Statistically Significant Dependencies for Spatial Co-location Pattern Mining and Classification Association Rule Discovery

by

Jundong Li

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

Abstract

Spatial co-location pattern mining and classification association rule discovery are two canonical tasks studied in the data mining community. Both of them focus on the detection of sets of features that show associations. The difference is that in spatial co-location pattern mining, the features are all spatial features which contain location information. While in classification association rule discovery, we constrain the mining process to generate association rules that always have as consequent a class label. Existing methods on these two tasks mostly use the support-confidence framework in an Apriori-like way or through a FP-growth approach to mine the co-location patterns and classification association rules which require the setting of confounding parameters. However, the lack of statistical dependencies between features in the used framework may lead to the omission of many interesting patterns and/or the detection of meaningless rules.

To address the above limitations, we fully exploit the property of statistical significance and propose two novel algorithms for these two tasks, respectively. The CMCStatApriori, a co-location mining algorithm, is able to detect more general and statistically significant co-location rules. We use it on real datasets with the National Pollutant Release Inventory (NPRI), and propose a classification scheme to help evaluate the discovered co-location rules. The second algorithm, SigDirect, an associative classifier, aims to mine classification association rules which show statistically significant dependencies between a set of antecedent features and a class label. Experimental results on UCI datasets show that SigDirect achieves a competitive if not better classification performance while indeed produces a very small number of rules. We also show the potential of integrating statistically significant negative classification association rules in the SigDirect algorithm.

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Publications

Conference Papers:

1. Jundong Li, Osmar R. Zaïane, and Alvaro Osornio-Vargas. Discovering Statistically Significant Co-location Rules in Datasets with Extended Spatial Objects. In *Proceedings of the 16th International Conference on Data Warehousing and Knowledge Discovery (DaWaK), 2014.*

2. Jundong Li, Osmar R. Zaïane. An Associative Classifier with Statistically Significant Positive and Negative Rules. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM), 2014.* (Submitted)

3. Jundong Li, Osmar R. Zaïane. SigDirect: Statistically Significant Dependent Classification Association Rules for Classification. In *Proceedings of the 14th IEEE International Conference on Data Mining (ICDM), 2014.* (Submitted)

Journal Papers:

1. Jundong Li, Aibek Adilmagambetov, Osmar R. Zaïane, Alvaro Osornio-Vargas and Osnat Wine. On Discovering Co-Location Patterns in Datasets: A Case Study of Pollutants and Child Cancers. *International Journal of Geoinformatica*, 2014. (Submitted)

Book Chapters:

1. Luiza Antonie, Jundong Li, Osmar R. Zaïane. Negative Association Rules. In *Frequent Pattern Mining (edited by Charu C. Aggarwal, Jiawei Han), Springer, 2014.*

Chapter 1

Introduction

1.1 Motivation

The concept of association rule mining was first introduced by Agrawal et al. [5] and extensively studied in the past decades [6, 32] which aims to find associations between items or itemsets in a transaction database. Assume a transaction database $\mathcal{D}$ consists of a set of items $\mathcal{I} = \{i_1, i_2, ..., i_m\}$, then an association rule is an implication of the form "$X \rightarrow Y(support, confidence)$", where X and Y are disjoint subsets of $\mathcal{I}$. The support indicates the probability that X and Y appear together in the transaction database $\mathcal{D}$. The strength of the rule is measured by confidence, which is the conditional probability of Y given X. The problem of discovering association rules in transaction database $\mathcal{D}$ consists of generating the rules that have a support and a confidence higher than given thresholds.

Spatial co-location mining, one of the canonical tasks of spatial data mining, could be seen as an extension of association rule mining. The task is very similar to association rule mining, the major difference being that in spatial co-location pattern mining, the features are all spatial features which contain location information. It tries to find a set of spatial features that are frequently co-located together, i.e. in a geographic proximity. A motivating application example is the detection of possible co-location rules between chemical pollutants and cancer cases with children. Due to the absence of a clear notion of transactions, it is nontrivial to use association rule mining techniques to tackle the co-location rule mining problem directly. Therefore, previous work [49, 61, 64, 63, 33] are mainly based on transaction-free

apriori-like algorithms. A prevalence measure threshold is required in the property of anti-monotonicity for effective pruning, the strength of co-location rules are determined afterwards with a minimum confidence threshold. However, the support-confidence framework fails to capture the statistical dependency between spatial features. On one hand, the antecedent and consequent spatial features may be independent of each other. On the other hand, some other strong dependent co-location rules may be ignored due to a low prevalence measure value. In the worst case, all detected co-location rules can be spurious, and strong co-location rules are totally missing. These two types of scenarios are called *type 1* (false positive) error and *type 2* (false negative) error. Another limitation of transaction-free apriori-like co-location mining algorithms is that they use only one distance threshold to determine the neighbourhood relationship. However, in real applications, a proper distance threshold is hard to determine. Meanwhile, with only one distance threshold, the neighbourhood relationship among spatial features can not be fully captured.

To solve the previous mentioned limitations of transaction-free apriori-like co-location mining algorithms, Adilmagambetov et al. [2] proposed a new transaction-based framework to discover co-location rules in datasets with extended spatial objects. Buffers are built around each spatial object, the buffer zone could be the same for each spatial object or it might be affected by some other spatial or non-spatial features, like the amount of chemical pollutants the facility emits, wind direction in this region, etc. Then, grids are imposed over the geographic space; each grid point intersects with a set of spatial objects could be seen as a transaction. As mentioned above, the usage of support-confidence framework may result in the discovery of weak co-location rules and the omission of strong co-location rules. Therefore, to find statistically significant co-location rules, a statistical test method is used instead of global thresholds. However, the statistical significance is not a monotonic property and it can not be used to prune insignificant rules as apriori-like algorithms. Thus in their work, they limit the size of the antecedent of a rule up to three features and test each possible candidate co-location rule to see if it passes a statistical test. The algorithm cannot scale up well for co-location rules with more than three

spatial features in the antecedent, and as such limits its use. Therefore, the first motivation of this thesis is to derive effective algorithms to detect more general and statistically significant co-location rules.

The other problem which is of our interest is introducing statistical dependencies into the problem of classification association rule discovery. Classification is another canonical task in the data mining and machine learning community. Given a set of attributes for an object, a classifier tries to assign it to one or more predefined classes. A typical classification method consists of two steps, the first step is to build a model on the training set whose attributes and class labels are known in advance. Then the ability of the model to correctly classify objects in the test dataset is evaluated.

Recent studies on associative classification integrate association rule mining and classification [40, 38, 7, 13]. These associative classifiers have proven to achieve competitive classification accuracies as decision trees [47], rule induction methods [46, 22], naïve-Bayes [26] as well as some probabilistic methods [39]. Besides, instead of taking a greedy algorithm as most rule based classifiers, associative classification directly mines the complete set of classification association rules (CARs) to avoid missing any important ones. Another advantage of associative classification is that each individual CAR in the model is human readable and can be interpreted. To classify an object, associative classifiers first adopt association rule mining methods to mine the CARs in the form of $X \rightarrow C$ with given support-confidence thresholds and constrain the consequent of the rule to be a class label. Then a subset of CARs after pruning are selected to form the classifier. The selection is usually made by utilizing the database coverage method [40]. Finally, once the classifier is built, it chooses one or more matching CARs to make predictions on the test dataset.

The existing associative classification methods mine the complete set of CARs mostly in an apriori-like fashion [6] or through a FP-growth way [32]. Although the rule generation process might be slightly different, all of them use a support-confidence framework to find CARs for classification. Therefore, they have the same shortcomings as most spatial co-location mining algorithms: first, it is diffi-

Table 1.1: An example of *type 1* error and *type 2* error.

Items	Class Label	Frequency
x	c_1	4400
y	c_2	15
z	c_2	5480
x, y	c_1	20
y, z	c_2	80
x, y, z	c_2	5

cult to determine the appropriate support and confidence thresholds for each dataset without any background information; second, the traditional association rule mining methods are based on frequency to prune infrequent patterns. The strength of a rule is decided afterwards with a confidence value. In this way, CARs do not capture the actual statistical dependencies between features and the corresponding classes. In the worst cases, we may only find spurious CARs while leaving statistically significant CARs undiscovered. Table 1.1 shows an example of these two types of errors.

Example 1. *A transaction dataset is shown in Table 1.1. Let the* min_support $= 1\%$ *and* min_confidence $= 50\%$*, respectively. On one hand, through an Apriori-like or a FP-growth method, we generate some CARs. The CAR:* $\mathbf{y} \rightarrow \mathbf{c_2}$ *is among them because its support is 1% and confidence is around 83% which meets the support-confidence thresholds requirement. Although the confidence of the CAR is very high, there is a very weak dependency between* $\mathbf{y}$ *and* $\mathbf{c_2}$*, because the support of* $\mathbf{c_2}$ *is much larger than the support of* $\mathbf{y}$*, i.e.* $5580 \gg 120$*. In other words,* $\mathbf{y}$ *might happen to appear together with* $\mathbf{c_2}$*, and in fact, they are more likely to be independent of each other. This is a typical example of type 1 error. On the other hand, it misses a strong CAR:* $(\mathbf{x}, \mathbf{y}, \mathbf{z}) \rightarrow \mathbf{c_2}$*. The CAR is not found because it has a very low support, 0.05%. But the confidence of the CAR is 100%. Besides, itemsets* $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ *always co-occurs with* $\mathbf{c_2}$ *which demonstrates that it is a CAR with strong dependency and the missing of this CAR is considered as an example of type 2 error.*

To avoid missing any strong CARs, most associative classifiers maintain a small

Table 1.2: CARs example ($minsup = 10\%$, $minconf = 90\%$).

CARs	confidence
$X \rightarrow c_1$	95%
$Y \rightarrow c_1$	90%
$XY \rightarrow \neg c_1$	92%
$X \rightarrow c_2$	90%
$Y \rightarrow c_2$	95%
$Z \rightarrow c_1$	95%
$Z \rightarrow c_2$	93%

minimum support threshold, but it is still of a high chance to miss strong CARs and at the same time it introduces a new problem: association rule mining ends up generating a huge number of CARs making it impossible to manually edit and even defeating the readability of the classification model. From another perspective, some post-processing strategies have been proposed to alleviate the *type 1* error [38, 11, 20], but the discovered CARs are still not statistically significant.

Searching for statistically significant CARs is a demanding and intractable task because statistical dependency is not a monotonic property, therefore, we cannot do some effective pruning like Apriori or FP-growth approaches. The situation gets worse especially if we have a large number of features since the number of candidate CARs grows exponentially with the number of features.

Another concern on associative classification algorithms is that most of them only use positive CARs ($X \rightarrow C$) in the classification process. By positive associations we refer to associations between items and classes existing in the transaction database. In addition to the positive CARs ($X \rightarrow C$), the negative CARs are also able to provide valuable information to discriminate between different classes. A negative CAR is in one of the following forms: $X \rightarrow \neg C$ or $\neg X \rightarrow C$ (where X means existence and $\neg X$ means absence of feature X in enough transactions). Let us consider the following example:

Example 2. *In a transaction database $\mathcal{D}$, we discover some positive and negative CARs as illustrated in Table 1.2. Assume now we have a new instance with items X and Y, how to classify the new instance? When only positive CARs are considered, it is hard to label the new instance because both class c_1 and class c_2 are possible.*

For class c_1 and class c_2, we both find two matching rules with a 95% and a 90% confidence value, respectively. But the problem is easier since we have a negative CAR $XY \rightarrow \neg c_1$ which reinforces the decision in favour of class c_2.

The above two examples point out the limitations of current association classification methods and potential direction to improve these algorithms. Firstly, we aim to find an effective and efficient way to mine statistically significant CARs, and build a novel associative classification algorithm based on the discovered CARs. Secondly, we will investigate if we can also introduce negative statistically significant CARs into the new classifier to help the classification process.

