalized as $r : l_1 \wedge \cdots \wedge l_m \rightarrow c$, where c is a class label, m is the number of literals. The length of rule r is the number of literals, i.e., $|r| = m$.*

EXAMPLE 1. *There are four rules r_1 to r_4,*
r_1 : (*Credit, Large*) $\wedge$ (*Newpayee*, 1) $\rightarrow$ *Fraud*
r_2 : (*Credit, Median*) $\wedge$ (*Newpayee*, 0) $\rightarrow$ *Fraud*
r_3 : (*Newpayee*, 0) $\rightarrow$ *Fraud*
r_4 : (*Credit, Median*) $\rightarrow$ *Fraud*
So, $|r_1| = 2, |r_2| = 2, |r_3| = 1, |r_4| = 1$. Here, attribute Credit is converted into nominal attribute from numeric one by the discretization principle, Small: Credit < 500, Median: Credit $\in [500, 2000)$, Large: Credit ≥ 2000.

For simplicity, we take one class label as our target to select the optimal rule set, so all the rules are involved to predict the specific class label. If there are multiple classes to be predicted, we can simply process them separately. In Example 1, we target the fraud class, so the resultant rule set is used to predict fraud.

DEFINITION 2. *A rule r can **cover** transaction t if and only if every attribute value v_i of transaction t satisfies its corresponding literal in rule r, denoted as $t \models r$.*

According to Table 2, in Example 1, we have $t_3 \models r_1, t_7 \models r_1, t_1 \models r_2, t_2 \models r_2, t_4 \models r_2$. Moreover, if we consider a rule set $R = \{r_1, \cdots, r_p\}$ in transaction base T, then the transactions covered by R form a transaction set $C_{[R,T]}$, formalized as $C_{[R,T]} = \{t | t \models r, \exists r \in R, t \in T\}$. In Example 1, $C_{[R,T]} = \{t_1, t_2, t_3, t_4, t_7\}$ if we take $R = \{r_1, r_2\}$ for Table 2. Further, we obtain the transaction set covered by a union of two rule sets by the formula $C_{[R_1 \cup R_2, T]} = C_{[R_1,T]} \cup C_{[R_2,T]}$.

Based on the covered transaction set $C_{[R,T]}$, accordingly, we propose the following coupled relationships over transaction set T between two rules r_i and r_j:

- *Overlapped Rules*: $C_{[\{r_i\},T]} \cap C_{[\{r_j\},T]} \neq \emptyset$
- *Independent Rules*: $C_{[\{r_i\},T]} \cap C_{[\{r_j\},T]} = \emptyset$
- *Coincident Rules*: $C_{[\{r_i\},T]} = C_{[\{r_j\},T]}$

where coincident rules are special cases of overlapped ones. In Example 1, r_4 is *overlapped* with r_2, r_1 is *independent* with r_2. Since $C_{[\{r_2\},T]} = C_{[\{r_3\},T]}$, r_2 and r_3 are called *coincident* rules according to the above conditions. Moreover, the coupled relationships can be extended to the rule set simply. $R_1 \subseteq R$ and $R_2 \subseteq R$, the coupled relationships of R_1 and R_2 over transaction set T are defined:

- *Overlapped Rule sets*: $C_{[R_1,T]} \cap C_{[R_2,T]} \neq \emptyset$
- *Independent Rule sets*: $C_{[R_1,T]} \cap C_{[R_2,T]} = \emptyset$
- *Coincident Rule sets*: $C_{[R_1,T]} = C_{[R_2,T]}$

Note that, a rule can also be independent of the rest of rules in R. Thus, we define the independence of R' as:

- *Island*: $R' (\subseteq R)$ is an island if $C_{[R',T]} \cap C_{[R \setminus R',T]} = \emptyset$.

Besides, we extend the overlapping concept to define the dependent overlapping between a single rule $r_0 \in R$ and R as $O_{\{r_0\}} = \{r | r \in R \setminus \{r_0\}, C_{[\{r\},T]} \cap C_{[\{r_0\},T]} \neq \emptyset\}$. Here $O_{\{r_0\}}$ stands for the rule set that overlapped with r_0. In Example 1, assume $R = \{r_1, r_2, r_3, r_4\}$, we have $O_{\{r_2\}} = \{r_3, r_4\}$. Besides, the transactions in subset $T' (\subseteq T)$ can be merged as a virtual transaction $t_{T'}$, and T' is called:

- *Transaction Block*: If either $C_{[\{r\},T]} \cap T' = \emptyset$ or $C_{[\{r\},T]} \cap T' = T'$ holds for any rule $r \in R$.

Accordingly, the gain factor of $t_{T'}$ is $w_{t_{T'}} = \sum_{t \in T'} w_t$. We also have that $t_{T'} \models r$ if $T' \models r$ holds for any $r \in R$.