1.2 Thesis Statement

In this thesis we address the following two challenges: mining statistically significant co-location rules and building an associative classifier with statistically significant classification association rules. We study the feasibility of resolving these challenges by claiming the following statements:

1. In a spatial dataset with spatial features of pollutants and cancers, it is possible to detect more general and statistically significant co-location rules without randomization test methods in which the size of the co-location rules is constrained to be below a certain threshold.

2. It is possible to find an effective way to detect statistically significant classification association rules without any confounding parameter settings and we can use the discovered classification association rules to build an associative classifier.

3. Negative classification association rules may be able to provide valuable information to discriminate between different classes, therefore, it is also possible to integrate negative statistically significant classification association rules to the associative classifier to reinforce the classification process.

1.3 Thesis Contribution

This thesis makes the following contributions:

1. In this study, we first investigate how to exploit the property of statistical significance to scale it up to detect more general co-location rules without any randomization tests. We propose a new algorithm: Co-location Mining Constrained StatApriori (CMCStatApriori) which is able to detect statistically significant co-location rules without any limitation on the rule size. CMC-StatApriori is based on the work of StatApriori [28, 30]. It uses the z-score to search for the statistically significant co-location rules with a fixed consequent spatial feature. The results of co-location rules are hard to evaluate even for domain experts, therefore, we also propose to use a classifier to help evaluate the results of the discovered co-location rules.

2. Second, we propose a novel parameter-free associative classifier, SigDirect. An extension of statistically significant dependency rule mining method, Constrained Kingfisher, is proposed on the basis of Kingfisher [29, 31], it pushes the rule shape constraint in the phase of generation of classification association rules by taking Fisher's exact test as a significance measure along with some effective pruning strategies. An instance centric rule pruning strategy is used to find a globally optimal CAR for each instance in the training dataset. In the classification phase, we conduct an empirical study on different classification heuristics and investigate how to combine the class predictions of selected rules to make a final classification model. The proposed classifier achieves competitive and even better classification performance as some well-known rule based classifiers and the state-of-the-art associative classifiers while generates an order of magnitude less classification association rules.

3. Third, we propose to integrate the negative statistically significant classification association rules into the SigDirect algorithm which shows a promising classification result compared to the classifier built with only positive statis-

tically significant classification association rules, a simple but effective rule pruning strategy is presented to prune noisy classification association rules without jeopardizing the classification accuracy.

1.4 Thesis Outline

1. Chapter 2 reviews the most important studies in the area of spatial co-location mining, associative classification, statistically significant rule mining and negative association rule mining. Concretely, Section 2.1 briefly describes the task of spatial co-location pattern mining and reviews recent studies from two perspectives: the support-confidence framework and the statistical test framework. Section 2.2 introduces the procedure of building an associative classifier and overviews existing associative classifiers. Section 2.3 covers the recent studies on statistically significant rule mining. Finally, Section 2.4 gives a summary of the most important negative association rule mining algorithms.

2. Chapter 3 addresses the problem of mining general and statistically significant co-location rules. In this chapter, Section 3.1 explains the problem definition. Section 3.2 introduces the proposed algorithm, Co-location Mining Constrained StatApriori (CMCStatApriori). Section 3.3 contains the description of the real spatial datasets with the National Pollutant Release Inventory (NPRI), experimental settings and the experimental results. Section 3.4 introduces a classifier we propose to help evaluate the discovered co-location rules.

3. Chapter 4 addresses the problem of mining statistically significant classification association rules for classification. In Section 4.1, some basic notations and definition are presented. Section 4.2 describes the process of generating statistically significant classification association rules with Constrained Kingfisher algorithm. Section 4.3 talks about the effects of the pruning strategies and the classification heuristics we take in the classifier. Section 4.4 introduces the empirical study on different classification heuristics and in-

vestigates how to combine the class predictions to make a final classification model. Section 4.5 shows the experimental results of the proposed SigDirect algorithm. Section 4.6 presents how to incorporate the negative classification association rules into the associative classifier and Section 4.7 shows a promising result when the negative classification association rules are also considered.

4. Chapter 5 summarizes the important conclusions of this thesis. In addition, we list some possible future work that can be done regarding the problems discussed in this study.

Chapter 2

Related Work

In this chapter we review some of the related work on spatial co-location mining, associative classification, statistically significant rule mining as well as negative association rule mining.

2.1 Spatial Co-location Pattern Mining

In this section, we review some related work about two different frameworks of co-location pattern mining algorithms: the support-confidence framework and the statistical test framework.

2.1.1 Support-Confidence based Co-location Pattern Mining

Spatial co-location mining has been extensively studied in the past decades. An initial summary of results on mining co-location patterns was proposed by Shekhar and Huang [49]. The algorithm they proposed is based on the neighbourhood relations between spatial features and the concept of participation index. The basic concept of this method is similar to the concepts of association rule mining. Due to the absence of explicit notion of transactions over the spatial dataset, they proposed to use neighbourhoods of instances of different spatial features to represent the groups of spatial items appearing together. For instance, if an instance of spatial feature A, an instance of spatial feature B and an instance of spatial feature C are in a spatial proximity of each other, a corresponding transaction is derived as $\{A, B, C\}$. Similar to association rule mining, the output co-location rule is

in the form of $C_1 \rightarrow C_2(PI, cp)$, where C_1 and C_2 are a set of spatial features. Prevalence measure *Participation Index* (*PI*) corresponds to the support and cp is the conditional probability that corresponds to the confidence. The co-location rule $C_1 \rightarrow C_2(PI, cp)$ is considered as prevalent, or interesting, if for each feature of the pattern at least $PI\%$ instances of that feature form a clique with the instances of all other features of the pattern according to the neighbourhood relationship. The candidate generation and pruning process works in a similar way like association rule mining methods. However, in the previous mentioned work, the authors assumed that different spatial features have a similar level of frequency, therefore, co-locations patterns with rare spatial features may be pruned due to a low prevalence value.

To address this problem, Huang et al. [35, 33] continued their previous work by introducing an algorithm that finds co-location patterns with rare spatial features. The authors proposed to use the *Maximal Participation Ratio* (*maxPR*) threshold to replace the *Participation Index* (*PI*) threshold. A co-location pattern C is considered prevalent or interesting if $maxPR(C)$ is greater than the minimum confidence threshold.

Based on the approach in [49], most of the following work tried to improve the running time of the mining procedure, [34] proposed a novel multi-resolution pruning technique and also showed that the PI has a spatial statistical interpretation that it acts as an upper bound for the cross-K function, which is often used as a statistical measure for spatial feature relationships.

Zhang et al. [67] proposed a fast co-location mining method combing the discovery of neighbourhood relationships with the mining process. Some heuristics are also proposed to optimize the mining process and to deal with constraint of memory. They extended a hash-based spatial join algorithm to identify neighbourhood relationships.

Yoo and Shekhar proposed a partial-join [65] and a join-less [63] algorithm to reduce the expensive computation time of the instance join. In the partial-join approach, the transaction is materialized to from a clique neighbourhood, while in the join-less approach, it tries to find star neighbourhoods instead of calculating

pairwise distances between spatial instances.

Xiao et al. [60] proposed a density-based approach to further improve the running time. They divided the spatial map into partitions and first identified instances in dense partitions. The method maintains a dynamic upper bound for the prevalence measure of candidate co-locations patterns, if the upper bound is less than a threshold, the candidate co-location pattern can be pruned.

Xiong et al. [61] extended the co-location pattern mining problem by introducing a framework for detecting co-location patterns in datasets with extended spatial objects. Extended spatial objects include points, lines and polygons. They proposed to build a buffer zone around each extended spatial object. The size of the buffer depends on size of the spatial object. *Coverage Ratio* is used as a prevalence measure, more specifically, if the area covered by the features of a candidate co-location pattern exceeds the *Coverage Ratio* threshold, the co-location pattern is considered to be interesting.

2.1.2 Statistical Test based Co-location Pattern Mining

The approaches mentioned above all use thresholds on prevalence measures, which causes meaningless co-location patterns to be considered as significant with a low threshold, and a high threshold may prune interesting but rare co-location patterns. Instead of using a threshold-based approach, Barua and Sander [14, 15] used the statistical test to mine statistically significant co-location patterns. Like [35, 33], the *Participation Index* is used as a prevalence measure, but the difference is that they did not set a prevalence threshold. Instead, for each candidate co-location pattern, the authors computed the probability p of seeing the same or greater value of the observed prevalence measure value under a null hypothesis model. In the null hypothesis model, the spatial features are assumed to be independent of each other. The candidate co-location pattern is considered statistically significant if $p \leq \alpha$, where α is a level of significance and is usually set to be 0.05.

Adimagambetov et al. [2] proposed a transactionization framework to find statistically significant co-location rules on extended spatial objects. In their problem, the spatial features are also linked to some other non-spatial features, for instance,

the spatial feature of pollutant has different amounts of release, and may be affected by wind direction. It is also important to mention that the consequent spatial feature in the co-location rule is fixed. They transformed spatial instances into transactions by buffers and grid points on the space. The expected support is used as a prevalence measure. The statistical test method they used is similar to [14, 15], in the null hypothesis model (randomization test), both the antecedent and consequent spatial features are randomized.

2.2 Associative Classification

The first reference to using association rules as CARs is credited to [17], while the first associative classifier, CBA, was introduced by Liu et al [40]. The main steps in building an associative classifier are as follows:

- Modeling the data into the transaction dataset $\mathcal{D}$ in which the numerical attributes are transformed to the discrete attributes.

- Generating the set of CARs from the transaction dataset $\mathcal{D}$. The CARs are in the form of $X \rightarrow C$ where X is a conjunction of attributes and C is a class label. The CARs are usually generated by pushing the constraint in the association rule mining process to generate association rules that always have as a consequent a class label given minimum support and minimum confidence thresholds.

- Pruning the discovered CARs by some rule pruning strategies. Previous rule generation phase usually generate an overwhelming number of CARs including many noisy CARs and it is very important to prune the rules to make the classifier effective and more efficient. The phase is employed to choose a best subset of CARs and weed out those rules that may introduce errors or are overfitting in the classification stage.

- Classifying a new unlabeled object to a predefined class. At this level a system that can make a prediction for a new object is built. The challenge here is

how to rank and make use of the set of rules from the previous phase to give a good prediction.

CBA mines the complete set of CARs through an apriori-like way, in addition, it ignores rules by "pessimistic error rate" as C4.5 [47]. In the rule pruning phase, CBA uses a method called "database coverage". Database coverage consists of going over all the rules ranked by confidence and evaluate them against the training instances. Whenever a rule applies correctly on some instances, the rule is marked and the instances eliminated until all training instances are covered. Finally, the unmarked rules are simply pruned. New instances are classified by the matching rule with highest ranking.

Motivated by the idea of CBA, many improvements have been proposed to build more accurate associative classifiers. CMAR [38] maintains a CR-tree to compactly store and retrieve rules, the CARs are discovered by a FP-growth approach. In addition to the database coverage method, CMAR also prunes lower ranked and more specific rules. The rule R_1: $X \rightarrow C$ with confidence $conf_1$ is a lower ranked and more specific rule w.r.t rule R_2: $X' \rightarrow C$ with confidence $conf_2$ if $X' \subsetneq X$ and $conf_1 \leq conf_2$. For a new unlabeled instance, CMAR makes a prediction based on multiple matching rules with a weighted χ^2 measure.

In the classification phase, ARC [7] takes all rules that apply within a confidence range, but instead, calculates the average confidence for each set of rules grouped by class labels in the consequent and selects the class label of the group with the highest confidence average.

There are some other variants for associative classification: Harmony [55] adopts an instance-centric approach to find the highest confidence rule for each training instance and builds the classification model from the union of these rules. In Harmony, some efficient pruning methods are employed to accelerate the rule discovery process in which the pruning strategies are incorporated within the FP-growth algorithm. Therefore, Harmony has both high efficiency and good scalability.

One deficiency of associative classification methods is the use of rules in the classification stage, some choose the best matching rule and some others make a prediction decision based on multiple matching rules, 2SARC [9] is a two-stage

classification model that is able to automatically learn to select the rules for classification. In the first stage, they use traditional association rule mining methods to mine CARs. Multiple predefined features which are outputs of the rules for associative classifier are then be used as input for a neural network, to train a new prediction model by weighing the different input features to train a more accurate classifier.

CCCS [11] uses a new measure, "Complement Class Support (CCS)" to mine positively correlated CARs to tackle the imbalanced classification problem. It forces the CCS to be monotonic, thus the complete set of CARs are discovered by a row enumeration algorithm. A associative classifier is then built upon these positively correlated CARs.

SPAR-CCC [53] is another associative classifier designed for imbalanced data. It integrates a new measure, "Class Correlation Ratio (CCR)" into the statistically significant rules, the classifier works comparably on balanced dataset and outperforms other associative classifiers on imbalanced dataset.

ARC-PAN [8] is the first associative classifier that uses both positive and negative CARs. The authors proposed to use the Person's ϕ correlation coefficient to mine generalized negative association rules. Based on the discovered positive and negative CARs, ARC-PAN gets a similar or even better classification result compared with other associative classifiers, it well demonstrates the potential of integrating negative association rules in the associative classification problem.

2.3 Statistically Significant Rule Mining

Searching for statistically significant rules with any consequent attribute is a very difficult problem, but finding statistically significant rules (with a fixed consequent) has been well studied. Most previous methods used chi-square as a statistical correlation measure and fully exploited the property of chi-square to prune uninteresting rules [43, 45, 44, 41, 13, 42]. However, these methods are mostly based on a minimum support threshold to mine rules and important post-processing work is required to get statistically significant rules. In spite of that, they still suffer from

the problem of missing significant rules (*type 2* error) heavily. Apart from the chi-square measure, many other measures like *lift, leverage* are proposed to express or validate if the antecedent and consequent part of a rule are independent of each other. Results, however, are not statistically significant.

Recently, researchers are focusing on searching for significant and non-redundant rules because the property of non-redundancy can be employed to do some effective pruning [37]. Another advantage of searching for non-redundant rules is that we can greatly reduce the search space since the pruned redundant rules do not add more information [66, 16].

Webb [56, 57, 58] has done a series of work on testing the significant and non-redundant rules with Fisher's exact test to avoid false discoveries. However, the tests can only be considered as a post-processing phase and the efficiency of the test is poor on high-dimensional dataset since the test space is extremely huge.

Hämäläinen et al. [28, 30, 29, 31] pioneered the development of statistically significant rule mining. In [28, 30], they proposed an algorithm, StatApriori, using z-score to mine the significant and non-redundant association rules. But the definition of redundancy in StatApriori is much more restrictive than the normal definition, therefore, they proposed another algorithm, Kingfisher [29, 31]. Kingfisher not only has a normal definition of redundancy, but also directly mines the global top-K statistically significant rules. Moreover, the discovered rules are able to show both positive and negative dependencies between antecedent and consequent items.

2.4 Negative Association Rule Mining

A negative association between two positive itemsets X, Y is a rule of the following forms: $\neg X \rightarrow Y$, $X \rightarrow \neg Y$ and $\neg X \rightarrow \neg Y$, where $\neg X$ and $\neg Y$ indicate the absence of itemsets X and Y in the transaction dataset $\mathcal{D}$. Mining association rules from a transactional dataset that contains information about both present and absent items is computationally expensive, traditional association rule mining algorithms cannot cope with mining rules when negative items are considered. This is the reason why new algorithms are needed to efficiently mine association rules with

negative items. Here we survey algorithms that efficiently mine some variety of negative association rules from data.

Brin et al. [18] mention for the first time the notion of negative relationships. They propose to use the chi-square test between two itemsets. The statistical test verifies the independence between the two itemsets. To determine the nature (positive or negative) of the relationship, a correlation metric is used.

Aggarwal and Yu [3, 4] introduce a new method for finding interesting itemsets in data. Their method is based on mining strongly collective itemsets. The collective strength of an itemset I is defined as $C(I) = \frac{1-v(I)}{1-E[v(I)]} \times \frac{E[v(I)]}{v(I)}$, $v(I)$ is the violation rate of an itemset I, i.e. the fraction of violations over the entire set of transactions and $E[v(i)]$ is its expected value. An itemset I is in a violation of a transaction if only a subset of its items appear in that transaction.The collective strength ranges from 0 to ∞, where a value of 0 means that the items are perfectly negatively correlated and a value of ∞ means that the items are perfectly positively correlated.

In [48], the authors present a new idea to mine strong negative rules. They combine positive frequent itemsets with domain knowledge in the form of a taxonomy to mine negative associations. The idea is to reduce the search space, by constraining the search to the positive patterns that pass the minimum support threshold. When all the positive itemsets are discovered, candidate negative itemsets are considered based on the taxonomy used.

Wu et al. [59] derive another algorithm for generating both positive and negative association rules. The negative associations discovered are of the following forms: $\neg X \rightarrow Y$, $X \rightarrow \neg Y$ and $\neg X \rightarrow \neg Y$. They add on top of the support-confidence framework another measure called *mininterest* for a better pruning of the frequent itemsets generated, which is used to assess the dependency between two itemsets.

The SRM algorithm [50, 51], discovers a subset of negative associations. The authors develope an algorithm to discover negative associations of the type $X \rightarrow \neg Y$. These association rules can be used to discover which items are substitutes for others in market basket analysis.

Antonie and Zaïane [10] propose an algorithm to mine strong positive and neg-

ative association rules based on the Person's ϕ correlation coefficient. In their algorithm, itemset and rule generation are combined and the relevant rules are generated on-the-fly while analyzing the correlations within each candidate itemset.

In [52], the authors extend an existing algorithm for association rule mining, GRD (generalized rule discovery), to include negative items in the rules discovered. The algorithm discovers top-K positive and negative rules.

Cornelis et al. [23] propose a new Apriori-based algorithm (PNAR) that exploits the upward closure property of negative association rules. With this upward closure property, valid positive and negative association rules can be discovered efficiently. Wang et al. [54] give a more intuitive way to express the validity of both positive and negative association rules, the mining process is very similar to PNAR.

MINR [36] is a method that uses Fisher's exact test to identify itemsets that do not occur together by chance, i.e. with a statistically significant probability. An itemset with a support greater than the positive chance threshold is considered for positive rule generation, while an itemset with a support less than the negative chance threshold is considered for negative rule generation.

Kingfisher [29, 31] is an algorithm developed to discover both positive and negative dependency rules. The dependency rule can be formulated on the basis of association rule and the statistical dependency of a rule can be calculated by the Fisher's exact test. In order to reduce the search space, the author introduces the basic branch-and-bound search with three lower bounds for the measure of p_F-value. Another two pruning strategies (pruning by minimality and pruning by principles of Lapis philosophorum) are also included to speed up the search.

Chapter 3

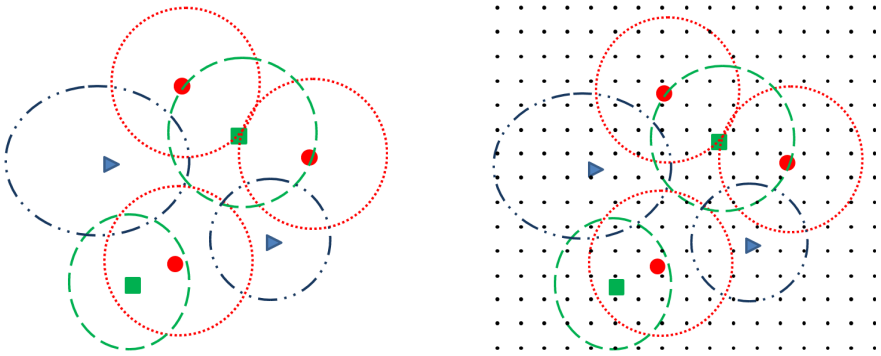
Statistically Significant Co-location Rule Mining

In this chapter, we will introduce the problem definition and the proposed method to mine statistically significant co-location rules. Experimental results and a novel evaluation method for the discovered co-location rules are also presented.

3.1 Problem Definition

The objective is to discover statistically significant co-location rules between a set of antecedent spatial features and a single fixed consequent spatial feature. A real world application of this task is to detect co-location rules between chemical pollutants (antecedent) and cancer cases or other morbidities (consequent). Since we do not intend to find the causality relationships, the goal is to identify potentially interesting co-location associations in order to state hypotheses for further study.

The task consists of three steps. In the initialization step, a buffer is built around each spatial object, and it defines the area affected by that object; for example, the buffer zone around an emission point shows the area polluted by a released chemical pollutant. The buffer shape is defined as circle in our problem, but the shape may also be affected by some other factors like wind direction. Considering the factor of wind direction, the circular buffer is transformed to elliptical. Figure 3.1(a) displays an example spatial dataset with buffers of various sizes (circular and elliptical) that are formed around spatial point objects. In the transactionization step, the transaction dataset is formed by imposing grids over all the buffer zones, as



(a) (b)

Figure 3.1: Transactionization step: (a) An example of spatial dataset with point feature instances and their buffers; (b) Grids imposed over the space.

shown in Figure 3.1(b). Then a transaction is defined as a set of spatial features corresponding to these objects [2]. After getting the derived transaction database $\mathcal{D}$ from the spatial dataset, we intend to detect statistically significant co-location rules in the statistical test step.

3.2 Algorithm

In this subsection, we introduce the proposed Co-location Mining Constrained Stat-Apriori (CMCStatApriori) algorithm which is able to detect statistically significant co-location rules without any rule length limitation.

CMCStatApriori is a variation of StatApriori [28, 30]; the main difference is that CMCStatApriori can efficiently detect more specific co-location rules, rules with one fixed consequent feature. Moreover, the non-redundancy definition in StatApriori is not very practical. In StatApriori, the association rule $X \rightarrow A$ (A is a single feature) is considered as redundant w.r.t. the rule $X' \rightarrow A'$, if $X'A' \subsetneq XA$ and the rule $X \rightarrow A$ is more significant than $X' \rightarrow A'$ under some interesting measure M. However, according to normal definition, the rule $X \rightarrow A$ is only considered as redundant w.r.t. the rule $X' \rightarrow A$, if $X' \subsetneq X$ and the rule $X \rightarrow A$ is more significant. In other words, to compare if one rule is considered as redundant to another rule, the consequent feature of both rules should be the same. Therefore, in CMCStatApriori, we do not intend to consider the non-redundancy.

For the co-location rule $X \rightarrow A$ ($F = \{f_1, ..., f_m\}$ is the set of spatial features

and $X \subsetneq F$, $A \in F$), the significance of dependency between X and A is compared with the null hypothesis in which X and A are independent. The statistical significance of the dependency is measured by the p-value, i.e. the probability of observing higher or equal frequency of X and A under null hypothesis. Suppose in the derived transaction dataset $\mathcal{D}$, each transaction can be viewed as an independent Bernoulli trial with two possible results, that $P(XA) = 1$ or $P(XA) = 0$. Thus, the statistical significance of the frequency of XA follows the binomial distribution and the p-value can be formulated as:

$$p = \sum_{i=\sigma(XA)}^{\sigma(A)} \binom{n}{i} (P(X)P(A))^i (1 - P(X)P(A))^{n-i} \tag{3.1}$$

where $\sigma(XA)$ is the observed frequency of XA, and n is the total number of transactions in $\mathcal{D}$.

The p-value is not a monotonic property, but z-score provides an upper bound for the critical value which corresponds to the p:

$$z(X \to A) = \frac{\sigma(XA) - \mu}{s} = \frac{\sqrt{nP(XA)(\gamma(XA) - 1)}}{\sqrt{\gamma(XA) - P(XA)}} \tag{3.2}$$

where $\mu = nP(X)P(A)$, $s = \sqrt{nP(X)P(A)(1 - P(X)P(A))}$ are the mean and standard deviation of the binomial distribution, respectively. $\gamma(XA) = \frac{P(XA)}{P(X)P(A)}$ is the lift for the co-location rule $X \to A$. It measures the strength of the dependency between X and A such that $\gamma(X \to A) > 1$ if X and A show a positive correlation. It is easy to notice that the z-score is a monotonically increasing function with the support and lift of XA: $\sigma(XA)$ and $\gamma(XA)$, therefore, it can be denoted as $z(X \to A) = f(\sigma(XA), \gamma(XA))$.