Based on the above concepts, we are ready to propose a metric, i.e., coverage gain, to evaluate the effectiveness of rule r in transaction data T in terms of classification power.

DEFINITION 3. ***Coverage Gain*** *is defined to sum the gain factors of transactions covered by rule r in T:*

$$g(\{r\}, T) = \sum_{t \in C_{[\{r\},T]}} w_t, \tag{3.1}$$

where w_t is the gain factor of transaction t.

As we know, BOI is a key factor of the most business concern. Thus, the optimal rule set is expected to be able to save the maximal loss. Certainly, if the detection number is the most interesting metric, we can simply set the gain factor of fraud and genuine to be +1 and -1 respectively.

Multiple rules $\{r_1, \cdots, r_p\}$ can be merged together to form a super rule $\tilde{R} = r_1 \cup r_2 \cup \cdots \cup r_p$. Then, the corresponding *coverage gain* of $\tilde{R}$ is:

$$g(\tilde{R}, T) = \sum_{t \in C_{[\{r_1\},T]} \cup \cdots \cup C_{[\{r_n\},T]}} w_t. \tag{3.2}$$

Further, for the rule set R, we define the rule set with the maximal coverage as follows.

DEFINITION 4. *Given a rule set R, if there exists a rule subset $\widehat{R} \subseteq R$, and for any $R' \subseteq R$, such that $g(R', T) \leq g(\widehat{R}, T)$, then $\widehat{R}$ is called the* ***Maximal Coverage Set*** *of R.*

In other words, $\widehat{R}$ is the minimal rule subset that obtains the maximal coverage gain. Note that $\widehat{R} = R$ does not necessarily hold.

Finally, rule selection can be formalized as the problem of *Maximal Coverage Gain Mining*.

DEFINITION 5. *The goal of rule selection is to find the smallest rule subset $\widehat{R}$ that has the* ***Maximal Coverage Gain (MCG)****, denoted as $G(R, T) = g(\widehat{R}, T)$.*

Suppose $|R| = p$, then the total number of subsets in rule set R is 2^p, so the computational complexity for a brutal-force method to discover $\widehat{R}$ is $O(2^p)$. Since the MCG mining with the brutal-force means is time-consuming and possibly cannot achieve the result in an acceptable time period, we develop a set of pruning strategies and index structure to speed up the whole process in the following sections.

4 Maximal Coverage Gain Mining

In this section, the algorithm for the MCG mining is proposed with pruning strategies, hinge set discovery and gain bounding. Pruning strategies aim at cutting the searching cost based on analyzing coupled relationships among individual rules and rule sets over data set T. Hinge set is the key part for applying the divide-and-conquer framework to cut the searching computational cost during the iteration process. Gain bounding studies the minimal (f_{min}) and maximal (f_{max}) contribution of a single rule to $MCG(R, T)$ from the global perspective. f_{min} and f_{max} can be effectively used to judge the qualification of rules.

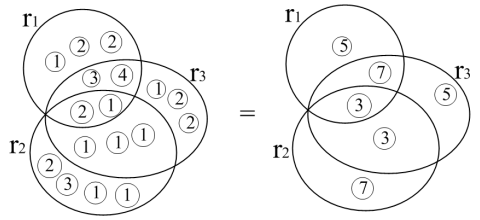

Figure 1: Transaction merging, each dot represents a transaction, and the value in every dot is the gain factor.

4.1 Pruning Strategies To mine MCG efficiently, we develop a series of pruning strategies from three levels: transaction reduction, rule deduction, and group interaction. In the following, we explicate these three levels separately. All the proofs supporting the theorems below are provided in the attached Appendix (b).

4.1.1 Transaction Reduction Transaction reduction targets saving memory and reducing the computational cost in data set scanning by merging and eliminating transactions.

1) Transaction Merging: The transaction subset T' can be merged as a virtual transaction $t_{T'}$ if T' is a transaction block, since the merging brings about no cost on *MCG*, i.e., $G(R, T)$. This property is formalized in Theorem 4.1 below.

THEOREM 4.1. *Given a transaction set $T' \subseteq T$, $t_{T'}$ is the virtual transaction merged for T'. Let $T'' = (T \backslash T') \cup \{t_{T'}\}$, then we have $G(R, T) = G(R, T'')$. If $w_{t_{T'}} = 0$, then we have $G(R, T) = G(R, T \setminus T')$.*

For example, in Figure 1, the number of transactions decreases dramatically from 17 to 6 after transaction merging. The merge operation helps to reduce the scan cost over transactions and save the memory.