Therefore, following StatApriori [28, 30], the search problem can be reformulated as searching for all statistically significant co-location rules in the form of $X \to A$ with the following requirements (the set of statistically significant co-location rules is denoted as P):

Definition 1. *Statistically significant co-location rules*

1. *$X \to A$ expresses a positive correlation, i.e. $\gamma(X \to A) > 1$*

2. *for all* $(Y \to A) \notin P, z(X \to A) > z(Y \to A)$

3. $z(X \to A) \geq z_{min}$

With this definition, the property "potentially significant" (PS) is defined as follows. It is a necessary condition to construct the set of statistically significant co-location rules.

Definition 2. *Let A be the fixed consequent feature, z_{min} is an user-defined threshold for the z-score, and upperbound(f) be an upper bound for the function f. The co-location rule $X \to A$ is defined as potentially significant, i.e. $PS(X) = 1$, iff $upperbound(z(X \to A)) \geq z_{min}$. Otherwise, the co-location rule is not considered as statistically significant.*

The property of PS displays a monotonic property in some specific situations:

Theorem 1. *Let A be the fixed consequent feature and $PS(X) = 1$, then for all $Y \subseteq X$ and $min(XA) = min(YA)$ we can get $PS(Y) = 1$, where $min(XA)$ denotes the feature with the minimum support in XA.*

The proof of Theorem 1 is straightforward, first we can see that:

$$\gamma(YA) = \frac{P(YA)}{P(Y)P(A)} \leq \frac{1}{P(Y)} \leq \frac{1}{P(min(YA))} \tag{3.3}$$

where $min(YA)$ denotes the feature with the smallest support among YA. Since the z-score can be represented as a monotonically increasing function with two variables $\sigma(YA)$ and $\gamma(YA)$, the upper bound of the z-score for the co-location rule $Y \to A$ now is:

$$upperbound(z(Y \to A)) = f(P(YA), \frac{1}{P(min(YA))}) \tag{3.4}$$

then we have:

$$\begin{aligned} &upperbound(z(X \to A)) = f(P(XA), \frac{1}{P(min(XA)}) \leq \\ &f(P(YA), \frac{1}{P(min(YA)}) = upperbound(z(Y \to A)) \end{aligned} \tag{3.5}$$

for all $Y \subseteq X$ such that $min(XA) = min(YA)$. We can see that the monotonic property is kept only when the minimum feature (the feature with the minimal support) in XA and YA are the same.

With the monotonic property of PS, we can derive the algorithm that discovers the potential significant co-location rules in the same way as the general Apriori-like algorithms do, alternating between the candidate generation and candidate pruning. First, the set of antecedent features are arranged in an ascending order by their frequencies. Let the renamed features be $\{f'_1, f'_2, ..., f'_{m-1}\}$, where $P(f'_1) \leq P(f'_2) \leq ... \leq P(f'_{m-1})$. The candidate generation process is the same as that in Apriori [6], for the l-set $P_l = \{f'_{a_1}, ..., f'_{a_l}\}$ $(a_1 < a_2 < ... < a_l)$, we can generate $(l+1)$-sets $P_l \cup \{f'_{a_j}\}$, where $a_j > a_l$. After the generation of the $(l+1)$-sets $P_l \cup \{f'_{a_j}\}$, we need to check if all of its l-set "regular" parents (the parents with the same minimum support feature when combined with A as $P_l \cup \{f'_{a_j}\} \cup A$) can indicate PS co-location rules. If all of its regular parents can indicate PS co-location rules, then $P_l \cup \{f'_{a_j}\}$ is added to the candidate set for the pruning process, otherwise, $P_l \cup \{f'_{a_j}\}$ can be pruned directly. In the pruning process, the PS co-location rule $X \rightarrow A$ is kept if it meets the z_{min} threshold, otherwise, it is removed.

A problem of StatApriori is that for each potentially significant set C, only the best rule is derived from C. For example, if $C \backslash A \rightarrow A$ is the best rule, where $A \in C$ and the "best" indicates that the rule has the highest z-score, then no other rules in the form of $C \backslash B \rightarrow B (B \neq A)$ is output. However, in our CMCStatApriori algorithm, this kind of problem does not exist, because the PS property is for the co-location rule and the consequent feature is fixed. The detailed pseudo code of CMCStatApriori is illustrated in Algorithms 1, 2 and 3.

3.3 Experiments

3.3.1 Datasets

We conduct our experiments on two real datasets which contain pollutant emissions and information about cancer cases for children in the provinces of Alberta and Manitoba, Canada. The sources of the data are the National Pollutant Release

Data: Set of antecedent features $F \backslash A$, the consequent feature A, derived transaction database $\mathcal{D}$, the threshold z_{min} for the z-score
Result: Set of potential statistically significant co-location rules P
$P_1 = \{f_i \in F \backslash A | PS(f_i) = 1\}$;
l = 1;
while $(P_l \neq \varnothing)$ **do**
    $C_{l+1} = GenCands(P_l, A)$;
    $P_{l+1} = PrunCands(C_{l+1}, z_{min}, A)$;
    $l = l + 1$;
end
$P = \cup_i P_i$;
return P;

Algorithm 1: CMCStatApriori Algorithm.

Data: Potentially significant l-sets P_l, the consequent feature A
Result: $(l+1)$-candidates C_{l+1}
for *all* $Q_i, Q_j \in P_l$ *such that* $|Q_i \cap Q_j| = l - 1$ **do**
    if $\forall Z \subseteq Q_i \cup Q_j$ *such that* $|Z| = l$ *and* $min(ZA) = min((Q_i \cup Q_j)A)$ *and* $Z \subseteq P_l$ **then**
        $C_{l+1}.add(Q_i \cup Q_j)$;
    end
end
return C_{l+1}

Algorithm 2: Algorithm GenCands.

Data: l-candidates C_l, threshold z_{min}, the consequent feature A
Result: Potentially significant l-sets P_l
$\mathcal{S}_l = \varnothing$;
for *all* $Q_i \in C_l$ **do**
    Calculate $P(Q_iA)$ and the upperbound of lift [28, 30] $\frac{1}{P(min(Q_iA))}$;
    if $z(P(Q_i, A), \frac{1}{P(min(Q_i, A))}) \geq z_{min}$ **then**
        $\mathcal{S}_l.add(Q_i)$;
    end
end
return P_l

Algorithm 3: Algorithm PruneCands.

Inventory (NPRI) [19] and the provincial cancer registries. The information on pollutants is taken for the period between 2002 and 2007 and contains the type of a chemical, location of release, and average amount of release per year. In order to get reliable results, the chemical pollutants that had been emitted from less than three facilities are excluded from the dataset. There are 47 different chemical pollutants and 1,422 chemical pollutant emission points in Alberta; 26 different chemical pollutants and 545 chemical pollutant emission points in Manitoba, several chemical pollutants might be released from the same location. The number of cancer cases are 1,254 and 520 in Alberta and Manitoba, respectively. In order to make the model more accurate, the wind speed and direction are also taken into account in these two provinces. The interpolation of wind information between wind stations is used. In Alberta, the data of 18 stations are from Environmental Canada [24] and 156 stations are from ArgoClimatic Information Service (ACIS) [1]. In Manitoba, the data of all 20 stations are all from Environment Canada [24]. We obtain the wind direction and speed in the locations of chemical facilities by making interpolations in the ArcGIS tool [27].

3.3.2 Experimental Settings

We are interested in co-location rules of the form of $Pol \rightarrow Cancer$, where Pol is a set of pollutant features and $Cancer$ is a cancer feature. Three different methods are compared: the co-location mining algorithm by Adilmagambetov et al. in [2] (denoted as CM), co-location mining algorithm with Kingfisher [29, 31] (denoted as CMKingfisher) and the proposed CMCStatApriori method. In all of these three methods, the distance between grid points is 1km.

CM CM needs a number of simulations to detect significant co-location rules, the number of simulations for the statistical test is set to be 99 and the level of significance α is set to be 0.05. The size of antecedent features of a candidate rule is up to three. The randomized datasets (simulations) that are used in the statistical test are generated according to the distributions of chemical pollutant emitting facilities and cancer cases. Chemical pollutant emitting facilities are not randomly

distributed, and are usually located close to regions with high population density, thus, CM does not randomize the pollutant facilities all over the region, instead, it keeps locations of facilities and randomize the pollutants within these regions. For the cancer cases, most of them are located within dense "urban" regions and the rest are in "rural" regions. Therefore, the cancer cases are randomized according to the population ratio of "urban" regions to "rural" regions. In each simulation of CM, both pollutant chemicals and cancer cases are randomized.

CMKingfisher Kingfisher [29, 31] is developed to discover positive and negative dependency rules between a set of antecedent features and a single consequent feature. The algorithm is based on a branch and bound strategy to search for the best, non-redundant dependency top-K rules. Kingfisher is able to detect statistically significant positive and negative rules with any possible consequent. But we are only interested in the positive rules whose consequent is "Cancer", therefore, after getting the derived transaction dataset T, we apply Kingfisher algorithm to get the complete set of co-location rules and extract the subset of co-location rules that we are interested in. The significance level α is 0.05.

CMCStatApriori The CMCStatApriori is the algorithm proposed in this paper. Unlike CM and CMKingfisher which use the p-value as a significance level, CMStatApriori uses the z-score which provides an upper bound for the p-value. In the experiment, the threshold of z-score is set to be 150 in the Alberta dataset. This threshold of 150 is too high in the Manitoba dataset and no co-location rules are output. Therefore, we set a lower z-score threshold of 40. Indeed, the lower the z-score threshold, the more co-location rules is generated. The parameter setting of z-score threshold of CMCStatApriori is discussed in the end of this chapter.

3.3.3 Experimental Results

Both CMKingfisher and CMCStatApriori are able to detect more general co-location rules (without limitation of size of antecedent features). However, to have a fair comparison with CM, we only list the co-location rules with up to three antecedent

features. The number of rules detected by these three methods and the number of rules overlaps with CM by CMKingfisher as well as CMCStatApriori are listed in Table 3.1. It can be observed that in the dataset of Alberta, both of CMKingfisher and CMCStatApriori have a small overlap with CM rules. The situation is slightly different in the dataset of Manitoba, around 80% and 30% of detected rules by CMKingfisher and CMCStatApriori also appear in CM.

Table 3.1: Number of co-location rules generated by different methods.

	Alberta		Manitoba	
	#rules	# rules in CM	#rules	# rules in CM
CM	273	–	170	–
CMKingfisher	108	7	23	19
CMCStatApriori	571	5	60	16

3.4 Evaluation

Environmental pollutants are suspected to be one of the causes of cancer in children. However, there are other factors that could lead to this disease. Therefore, it is a difficult task to evaluate the detected co-location rules even for domain experts. To assist in evaluating the discovered co-location rules, we propose to use a classifier with the discovered co-location rules as a predictive model. Meanwhile, we also develop a tool to visualize the discovered co-location rules in the map. Figure 3.2 gives a snapshot of detailed information of all discovered co-location rules on Alberta dataset by CM. Each time you click a co-location pattern ID in Figure 3.2, a corresponding map will be displayed, it shows the locations and buffer zones of the pollutant features and the cancer feature, as shown in Figure 3.3. In Figure 3.4, we show a regional zoom in view of a discovered co-location rule.

The results by different methods are carefully and painstakingly evaluated manually by experts in our multidisciplinary team. However, the systematic evaluation by classification provides an estimation of the best quality co-location rule set.