2) Transaction Evacuation: Once a rule r is selected as an element of $\widehat{R}$, r and the transactions covered by r can be removed immediately, since it has no impact on $G(R, T)$ finally. Formally, for a set of such rules, we have:

THEOREM 4.2. *For $\forall R' \subseteq \widehat{R}$, we have $G(R, T) = g(R', T) + G(R \setminus R', T \setminus C_{[R', T]})$.*

As evidenced by our experiments on different data sets (see Section 5.5), by removing the transactions covered by $R' \subseteq \widehat{R}$, the candidate rule number also shrinks dramatically due to the elimination of many plain rules which no longer cover transactions. More importantly, after taking away one or two rules around R', a big island can be split into multiple smaller ones (defined in Section 3) easily. For example, Figure 2 shows such an island. Suppose r' is confirmed to be put into $\widehat{R}$, after collecting r' and removing the transactions it covered, a ring-like new island appears. This new island is easier to be divided into smaller islands (i.e., R_a, R_b and R_c)

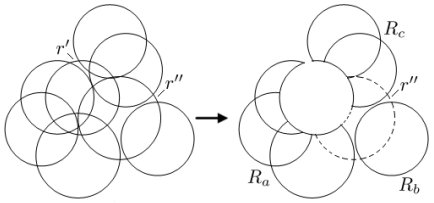

Figure 2: Islands after removing r'.

by taking r'' away. Here, $\{r''\}$ becomes a key connection of the new island, we define it as *Hinge Set* (described in Section 4.2) of the new island. However, compensation needs to be given for taking r'' away in splitting the new island. In Section 4.1.3, we introduce the enumeration process to compensate the impact of removing r''.

4.1.2 Rule Deduction Rule deduction focuses on elimination of redundant rules and the qualification for becoming an element of $\widehat{R}$ as early as possible based on relationship between a single rule and the rule set. The theorem and remark below formalize how to perform the rule deduction.

1) Coincident Rules: As we know, often too many rules are produced in the rule generation phase, so the rule overlapping over T is serious. As a result, many transactions are matched by multiple rules.

THEOREM 4.3. *If $r_1 \in R$ and $r_2 \in R$ are coincident rules over T, then $G(R,T) = G(R \setminus \{r_1\}, T) = G(R \setminus \{r_2\}, T)$.*

So, for the coincident rules, we only keep the distinct one and remove the rest.

2) Maximal/Minimal Contribution of Rules: Apart from the above coincident rules, rules can be selected according to their contribution to *MCG* (i.e., $G(R,T)$). Formally:

THEOREM 4.4. *Given a rule $r \in R$, let*

$$f_{max}(r) = \max_{R' \subseteq O_{\{r\}}} g(\{r\}, T \setminus C_{[R',T]}), \tag{4.3}$$

$$f_{min}(r) = \min_{R' \subseteq O_{\{r\}}} g(\{r\}, T \setminus C_{[R',T]}), \tag{4.4}$$

where $O_{\{r\}}$ is the overlapping rule set of rule r. If $f_{max}(r) < 0$, then $r \notin \widehat{R}$; If $f_{min}(r) > 0$, then $r \in \widehat{R}$. Here, $\widehat{R}$ is the maximal coverage set of R.

Note that $f_{max}(r)$ is the *contribution upper bound* of r to $G(R,T)$. If $f_{max}(r) \leq 0$, the contribution of r is negative, so r can be removed directly. $f_{min}(r)$ is the *contribution lower bound* of r to $G(R,T)$. When $f_{min}(r) > 0$, the contribution of r is positive, so r can be put into $\widehat{R}$ immediately. Furthermore, we can estimate the contribution of a rule subset $R'' \subseteq R$ as well. Specifically,

$$f_{max}(R'') = \max_{R' \subseteq O_{R''}} g(R'', T \setminus C_{[R',T]}), \tag{4.5}$$

$$f_{min}(R'') = \min_{R' \subseteq O_{R''}} g(R'', T \setminus C_{[R',T]}). \tag{4.6}$$

4.1.3 Group Interaction Group interaction studies the necessary and sufficient conditions for rule pruning based on the coupled relationships among groups. We adopt the Divide-and-Conquer concept to split a big rule group into several smaller islands, which are easily to be processed. The theorems below lay a solid foundation for group interaction.

1) Island Combination: MCG is a quality measurement of the optimal rule set from the global perspective. It is built by the contribution from all islands. We have the following method to combine the coverage gain of all islands.

THEOREM 4.5. *Given islands $R', R'' \subseteq R$, if $R' \cap R'' = \emptyset$, then $G(R' \cup R'', T) = G(R', T) + G(R'', T)$.*

According to Theorem 4.5, if the rule set R consists of multiple islands, namely, $R = R_1 \cup \cdots \cup R_q$, then the *MCG* problem can be divided into multiple independent sub-problems that are easier to solve. Formally, we have $G(R,T) = G(R_1,T) + \cdots + G(R_q,T)$.