In the classification evaluation scheme, we consider co-location rules generated by either method as a classifier. To evaluate the discovered co-location rules, we randomly sample some grid points on the geographic space. The randomly sampled

sup_pol	Expected support (Pollutants)
sup_pol_can	Expected support (Pollutants+Cancer)
conf_obs	Expected confidence (Pollutants -> Cancer)
count_pol_all	# of transactions that contain all pollutants
count_pol	# of transactions that contain pollutants at same facilities
count_pol_can	# of transactions that contain pollutants at same facilities + Cancer
p-value	probability of seeing the same or greater value of ExpConf in randomized datasets

conf_obs	color
[0.9-1.0]	
[0.8-0.9)	
[0.7-0.8)	
[0.6-0.7)	
[0.5-0.6)	
[0.4-0.5)	
[0.3-0.4)	
[0.2-0.3)	
[0.1-0.2)	
[0-0.1)	

pattern_Id	pol1	name1	pol2	name2	pol3	name3	sup_pol	sup_pol_can	conf_obs	count_pol_all	count_pol	count_pol_can	p-value
9	2a	Tetrachloroethylene					48.41	33.26	0.69				0.05
10	2b	Acetaldehyde					441.45	165.65	0.38				0.04
16	2b	Dichloromethane(methylene chloride)					268.36	168.35	0.63				0.02
29	3	2-Butoxyethanol					315.83	192.33	0.61				0.02
36	3	Isopropanol					615.59	268.10	0.44				0.02
38	3	4,4'-Methylenediphenyl diisocyanate					116.77	77.85	0.67				0.01
72	1	Benzene	3	Acrolein			14.28	12.45	0.87	389	278	90	0.02
164	1	1,3-Butadiene	3	2-Butoxyethanol			0.11	0.10	0.95	13	0	0	0.02
171	1	1,3-Butadiene	3	Isopropanol			0.01	0.01	0.96	73	0	0	0.02
173	1	1,3-Butadiene	3	4,4'-Methylenediphenyl diisocyanate			0.00	0.00	0.96	64	0	0	0.01
299	1	Arsenic and arsenic compounds	3	4,4'-Methylenediphenyl diisocyanate			2.59	2.47	0.95	12	0	0	0.03
339	1	Cadmium and cadmium compounds	3	4,4'-Methylenediphenyl diisocyanate			1.81	1.74	0.96	12	0	0	0.04
388	2a	Tetrachloroethylene	2b	Acetaldehyde			0.25	0.24	0.95	18	0	0	0.02
403	2a	Tetrachloroethylene	3	Acenaphthene95-69-2			7.58	6.44	0.85	45	0	0	0.05
405	2a	Tetrachloroethylene	3	Benzo[ghi]perylene			0.29	0.14	0.48	3	0	0	0.04
582	2b	Chrysene	3	2-Butoxyethanol			0.00	0.00	0.93	7	0	0	0.01
660	2b	Dichloromethane (methylene chlorid	3	Sulfur dioxide			156.70	116.86	0.75	740	4	0	0.01
662	2b	Dichloromethane (methylene chlorid	3	Xylenes			152.14	111.43	0.73	720	45	27	0.03
701	2b	Ethylbenzene	3	Acrolein			10.17	9.24	0.91	405	324	68	0.04
879	3	Acenaphthene95-69-2	3	2-Butoxyethanol			0.03	0.02	0.95	14	0	0	0.03
907	3	Acrolein	3	Isopropanol			1.89	1.77	0.93	170	128	52	0.03
915	3	Acrolein	3	Sulfur dioxide			27.73	21.18	0.76	483	295	60	0.02
916	3	Acrolein	3	Toluene			10.17	9.24	0.91	403	236	56	0.03
917	3	Acrolein	3	Xylenes			23.87	18.28	0.77	312	180	68	0.05
919	3	Benzo[ghi]perylene	3	Benzo[e]pyrene			0.74	0.54	0.73	3	3	3	0.02
925	3	Benzo[ghi]perylene	3	Fluorene			0.97	0.44	0.46	3	3	3	0.04
929	3	Benzo[ghi]perylene	3	4,4'-Methylenediphenyl diisocyanate			2.85	1.56	0.55	10	0	0	0.05
961	3	2-Butoxyethanol	3	Fluoranthene			0.04	0.04	0.96	2	0	0	0.04
962	3	2-Butoxyethanol	3	Fluorene			0.62	0.58	0.93	15	0	0	0.04
972	3	2-Butoxyethanol	3	Sulfur dioxide			152.09	111.20	0.73	780	53	53	0.03
1014	3	Ethylene	3	4,4'-Methylenediphenyl diisocyanate			0.00	0.00	0.96	54	0	0	0.02
1053	3	Hydrochloric acid	3	4,4'-Methylenediphenyl diisocyanate			1.09	1.05	0.96	12	0	0	0.04
1061	3	Hydrochloric acid	3	Xylenes			171.01	103.61	0.61	785	124	124	0.04

Figure 3.2: Detailed information of all discovered co-location rules on Alberta dataset by CM.

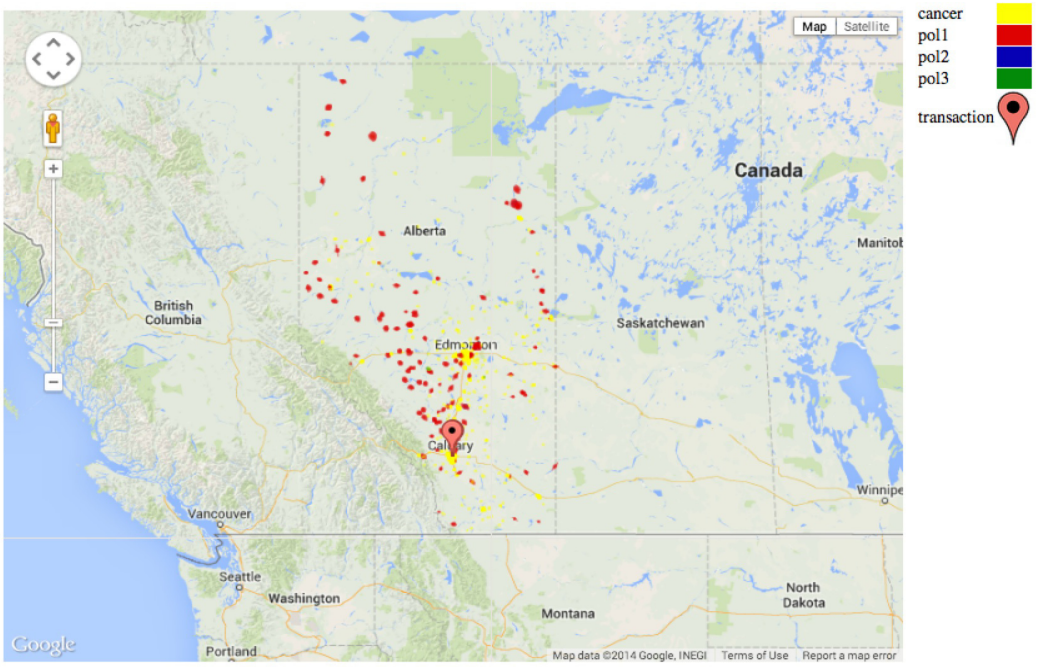


Figure 3.3: A visualization of a co-location rule in the map.

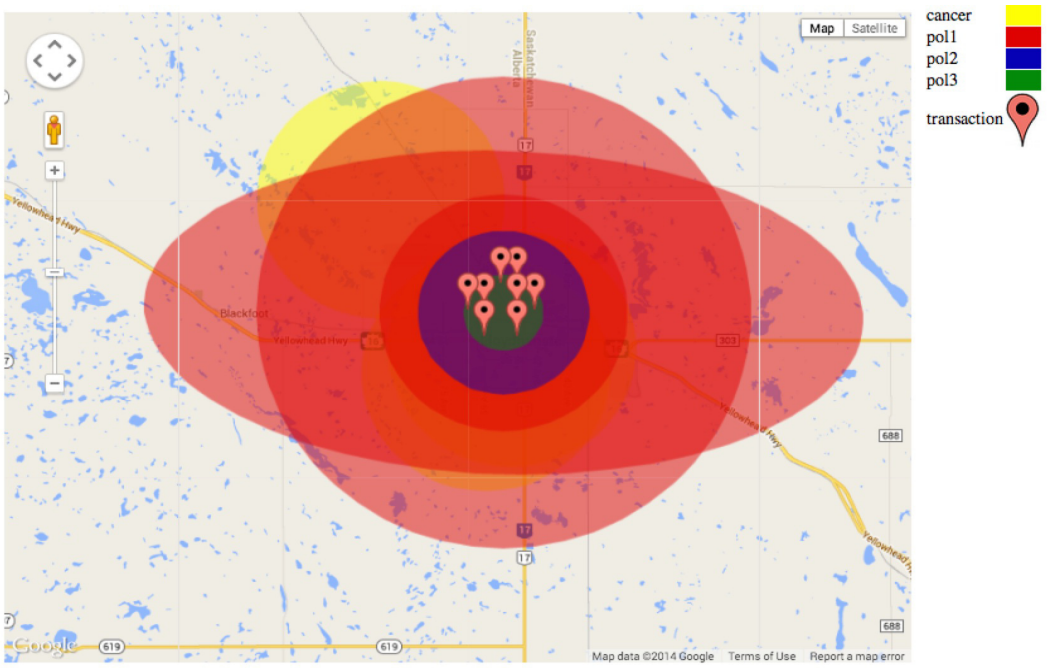


Figure 3.4: A regional zoom in view of the co-location rule in Figure 3.3.

grid point has to intersect with at least one pollutant feature; it either intersects with cancer or not. For the type of grid point $(Pol_{grid}, Cancer)$ intersects with both pollutant(s) and cancer, if we can find at least one co-location rule $Pol \rightarrow Cancer$ in the classifier that correctly matches it, i.e. $Pol \subseteq Pol_{grid}$, the grid point is indicated as correctly classified. For the other type of grid point $(Pol_{grid}, \neg Cancer)$ intersects with pollutant(s) but not cancer, if there does not exist any co-location rules $Pol \rightarrow Cancer$ that match it, i.e. $Pol \nsubseteq Pol_{grid}$, the grid point is also indicated as correctly classified. Otherwise, the grid points are considered as misclassified. The ratio of correctly classified grid points to the total number of sampled grid points is output as the classification accuracy. Figure 3.5 shows a toy example of the evaluation process. In the datasets of Alberta and Manitoba, we randomly sample 1000 grid points each time, repeats 100 times, and calculate the average classification accuracy for the previously mentioned three methods. Table 3.2 and Table 3.3 present the evaluation results, along with the classification accuracy (ACC), the specificity (SPE) and sensitivity (SEN) are also listed. As can be observed from the classification accuracy, CMCStatApriori is better than CM and CMKingfisher. The classification accuracy is much higher in Alberta compared with Manitoba.

One possible explanation is that the co-location association between chemical pollutants and children cancer cases is stronger in Alberta. Both the number of co-location rules and the classification accuracy is very low in Manitoba, therefore, it is possible that chemical pollutants and children cancer cases are more likely to be independent in Manitoba. We can also notice that the specificity is much higher than the sensitivity in both datasets. High specificity means that grid points without cancer are seldom misclassified; on the other hand, low sensitivity indicates that grid points with cancer are mostly misclassified. This phenomenon may imply that the co-location associations between chemical pollutants and children cancer cases is weak. However, these assumptions still need to be carefully scrutinized.

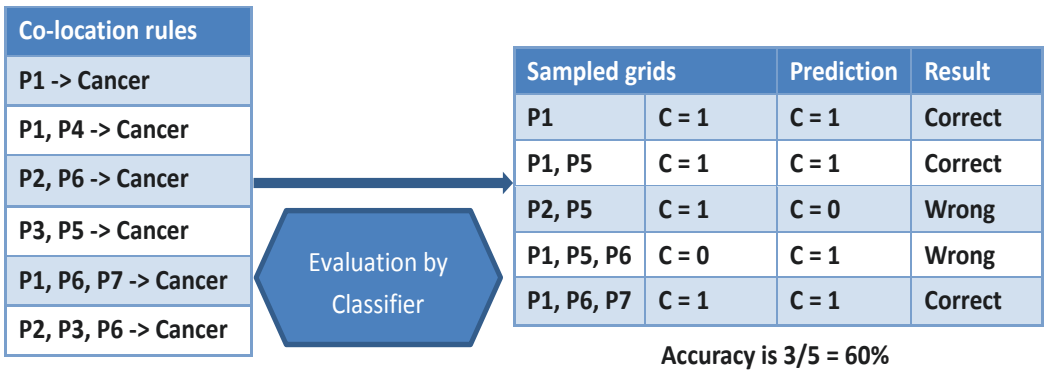


Figure 3.5: Toy example of the classifier evaluation.

Table 3.2: Evaluation of CM, CMKingfisher and CMCStatApriori using Accuracy, Specificity and Sensitivity on Alberta dataset.

	Alberta		
	ACC	SPE	SEN
CM	83.9 ± 3.3	97.6 ± 1.6	11.4 ± 8.1
CMKingfisher	69.2 ± 4.1	77.4 ± 4.1	$\mathbf{28.6 \pm 11.4}$
CMCStatApriori	$\mathbf{84.7 \pm 3.4}$	$\mathbf{99.6 \pm 0.7}$	6.6 ± 6.4

The only parameter in CMCStatApriori is the z_{min}. In this subsection, we also discuss the effect of the parameter z_{min}. As shown in Figure 3.6 and Figure 3.7, the number of discovered co-location rules drops when we increase z_{min}. We were not able to find any statistically significant co-location rules when $z_{min} > 170$ in Alberta and when $z_{min} > 50$ in Manitoba. In Figure 3.8 and Figure 3.9, the average

Table 3.3: Evaluation of CM, CMKingfisher and CMCStatApriori using Accuracy, Specificity and Sensitivity on Manitoba dataset.

	Manitoba		
	ACC	SPE	SEN
CM	22.0 ± 4.3	55.8 ± 11.2	$\mathbf{13.4 \pm 3.7}$
CMKingfisher	26.6 ± 4.6	$\mathbf{96.4 \pm 3.6}$	8.7 ± 3.0
CMCStatApriori	$\mathbf{27.4 \pm 4.1}$	83.4 ± 7.7	12.2 ± 3.4

classification accuracy of the sampled grid points is presented. The classification performance is poor when the z-score threshold is set to be low. Besides, there exists a turning point ($z_{min} = 100$ in Alberta, $z_{min} = 30$ in Manitoba) where the accuracy improves dramatically. In the Alberta dataset, there is not much difference when z_{min} varies from 110 to 170, while in the Manitoba dataset, the performance is best when z_{min} is set to be 40.

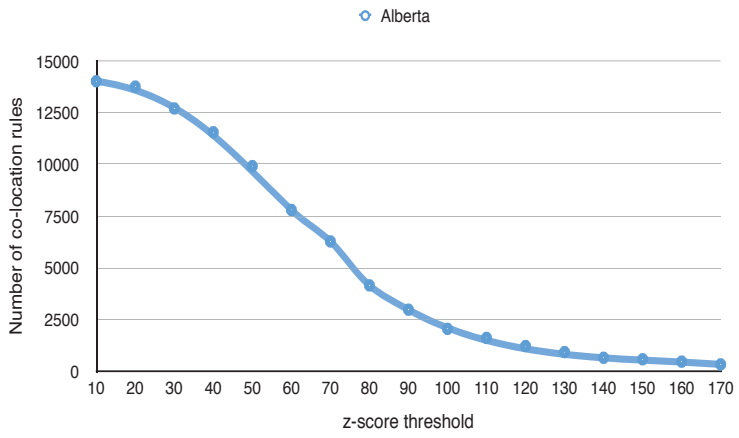


Figure 3.6: Number of co-location rules on Alberta dataset.

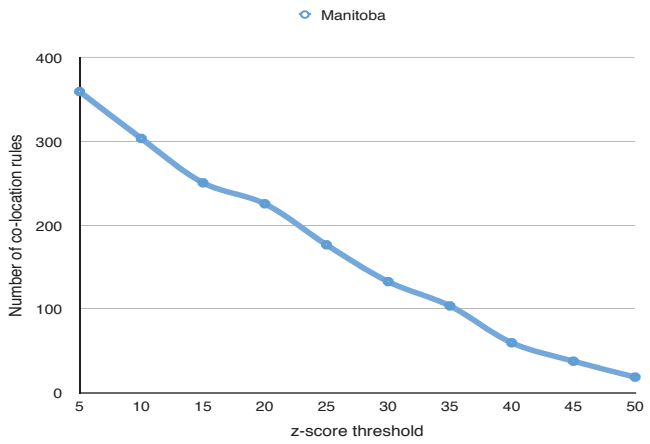


Figure 3.7: Number of co-location rules on Manitoba dataset.

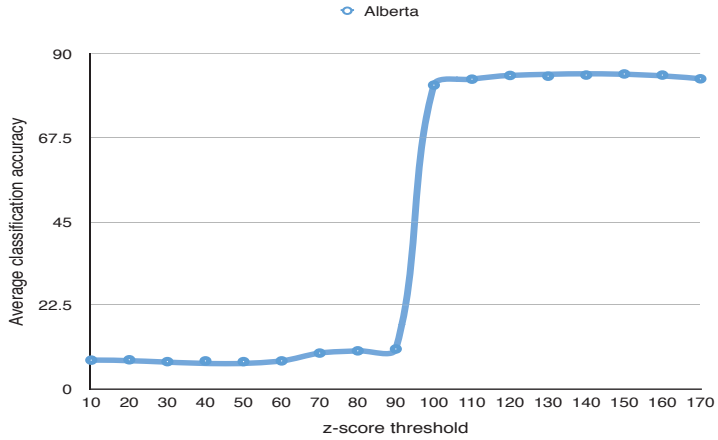


Figure 3.8: Average classification accuracy of sampled grid points on Alberta dataset.

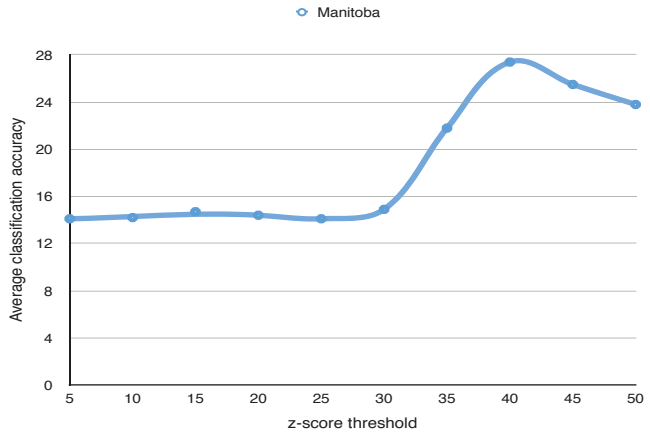


Figure 3.9: Average classification accuracy of sampled grid points on Manitoba dataset.

Chapter 4

Statistically Significant Classification Association Rules for Classification

This Chapter introduces the proposed associative classifier, SigDirect (**S**tatistically S**IG**nificant **D**ependent Class**I**fication Association **R**ul**E**s for **C**lassifica**T**ion). Unlike other associative classifiers, SigDirect does not need any confounding parameter settings. SigDirect consists of three phases: In the first step, SigDirect directly mines the complete set of statistically significant CARs. We propose a rule generator algorithm, Constrained-Kingfisher, which is based on Kingfisher [29, 31], by pushing the rule shape and constraint in the rule generation phase, the discovered CARs are statistically significant as well as non-redundant. Once the CARs are generated, we then apply a rule pruning strategy to build the classifier by selecting the best CAR for each transaction in the training dataset similar to Harmony [55]. In our algorithm, the best CAR is considered as the CAR with the highest confidence value. Finally, we investigate different strategies to select a subset of CARs for prediction. At the end of this chapter, we also introduce how to integrate the negative statistically significant CARs into the associative classifier and show a promising experimental result.

4.1 Basic Notations and Definitions

Definition 3. Dependency of a CAR*:*

Let $\mathcal{D}$ be a transaction database, it consists of a set of items $\mathcal{I} = \{i_1, i_2, ..., i_m\}$ and a set of class label $C = \{c_1, c_2, ..., c_n\}$. Each transaction T is associated with a set

of items X and a particular class label c_k, where $X \subseteq \mathcal{I}$ and $c_k \in C$. A CAR is in the form of $X \rightarrow c_k$, the antecedent part and the consequent part of the CAR is dependent if and only if $P(X, c_k) \neq P(X)P(c_k)$.

Definition 4. Fisher's exact test*:*

The dependency of the CAR $X \rightarrow c_k$ is considered statistically significant at level α, if the probability p of observing equal or stronger dependency in a dataset complying with a null hypothesis is not greater than α. In the null hypothesis, X and c_k are assumed to be independent of each other. The probability p, i.e. p-value, can be calculated by Fisher's exact test [29, 31]:

$$p_F(X \rightarrow c_k) = \sum_{i=0}^{min\{\sigma(X,\neg c_k)\sigma(\neg X,c_k)\}} \frac{\binom{\sigma(X)}{\sigma(X,c_k)+i}\binom{\sigma(\neg X)}{\sigma(\neg X,\neg c_k)+i}}{\binom{|\mathcal{D}|}{\sigma(c_k)}} \tag{4.1}$$

where $\sigma(X)$ denotes the frequency of X. The significance level α is usually set to be 0.05.

Definition 5. Confidence*:*

The confidence of the CAR $X \rightarrow c_k$ is:

$$conf(X \rightarrow c_k) = \frac{\sigma(X, c_k)}{\sigma(X)} \tag{4.2}$$

Definition 6. Parent and Child CAR*:*

Let the CAR $X \rightarrow c_k$ like before. The CAR $Y \rightarrow c_k$ is considered as its parent CAR if $Y \subsetneq X$ and $|Y| = |X| - 1$. The CAR $X \rightarrow c_k$ is considered to be the child CAR of $Y \rightarrow c_k$.

Definition 7. Non-redundant CARs*:*

The CAR $X \rightarrow c_k$ is non-redundant, if there does not exist any CARs in the form of $Y \rightarrow c_k$ such that $Y \subsetneq X$ and $p_F(Y \rightarrow c_k) < p_F(X \rightarrow c_k)$.

Definition 8. Minimality*:*

The CAR $X \rightarrow c_k$ is minimal, if and only if $X \rightarrow c_k$ is non-redundant, and, there does not exist any CARs in the form of $Z \rightarrow c_k$ such that $X \subsetneq Z$ and $p_F(Z \rightarrow c_k) < p_F(X \rightarrow c_k)$.

It has been proven in the literature [37, 29, 31] that if the CAR $X \rightarrow c_k$ is minimal, we can get $P(c_k|X) = 1$, i.e. the conditional probability of c_k given X is 1.

4.2 Generating Classification Association Rules

To find the relevant CARs for classification, SigDirect first needs to generate the complete set of statistically significant CARs. It means the rule in the form of $X \rightarrow c_k$ has a relevant small p_F-value, i.e. $p_F(X \rightarrow c_k) \leq \alpha$. Since the p_F value is not a monotonic property, it is impossible for us to do some pruning as apriori-like algorithms. One possible solution is to enumerate the whole search space, the size of the whole search space is $|\mathcal{P}(\mathcal{I})|.|C|$, where $\mathcal{P}(\mathcal{I})$ is the power set of $\mathcal{I}$, it grows exponentially with the size of antecedent items, therefore deriving effective pruning strategies is urgent and of great importance.

The first pruning strategy is useful at the beginning of the algorithm, it can help us weed out some items that are impossible in the antecedent of a CAR.