2) Group Mutex: We can use the coupled relationship to judge the qualification of a rule group. A concrete example is explained in the attached Appendix(b). We call this property *group mutex*, having the following theorem:

THEOREM 4.6. *Given a rule $r \in R$, if $g(\{r\}, C_{[\{r\},T]} \setminus C_{[O_{\{r\}},T]}) < 0$ holds, then $(O_{\{r\}} \cup \{r\}) \not\subseteq \widehat{R}$.*

Theorem 4.6 judges the qualification of rules under consideration based on the confirmed rule group in $\widehat{R}$. According to our experiments, the group mutex increasingly prunes rules as more rules are added into $\widehat{R}$, especially when the rules overlap in $\widehat{R}$ frequently.

3) Rule Association: The following theorem judges the qualification of r when the rules overlapped with it are all determined.

THEOREM 4.7. *Given a rule $r \in R$, let $R' = O_{\{r\}} \cap \widehat{R}$, if $g(\{r\}, T \setminus C_{[R',T]}) > 0$ holds, then $r \in \widehat{R}$.*

4) Island Splitting: According to the complexity analysis in Section 3, enumerating all the subgroups of a big island is costly. So we adopt the Divide-and-Conquer method to resolve the problem. Here, we introduce the method for splitting big islands. The enumeration tree is used to describe how the compensation is given after splitting a big island.

THEOREM 4.8. *Given a rule $r \in R$, then $G(R,T) = \max(G(R \setminus \{r\}, T), G(R \setminus \{r\}, T \setminus C_{[\{r\},T]}) + g(\{r\}, T))$.*

The theorem above tries to evaluate the impact of eliminating r from R recklessly. Intuitively, in order to counterbalance the impact, we calculate MCG with two solutions, removing r and adding r into $\widehat{R}$, respectively. The higher one is the optimal solution. The above theorem can also be extended to handle the case when multiple rules are under consideration of elimination. We have the following remark.

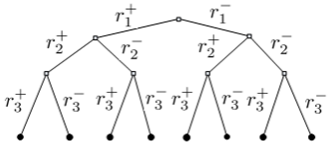

Figure 3: Enumeration process.

REMARK 1. *Given an island $R' \subseteq R$, if there exists $R'^H = \{r_1, \cdots, r_p\} \subseteq R'$, and $R'' = R' \backslash R'^H$ consists of k $(k \geq 2)$ islands, then we have:*

$$G(R', T) = \max_{R'^{H'} \subseteq R'^H} (G(R'', T \setminus C_{[R'^{H'}, T]}) + g(R'^{H'}, T)).$$

R'^H is called the Hinge Set of R'. The hinge set is the key set to unite the whole islands.

For example, in Figure 2, the hinge set is $R'^H = \{r', r''\}$. It connects three sub-islands R_a, R_b and R_c, the lengths of these sub-islands are $|R_a| = 3, |R_b| = 1$ and $|R_c| = 2$.

A big island can be split into multiple smaller sub-islands for efficient processing. After taking away the hinge set, a compensation needs to be given in order to cancel out its impact on *MCG*. Here, an enumeration process is executed based on Remark 1.

According to Remark 1, in order to compensate the impact caused by removing the hinge set, an enumeration process is executed, as shown in Figure 3. Hinge set $R'^H = \{r_1, r_2, r_3\}$ contains 3 rules, they are enumerated orderly, r_1^+ means that rule r_1 is supposed to be selected into $\widehat{R}$, while r_1^- indicates that rule r_1 is removed from $\widehat{R}$. So there are eight round tests during the enumeration process, i.e., $\{r_1^+, r_2^+, r_3^- +\}, \{r_1^+, r_2^+, r_3^-\}, \{r_1^+, r_2^-, r_3^+\}, \{r_1^+, r_2^-, r_3^-\}$, $\{r_1^-, r_2^+, r_3^+\}, \{r_1^-, r_2^+, r_3^-\}, \{r_1^-, r_2^-, r_3^+\}, \{r_1^-, r_2^-, r_3^-\}$.

In Figure 2, $n_a = |R_a|, n_b = |R_b|$ and $n_c = |R_c|$, the computational cost of a brutal-force method is $O(2^{(3+n_a+n_b+n_c)})$. After splitting the big island into three smaller islands, the complexity reduces to $O(2^{(3+max(n_a,n_b,n_c))})$. If $max(n_a, n_b, n_c)$ is still large, the islands whose lengths exceed K (i.e., a pre-defined threshold of the maximal rule number in an island when calculating *MCG* of this island immediately, in our experiment, we choose $K = 3$) can be divided iteratively, until the size of island is less than K.

An island may have multiple hinge sets, finding a small one that connects multiple similar-size islands is very important to reduce the computational cost. There are two key factors to determine the computational cost for the enumeration process: the balance in size among all sub-islands generated (we call it *splitting gap* for short) and the size of hinge set (we call it *hinge scale* for short). In Figure 2, the splitting gap is $max(n_a, n_b, n_c) - min(n_a, n_b, n_c) = 3 - 1 = 2$, and the hinge scale is 2 (the hinge set contains 2 rules r' and r'').

Table 3: Diagonal Matrix of R on T'

	r_2	r_3	r_1	r_4	r_5	r_6	r_7
T_{13}	0	0	0	0	0	0	1
T_{14}	0	0	0	0	0	1	0
T_{15}	0	0	0	0	0	1	1
T_9	0	0	0	0	1	0	0
T_{10}	0	0	0	0	1	0	1
T_{11}	0	0	0	0	1	1	0
T_{12}	0	0	0	0	1	1	1
T_7	0	0	0	1	0	0	0
T_8	0	0	0	1	1	0	0
T_1	0	0	1	0	0	0	0
T_3	0	1	0	0	0	0	0
T_6	0	1	0	1	0	0	0
T_2	0	1	1	0	0	0	0
T_5	1	0	0	0	0	0	0
T_4	1	1	0	0	0	0	0

The computational complexity of a straightforward way to discover an optimal hinge set is $O(n! \times m!)$, where n is the number of transaction blocks covered by R', and m is the number of rules in R'. An effective method to find an optimal hinge set is discussed in the next section.

4.2 Hinge Set Discovery In this part, we specify in detail how to discover the hinge set of an island. Firstly, we build an adjacent matrix for R over T. Each column in Table 3 stands for a rule, each row stands for a transaction block, the value in the matrix can be either 0 or 1. Let m_{ij} stand for a joint cell of row i and column j in the matrix, $m_{ij} = 1$ means $T_i \models r_j$, otherwise $m_{ij} = 0$ means $T_i \not\models r_j$. Next, the algorithm for the hinge set discovery is introduced by the observations step by step.

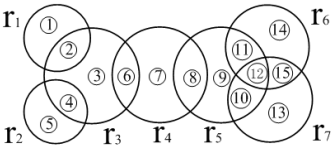

Figure 4: Coverage of R on T'.

In Figure 4, $R = \{r_1, \cdots, r_7\}$ on the transaction block set $T' = \{T_1, \cdots, T_{15}\}$. After carefully arranging the order of elements in R and T', the adjacency matrix can be transformed into an approximate diagonal matrix, as shown in Table 3. If rule r_4 and the transaction blocks (i.e., T_7, T_8, T_6) covered by this rule are removed, then the diagonal matrix in Table 3 can be divided into two isolated diagonal blocks: $(\{r_1, r_2, r_3\}, \{T_1, T_3, T_2, T_5, T_4\})$ and $(\{r_5, r_6, r_7\}, \{T_{13}, T_{14}, T_{15}, T_9, T_{10}, T_{11}, T_{12}\})$. Therefore, rule r_4 is the optimal hinge set of R on T'.

When given a rule set R and a transaction block set T', we have the following observations in terms of identifying the optimal hinge set.

– An adjacent matrix can be converted into a diagonal matrix by adjusting the order of columns. (detailed in Observation 1 in the Appendix (c))

– In a group of rules, it is most likely to obtain an approximate diagonal matrix by putting the rules that contain fewer transaction blocks at the first few columns. (detailed in Observation 2 in the Appendix (c))

– Choosing the rules that contain more transaction blocks as a part of the hinge set can generate more islands. (detailed in Observation 3 in the Appendix (c))

– The hinge set is identified by recursively checking the columns of the diagonal matrix from the right side to the left side. (detailed in Observation 4 in the Appendix (c))

Accordingly, Algorithm 1 attached in the Appendix (d) is designed to find the optimal hinge set in an efficient way.

4.3 Gain Bounding In this part, we study the maximal and minimal gain of a single rule from the perspective of global contribution. Intuitively, if a rule r's contribution for $MCG(R,T)$ is positive, then r should be added into $\widehat{R}$; otherwise, r should be removed.