Theorem 2. *There exists a threshold γ smaller than 0.5. When the frequency of an item I is smaller than γ, i.e. $P(I) < \gamma$, the item I is impossible to be in the antecedent part of any statistically significant CARs.*

Proof. First, we assume that item I can be the consequent of a CAR $X \rightarrow I$, where $X \subseteq \mathcal{I} \backslash I$. According to [29, 31], the minimum p_F value of the rule $X \rightarrow I$ is $\frac{\sigma(I)!\sigma(\neg I)!}{|\mathcal{D}|!}$, the minimum value is smallest when $\sigma(I) = \sigma(\neg I) = \frac{|\mathcal{D}|}{2}$. If $\frac{\frac{|\mathcal{D}|}{2}!\frac{|\mathcal{D}|}{2}!}{|\mathcal{D}|!} > \alpha$, then it is certain that $\frac{\sigma(I)!\sigma(\neg I)!}{|\mathcal{D}|!} > \alpha$. If $\frac{\frac{|\mathcal{D}|}{2}!\frac{|\mathcal{D}|}{2}!}{|\mathcal{D}|!} \leq \alpha$, we can still make $\frac{\sigma(I)!\sigma(\neg I)!}{|\mathcal{D}|!} > \alpha$ when either $\sigma(I)$ or $\sigma(\neg I)$ deviates a lot from $\frac{|\mathcal{D}|}{2}$. It indicates there exists a threshold $\gamma \leq 0.5$, thus when $\sigma(I) < \gamma|\mathcal{D}|$ or $\sigma(I) > (1-\gamma)|\mathcal{D}|$, the item I cannot appear in the consequent of a statistically significant CAR. As mentioned above, we intend to find CARs in which item I can only be in the antecedent part. If the condition of $\sigma(I) < \gamma|\mathcal{D}|$ or $\sigma(I) > (1-\gamma)|\mathcal{D}|$ holds, according to [29, 31], when $\sigma(I) < \frac{|\mathcal{D}|}{2}$, the item I cannot even appear in the antecedent part. In this way, we can derive a threshold γ smaller than 0.5, when the frequency of the item

I is smaller than γ, the item cannot appear in any statistically significant CARs, let alone in the antecedent part. □

Through Theorem 2, some impossible antecedent items I are pruned before further analysis. Assume there are s antecedent items left, where $s \leq m$, then the rest s items are arranged and renamed in ascending order by their frequencies, i.e. $\mathcal{I}_{rest} = \{i_1, i_2, ..., i_s\}$, where $P(i_1) \leq P(i_2) \leq ... \leq P(i_s)$. Then, in order to traverse the whole search space, an enumeration tree is built over the reordered antecedent itemsets $\mathcal{I}_{rest}$ (Figure 4.1). Since the enumeration tree lists the whole search space, for each node in the enumeration tree, we check all the n possible CARs $X \rightarrow c_k (X \subseteq \mathcal{I}_{rest}, k \in \{1, ..., n\})$ to see if they are statistically significant, where X denotes the antecedent item sets in the corresponding node, as illustrated in Figure 4.1.

The second pruning strategy is of great importance since it has an effect in the whole search process.

Theorem 3. *Let $X \rightarrow c_k$ be a CAR as before, where $X \subseteq \mathcal{I}_{rest}, k \in \{1, ..., n\}$ and $Q \subseteq (\mathcal{I}_{rest} \backslash X)$. If $\sigma(X) \leq \sigma(c_k)$, then we can get:*

$$p_F(XQ \rightarrow c_k) \geq \frac{\sigma(\neg X)!\sigma(c_k)!}{|\mathcal{D}|!(\sigma(c_k) - \sigma(X))!} \tag{4.3}$$

Proof. This theorem is from Kingfisher, the detailed proof can be referred in [29, 31]. □

In Theorem 3, we can find that best value of $p_F(XQ \rightarrow c_k)$ can be considered as a low bound of $p_F(X \rightarrow c_k)$, i.e. $best(p_F(XQ \rightarrow c_k)) \leq p_F(X \rightarrow c_k)$, the "best value" is the lower bound in Theorem 3. Therefore, if the lower bound is larger than α, the dependency of the CAR $X \rightarrow c_k$ is not statistically significant and can be pruned. Otherwise, we define the CAR as ***PSS***, "potentially statistically significant", the CAR needs to be further calculated to get the exact p_F-value to see if it is indeed statistically significant.

Another important property is necessary in the search algorithm.

Theorem 4. *If the CAR $X \rightarrow c_k$ is PSS, then any of its parent CARs $Y \rightarrow c_k$ is also PSS, where $Y \subsetneq X$ and $|Y| = |X| - 1$.*

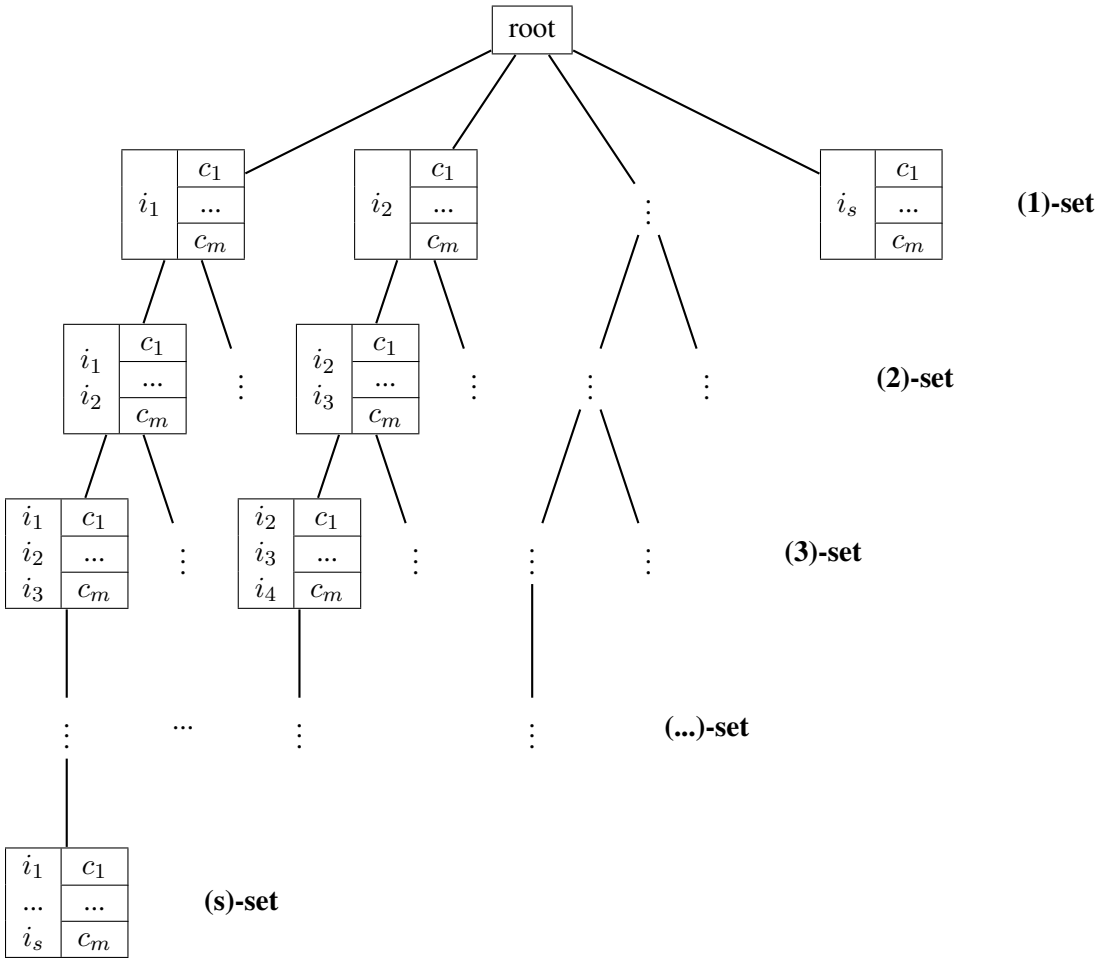


Figure 4.1: Enumeration of the whole search space of SigDirect.

Proof. To make $X \rightarrow c_k$ to be PSS, there are two supporting scenarios. First situation is when $\sigma(X) > \sigma(c_k)$. Since $Y \subsetneq X$, thus $\sigma(Y) > \sigma(X) > \sigma(c_k)$, and it is easy to see the parent CAR $Y \rightarrow c_k$ is also PSS. The second situation is when $\sigma(X) \leq \sigma(c_k)$ and $best(p_F(XQ \rightarrow c_k)) < \alpha$, where $Q \subseteq (\mathcal{I}_{rest} \backslash X)$. $XQ = Y(X \backslash Y)Q = YR$ holds for any $Q \subseteq (\mathcal{I}_{rest} \backslash X)$, because $(X \backslash Y) \subseteq (\mathcal{I}_{rest} \backslash Y)$ and $Q \subseteq (\mathcal{I}_{rest} \backslash X) \subseteq (\mathcal{I}_{rest} \backslash Y)$, thus $R = (X \backslash Y)Q \subseteq (\mathcal{I}_{rest} \backslash Y)$ and therefore, there must exists $R \subseteq (\mathcal{I}_{rest} \backslash Y)$ making $best(p_F(YQ \rightarrow c_i)) < \alpha$, i.e. CAR $Y \rightarrow c_i$ is PSS. □

With theorems 2, 3, 4, we can summarize the whole statistically significant CARs generation algorithm Constrained-Kingfisher: we first list all the candidate (1)-set CARs, those with only 1 item in the antecedent, as illustrated in the (1)-set level in Figure 4.1. Then for each candidate (1)-set CAR, we use Theorem 3 to see if it is PSS, non-PSS CARs can be directly pruned. PSS CARs are further checked to validate if they are statistically significant, i.e. $p_F \leq \alpha$. Between level 2 to level s, we take a breadth-first strategy, for each candidate CAR in these levels, we check if all of its parent CARs are PSS, if any of its parent CAR is not PSS (pruned already), then the candidate CAR is also not PSS and therefore can be pruned. In other words, if and only if all of its parent CARs are PSS, the CAR is considered PSS and will be further analyzed. Non-redundancy property is also taken into account, statistically significant CARs are checked if they are non-redundant before they are output. Furthermore, the minimality is also checked, if the non-redundant rule is also minimal, the CAR is also marked preventing expansion and all of its children CARs can be directly pruned. The pseudocode of the whole procedure is illustrated in Algorithm 4.

4.3 Rule Pruning

In the classification association rules generation phase, we have taken the non-redundancy property into consideration. However, the number of statistically significant CARs could still be very large. One possible disadvantage of a large number of CARs is that it could contain some noisy information which may mislead

Data: Transaction Database $\mathcal{D}$, set of antecedent item sets $\mathcal{I}$, class set C, significance level $\alpha = 0.05$.
Result: Statistically significant CARs set $\mathcal{R}$.
Prune impossible antecedent items with Theorem 2;
$\mathcal{I}_{rest}$: the arranged and renamed antecedent item set;
Create root node and level-1 nodes;
Set $l = 2$;
for *each candidate rule r in level-1* **do**
  if *r is* PSS **then**
    if $p_F(r) \leq \alpha$ **then**
      if *r is minimal* **then**
        r.minimal = 1;
        $\mathcal{R}$.add(r);
      else
        $\mathcal{R}$.add(r);
      end
    end
  else
    prune rule r from the enumeration tree;
  end
end
while $l \leq |\mathcal{I}_{rest}|$ **do**
  for *each candidate rule r in level* l **do**
    if *all parent rules of r are PSS and not minimal* **then**
      if $p_F(r) \leq \alpha$ **then**
        if *r is non-redundant* **then**
          if *r is minimal* **then**
            r.minimal = 1;
            $\mathcal{R}$.add(r);
          else
            $\mathcal{R}$.add(r);
          end
        end
      end
    else
      prune rule r and all its decedent rules from the enumeration tree;
    end
  end
  $l = l + 1$;
end

Algorithm 4: Constrained Kingfisher algorithm to generate statistically significant CARs.

the classification process. Another drawback is that a large number of CARs will make the classification process slower. This could be an important problem in applications where fast responses are required. Moreover, in classification applications where evidence checking is required, rule based models are an advantage but a large number of rules is a significant drawback and defeats the purpose. In order to reduce the number of CARs in the classification phase, many associative classifiers take a sequential database coverage paradigm. However, the final set of CARs may not be the globally best CARs for some instances in the training dataset. In order to reduce the number of CARs and find the globally best rules for all training instances, we take an instance centric rule pruning approach as Harmony [55], the classifier selects the best CAR for each instance in the training dataset, the best CAR is defined as the matching CAR with the highest confidence value. Each candidate CAR may be selected by multiple training instances, therefore, each candidate CAR is associated with an attribute "count", it indicates how many times the CAR is selected in the pruning process.