Before applying hinge set to split big islands, the contribution bound of rule set, i.e., $f_{max}(R)$ (4.5) and $f_{min}(R)$ (4.6), can be used to perform bound checking to determine the qualification of rules at the beginning. We call this operation *Gain Bounding*. There are three types of rules that can be detached to shrink the size of islands: coincident rules, core rules whose $f_{min} > 0$, redundant rules whose $f_{max} < 0$. Algorithm 2 attached in the Appendix (d) tends to find these rules. The calculation of $f_{min}(r)$ can be converted to the problem of calculating $G(O_r, C_{[\{r\},T]})$, so $f_{min}(r,R,T) = g(r,T) - G(O_r, C_{[r,T]})$. Algorithm 3 attached in the Appendix (d) performs the calculation of $f_{min}(r,R,T)$ based on this transition. The computation on the upper bound of r can be transformed to the problem of computing $G(r,T)$ as well. According to the definition of $f_{max}(r)$ in (4.3), we have $f_{max}(r) = g(r,B) + G(r,T^-)$, where$T^- = Inverse(T) = \{t | t \in T, t.w^- = -1 * |t.w|\}$. We use $t.w^-$ to replace $t.w$ when calculating $f_{max}(r)$. Algorithm 4 attached in the Appendix (d) implements the calculation of $f_{max}(r,R,T)$.

4.4 Max Coverage Gain Mining: MCGminer In this part, we propose the main algorithm for MCG mining. Algorithm 5 attached in the Appendix (d) describes the implementation of the mining procedure. The mining of MCG follows three phases as below:

– *Phase 1: Data Preparation*. This task includes the building of transaction blocks and the generation of adjacent matrix.

– *Phase 2: Pre-pruning*. In this phase, plenty of redundant rules are eliminated. Meanwhile, some core rules are discovered and pushed into $\widehat{R}$. By detaching these rules from the whole rule set R, the adjacent matrix M is simplified. As a result, the searching space is considerably reduced, which is also verified in experiments.

– *Phase 3: Iteration and Island Splitting*. For each big island, the corresponding hinge set is identified and used to build an iteration tree, which decomposes the big island into multiple smaller islands that are easy to be processed.

5 Experimental Evaluation

In this section, several experiments are performed on extensive UCI data sets and an online banking transaction database to show the effectiveness, efficiency and scalability of our proposed measure *MCG* and its corresponding algorithm *MCGminer*. The algorithm and mechanisms have been deployed into an online banking risk management system i-Alertor [1] installed in a major Australian bank.

5.1 Baseline Settings The baselines we choose are four typical rule-based methods: *C4.5* [8], *CBA* [4], *CMAR* [5] and *CAEP* [6]. Before implementing the task of prediction, all the methods are adjusted to follow a two-stage procedure, i.e., rule generation and rule selection. To test the rule selection difference made by following *MCGminer*, we keep the first step of these four methods unchanged and replace their original rule selection modules with our proposed $MCGminer$, and denote them as *C4.5**, *CBA**, *CMAR** and *CAEP**, respectively. We then compare the accuracy of the adapted methods with their respective original ones. Furthermore, we also test the effectiveness of pruning strategies proposed in Section 4. According to the process of island splitting in Section 4, there are two key factors to determine the enumeration cost: *splitting gap* and *hinge scale*. Thus the cost of enumeration process will be measured by splitting gap and hinge scale. Finally, computational cost evaluation is conducted, in which we propose a benchmark algorithm *GA*MCG* that applies the genetic algorithm to calculate *MCG* with convergence to the globally optimal result, which obtains the accurate rule set by keeping the local optimal of current generation to the next one [10].

5.2 Data Sets Two categories of data, including 13 UCI data sets and an Online Banking (*OB*) transaction database, are used in our experiments. *OB* is provided by a major bank in Australia, it has two class labels: fraud and genuine. There are 1,251 Fraud out of 3,200,000 Genuine; 150 features are involved, 102 of them are numerical and the rest 48 features are nominal. For UCI data sets, we simply assign 1 to be the gain value for targeted class, and -1 for the rest. As for the online banking transactions, the gain values for all false positive alerts are $-100 (as mentioned in the task description

Table 4: The Improvement on Accuracy Evaluation

Data Set	Original				Improved by MCG				Average
	$C4.5$	CBA	$CMAR$	$CAEP$	$C4.5^*$	CBA^*	$CMAR^*$	$CEAP^*$	Improvement
Australian	86.1%	85.4%	85.9%	86.2%	87.1%	86.5%	87.1%	89.4%	1.89%
Cleve	78.2%	82.8%	82.2%	83.2%	79.5%	87.1%	86.2%	88.7%	4.58%
German	72.4%	73.8%	74.9%	72.5%	75.7%	74.4%	76.3%	75.1%	2.70%
Heart	80.1%	81.9%	82.1%	83.8%	82.2%	82.8%	83.7%	85.1%	1.81%
Hepatitis	80%	80.2%	80.2%	83.1%	83.3%	82.5%	82.2%	84.3%	2.73%
Iono	90%	92.1%	91.5%	91%	91.4%	93.5%	92.5%	93%	1.59%
Iris	95.3%	94.6%	94%	94.6%	95.5%	94.1%	95.2%	94.8%	0.29%
Labor	79.2%	86.3%	89.7%	88.6%	80.2%	87.3%	91.9%	90.6%	1.78%
Pima	75.5%	72.9%	75.1%	75%	75.7%	73.2%	77.1%	79%	2.2%
Sonar	70.2%	77.5%	79.4%	79.9%	71.2%	78.6%	79.8%	79.7%	0.77%
Vehicle	72.6%	68.7%	68.8%	66.3%	73.7%	69.7%	69.9%	67.9%	1.75%
Waveform	78.1%	80.1%	83.2%	84.6%	79.4%	83.3%	84.2%	86.1%	2.16%
Wine	92.7%	95%	95%	96.1%	92.8%	96.4%	95.2%	96.7%	0.6%
OB_1 ($\mu = 1,000$)	45.2%	47.6%	53.5%	63.7%	51%	54.3%	58.4%	68.9%	11.06%
OB_2 ($\mu = 10,000$)	40.7%	41.1%	46.2%	46.1%	49.0%	51.3%	53.3%	63.0%	24.3%
OB_3 ($\mu = 100,000$)	35.6%	35.6%	36.1%	37.8%	46.7%	47.3%	49.4%	57.0%	37.92%

of Figure 1), while the true positive alerts take the dollar value loss for individual transaction.

5.3 Accuracy Evaluation The accuracy evaluation is performed on 13 UCI data sets and an online banking transaction database. Three transaction subsets (i.e., OB_1, OB_2 and OB_3) are generated by random sampling with class-imbalanced rates $\mu = 1000, 10000, 100000$, respectively. The class-imbalanced rate is defined as the ratio between the number of Genuine and the number of Fraud.

As shown in Table 4, *MCGminer* enhances the accuracy among all the data sets, and the last column presents the average improvement for each data set. For online banking transactions, we can observe a drastic improvement on accuracy, ranging from 5.65% to 26 %. As μ increases, the improvement by *MCG* on accuracy becomes larger and larger. More importantly, the prediction accuracy on OB_1, OB_2 and OB_3 with all the methods incorporating *MCG* is far less sensitive to the imbalance rate μ than that with their original counterparts. The averaged accuracy decrease ratio of the original methods on data *OB* in terms of lower to higher imbalance rate is 16.33%, while the corresponding decrease ratio of the methods incorporating *MCG* is only 7.00%.

5.4 Stability of Detection Rate Against Imbalance Rate In this part, we study the stability of *MCG* against the imbalance rate μ in real-time prediction on *OB*. We use Detection Rate (DR) (i.e. DR is the percentage of the Fraud caught among the total Fraud) to evaluate the prediction capability of each method under different values of μ. As shown in Figure 5, with the increase of the imbalance rate μ, DR decreases in all original methods (i.e., *C4.5*, *CBA*, *CMAR* and *CAPE*) with different trends. However, the corresponding *MCG*-driven classifiers all exhibit a rather stable curve in terms of DR, compared to the original classifiers.

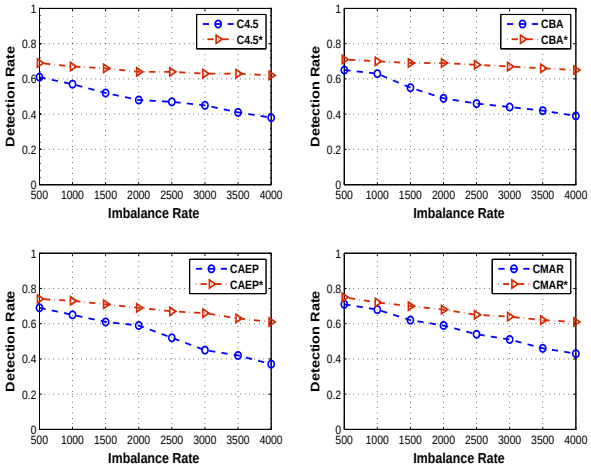

Figure 5: Hinge scale test.

5.5 Effectiveness of Pruning Strategies Below, we test the scalability of *MCG* on large data set *OB* by checking the distribution of the *splitting gap* and *hinge scale*. The more balance of the sub-islands the less computational cost by the enumeration process. Thus, we calculate the statistics in terms of the frequency of different gap values and hinge scale among all rounds of the splitting operation. In order to get stable results, we assemble six groups of rule sets (i.e., R_1, R_2, R_3, R_4, R_5 and R_6) by random sampling with replacement. As a result, Figure 6(a) shows that the frequency distribution of different hinge scale for six groups. The frequency reaches the highest value at around $gap = 3$, and then drops rapidly for $gap > 3$.

5.6 Scalability on Number of Rules Another important factor that determines the computational cost for MCG mining is the rule number in R. Since Genetic Algorithm

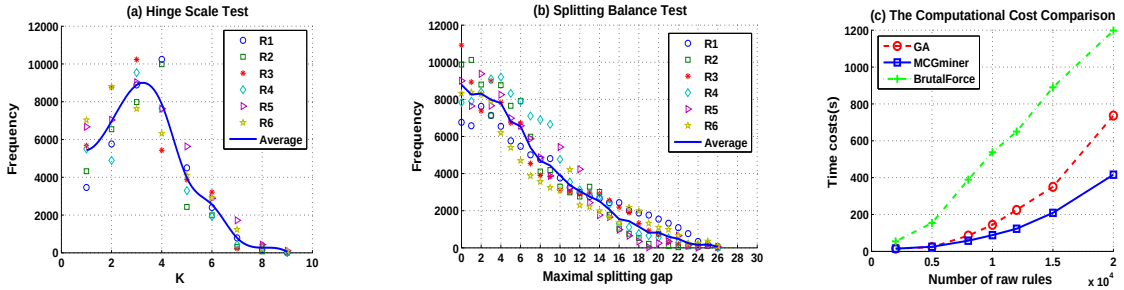

Figure 6: Hinge scale test, splitting balance test and the computational cost comparison.

(GA) is frequently used for solving high complexity mining problems, we build a GA-based algorithm that converges to global optimal result [10] and a brutal-force program to compare with *MCGminer* in terms of computational cost against various data scales. As shown in Figure 6(c), to achieve the same detection rate, the computational cost of $MCGminer$ is slightly better than that of GA when the number of rules is smaller than 100,000. When the number of rules further increases, $MCGminer$ gains even better advantage of computational cost (i.e., $MCGminer$ only costs less than 60% of the running time of GA when the number of rules approaches 150,000). Besides, the brutal-force program costs much more time than *GA* and *MCGminer* with any number of rules.

According to Section 4.4, the splitting gap is the size difference between the maximal island and the minimal one, which is decisive to the depth of enumeration process. Therefore, a small gap costs less time. Figure 6(b) presents the distribution of splitting gap for all the splitting operations. We can see that the frequency drops quickly on all six sample groups and it keeps as a small value when $gap > 15$.

6 Conclusions

The effective selection of optimal rules for detecting anomalies in a large scale (say >1,000,000) of highly imbalanced data (say >60000:1) is a crucial and challenging issue in developing rule-based systems for the real-life big data analytics. The existing approaches cannot be deployed to address this issue effectively and cannot deliver the global optimization results. In this paper, we have proposed a novel metric, *Maximal Coverage Gain* (*MCG*), to select the globally optimal rule set from a large number of generated rules. MCG guarantees the optimal prediction capability, especially in a cost-sensitive way. A collection of built-in mechanisms including rule interaction, hinge set, gain bounding and pruning strategies are developed and incorporated into an efficient algorithm i.e., *MCGminer*, to effectively mine *MCG*. Substantial experiments show that MCGminer and the classifiers built with our proposed metric and mechanisms dramatically outperform the typical existing baseline methods in tackling large imbalanced data in terms of accuracy, scalability, stability and efficiency. The algorithm and mechanisms have been successfully deployed into an online banking risk management system i-Alertor for a major Australian bank. We project that the computational cost and system performance can be highly updated further.

Acknowledgments The work is funded by the Australian Research Council Discovery grant (DP1096218) and Linkage grant (LP100200774).

References

[1] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, *Effective detection of sophisticated online banking fraud on extremely imbalanced data*, World Wide Web, (2012), pp. 1-27.

[2] D. Gibson, J. Kleinberg, and P. Raghavan, *Building an associative classifier based on fuzzy association rules*, Computational Intelligence Systems, 1 (2008), pp. 262-273.

[3] F. Pach, and A. Gyenesei, *Compact fuzzy association rule-based classifier*, *Expert System*, 34 (2008), pp. 2406-2416.

[4] B. Liu, W. Hsu, and Y. Ma, *Integrating classification and association rule mining*, in KDD, 1998.

[5] W. Li, J. Han, and J. Pei, *CMAR: Accurate and efficient classification based on multiple class-association rules*, in ICDM, pp. 369-376, 2001.

[6] G. Dong, X. Zhang, L. Wong, and J. Li, *CAEP: Classification by aggregating emerging patterns*, in LNCS, 1999.

[7] L. Cao, Y. Ou, and P. S. Yu, *Coupled behavior analysis with applications*, *IEEE TKDE*, 24 (2011), pp. 1378-1392.

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

[9] J. Alcal, R. Alcal, and F. Herrera, *A fuzzy association rule based classification model for high-dimensional problems with genetic rule selection and lateral tuning*, IEEE Transactions on Fuzzy Systems, 19 (2011), pp. 857-872.

[10] R. F. . Hartl, *A global convergence proof for a class of genetic algorithms*, manuscript (Vienna University of Technology), 1991.