The detailed algorithm is shown in Algorithm 5:

Data: Set of statistically significant rules $\mathcal{R}$ found in the rule generation phase, transaction database $\mathcal{D}$.
Result: A subset of rules $\mathcal{R}_{new}$ for the classification process.
for *each instance t in the transaction database* $\mathcal{D}$ **do**
  Scan the set of candidate rules in $\mathcal{R}$ to find the matching rule r, i.e. ($r.antecedent \subseteq t.antecedent$ and $r.classlabel = t.classlabel$) with highest confidence value;
  if $r \notin \mathcal{R}_{new}$ **then**
    $\mathcal{R}_{new}$.add(r);
    r.count = 1;
  else
    $\mathcal{R}_{new}.r$.count + 1;
  end
end

Algorithm 5: Rule pruning phase in SigDirect.

4.4 Classifying New Instances

After the rule pruning phase, the subset of the most statistically significant CARs form the actual classifier. In this phase, we utilize the built classifier to make new predictions. Given a new instance without class label, the classification process searches the subset of CARs matching the new instance to make a class prediction. This subsection discusses the three approaches that we take to label new instances.

A simple solution is to select the matching rule in $\mathcal{R}_{new}$ with highest confidence value or lowest p_F-value and assign its label to the new instance. Another alternative is to divide all matching rules into groups according to their class labels. The groups are then ordered according to the average confidence value or average p_F-value. The class that has the highest average confidence value or the lowest average p_F-value will be assigned to the new instance. However, these two classification heuristics are often biased to minority classes. To solve this problem, an intuitive way is to calculate the total confidence value or total p_F-value instead of the average values. But p_F-value is different from confidence, the lower the value, the better the rule is, therefore, simply sum up the p_F value does not make any sense. To make it compatible to the confidence measure, we transform the p_F-value to log scale, all the subsequent steps are on the log-transformed values.

Then, we propose three different heuristics, denoted as S1, S2 and S3, to consider the sum of $ln(p_F)$, sum of confidence and sum of $ln(p_F)$.confidence of matching rules in each class, respectively:

- S1: Calculate the sum of $ln(p_F)$ of matching rules in each class, the class label of the new instance is determined by the class of the lowest value
- S2: Calculate the sum of confidence of matching rules in each class, the class label of the new instance is determined by the class of the highest value
- S3: Calculate the sum of $ln(p_F)$.confidence of matching rules in each class, the class label of the new instance is determined by the class of the lowest value

Algorithm 6 describes three heuristic classification methods of a new instance.

Data: A new instance o to be classified. Set of rules $\mathcal{R}_{new}$ from rule pruning phase.
Result: Class label of the new instance o.
$T = \emptyset$; // set of rules matching o
for *each rule r in* $\mathcal{R}_{new}$ **do**
  $i = 1$;
  while $i \leq r.count$ **do**
    if r.*antecedent* $\subseteq$ o.*antecedent* **then**
      T.add(r);
    end
    $i = i + 1$;
  end
end
divide T into n subsets by class labels: $T_1, T_2, ..., T_n$;
// Classification with S1
for *each subset* $T_1, T_2, ..., T_n$ **do**
  sum up the $ln(p_F)$ values of matching CARs in each subset
end
assign the class with the lowest sum of $ln(p_F)$ value to o;
// Classification with S2
for *each subset* $T_1, T_2, ..., T_n$ **do**
  sum up the confidence values of matching CARs in each subset
end
assign the class with the highest sum of confidence value to o;
// Classification with S3
for *each subset* $T_1, T_2, ..., T_n$ **do**
  sum up the $ln(p_F)$.confidence values of matching CARs in each subset
end
assign the class with the lowest sum of $ln(p_F)$.confidence value to o;

Algorithm 6: Classification of new instances in SigDirect.

4.5 Experiments

4.5.1 Datasets

We evaluate our SigDirect method on 20 datasets from UCI Machine Learning Repository [12]. In these datasets, the numerical attributes have been discretized by the author of [21], the discretization strategy is different from that used in [40, 38], thus the classification performance may be different from the results reported before. All the following experimental results on each dataset are reported as an average of a 10-fold cross validation.

4.5.2 Classification Accuracy

We evaluate our SigDirect with three different classification strategies S1, S2, S3 against two rule based classifiers C4.5 [47] and FOIL [46], three associative classifiers CBA [40], CMAR [38] and a hybrid between rule based and associative classifier CPAR [62] on the previous mentioned 20 discretized UCI datasets. The results are reported in the form of average classification accuracy over 10-folds. All classification methods are evaluated on the same generated 10-folds to ensure a fair comparison. The parameters of C4.5 are set as default values [47]. In FOIL, we allow a maximum of 3 features in the antecedent of a rule. In CBA, CMAR, the minimum support is set to be 1%, the minimum confidence is 50%, the maximum number of antecedent features and the maximum number of mined classification association rules are set to be 6 and $80,000$, respectively. In CPAR, we also follow the same parameter settings as [62], minimum gain threshold set to 0.7, total weight threshold to 0.05 and decay factor to 2/3.

Table 4.1 presents the classification accuracy of the following methods: C4.5, FOIL, CBA, CMAR, CPAR and our SigDirect method with three different classification heuristics S1, S2 and S3. Along with the accuracy result, the name of the dataset and the number of records are also reported.

As can be observed from Table 4.1, the proposed SigDirect with S2 achieves the best overall classification accuracy, followed by SigDirect with S3 and SigDirect with S1. All of these three classifiers outperform C4.5, FOIL, CBA, CMAR and

CPAR on the average over the 20 datasets.

To have a more fair comparison between these classifiers, we show how many times the classifier is the best and how many times it is the runner-up. Table 4.2 shows the comparison results, SigDirect with S2 is still the best among these classifiers. It wins 8 out of 20 datasets, i.e. 40% of all datasets, and is the runner-up 5 times. CMAR, in the second place, wins in 6 datasets and gets the runner-up twice.

Combing the comparison results from Table 4.1 and Table 4.2 together, SigDirect with S2 is always the best, SigDirect with S1 and SigDirect with S3 can be considered as competitive classifiers. It demonstrates that in the classification accuracy aspect, our SigDirect classification method can be viewed as a competitive and even slightly better classifier with the state-of-the-art rule based and associative classifiers.

4.5.3 Number of Rules

In associative classification, the number of CARs before and after rule pruning phase are both very important indicators to measure a classifier. On one hand, if we get a small number of CARs after rule generation phase, people are able to sift through these CARs to determine validity, to choose a subset of them or even to edit them to inject domain knowledge not reflected in the training data. Moreover, rule pruning strategies are possible since these CARs are more readable. On the other hand, a small number of CARs post rule pruning can make the classification phase faster. In addition, after rule pruning phase, because of transparency of the CARs, manually updating some CARs is favourable and practical in many applications if the number of CARs is reasonable. Therefore, we evaluate the number of CARs generated by our Constrained-Kingfisher algorithm and the number of CARs after rule pruning phase (instance centric way). Table 4.3 shows the number of CARs of two associative classifiers CBA, CMAR and our SigDirect method. The number of CARs before and after rule pruning phase are both presented. We also list the number of rules in C4.5, FOIL and CPAR in Table 4.3. In CBA and CMAR, the rule generation stops if the number of CARs is larger than 80,000, but even in this situation, we can find that the number of CARs generated by SigDirect is much

Table 4.1: Comparison of classification results: C4.5, FOIL, CBA, CMAR, CPAR and SigDirect.

Dataset	#cls	#rec	C4.5	FOIL	CBA	CMAR	CPAR	SigDirect		
								S1	S2	S3
adult	2	48842	78.8	**84.6**	84.2	81.3	77.3	83.9	83.9	84.1
anneal	6	898	76.7	**98.8**	94.5	90.7	95.1	96.8	94.0	96.7
breast	2	699	91.5	89.3	**94.1**	89.9	93.0	91.4	91.7	91.6
cylBands	2	540	69.1	74.1	76.1	**76.5**	70.0	74.4	73.7	74.4
flare	9	1389	82.1	83.8	84.2	**84.3**	63.9	83.0	84.2	84.2
glass	7	214	65.9	66.5	68.4	**71.1**	64.9	66.8	69.6	68.7
heart	5	303	**61.5**	55.2	57.8	56.2	53.8	56.4	58.1	57.4
hepatitis	2	155	84.1	77.8	42.2	79.6	75.5	83.2	**85.2**	82.6
horseColic	2	368	70.9	**83.4**	78.8	82.3	81.2	81.3	80.7	81.3
ionosphere	2	351	84.6	86.6	32.5	**91.5**	88.9	87.2	85.5	87.2
iris	3	150	91.3	94.0	93.3	94.0	**94.7**	94.0	94.0	93.3
led7	10	3200	**73.8**	60.5	73.1	73.2	71.3	**73.8**	**73.8**	73.7
letRecog	26	20000	50.4	50.0	32.5	28.3	58.2	48.2	**58.8**	52.6
mushroom	2	8124	92.8	99.5	46.7	**100.0**	98.5	**100.0**	**100.0**	**100.0**
pageBlocks	5	5473	92.0	92.4	90.9	90.1	**92.5**	91.2	91.2	91.2
penDigits	10	10992	70.5	84.1	**92.3**	87.4	80.5	84.3	88.4	84.6
pima	2	768	71.7	71.9	74.6	74.4	74.0	74.6	**75.1**	74.6
soybean	19	683	60.3	88.0	89.2	88.1	83.1	89.5	**90.0**	89.8
wine	3	178	75.8	88.2	49.6	**92.7**	88.2	92.1	**92.7**	92.1
zoo	7	101	91.0	93.1	40.7	93.0	**94.1**	**94.1**	**94.1**	**94.1**
Average			76.7	81.1	69.8	81.3	79.9	82.3	**83.2**	82.7

Table 4.2: Best and runner-up counts comparison between C4.5, FOIL, CBA, CMAR, CPAR and SigDirect.

Classifiers	Best	Runner-up
C4.5	1	2
FOIL	3	4
CBA	2	4
CMAR	6	2
CPAR	3	3
SigDirect with S1	3	4
SigDirect with S2	8	5
SigDirect with S3	2	5

smaller than that generated by CBA and CMAR, in most datasets (18 out of 20), the number is even an order of magnitude smaller. It can also be observed, after the rule pruning process, the number of CARs by SigDirect is smaller, in 16 out of 20 datasets, the number of CARs is below 100, which makes it more readable and more manually editable.

All in all, SigDirect dramatically reduce the number of CARs compared with CBA and CMAR in the rule generation phase without jeopardizing accuracy and even improving it. After rule pruning phase, the number of CARs for classification is still very small. The overall small number of CARs makes SigDirect superior to other associative classifiers when there is a slight difference between classification accuracies. The number of CARs remains comparable and even smaller than the case of C4.5, FOIL and CPAR.

4.5.4 Effects of Pruning Strategies and Classification Heuristics

In SigDirect, we take an instance-centric method to do rule pruning to reduce the number of CARs. Here, we first compare the effect of this pruning strategy with the database coverage paradigm (pruned by confidence) which is widely. In Table 4.4, the classification results with these two different rule pruning strategies are presented and compared. As can be observed, the classification accuracy indeed improves when we take the instance centric pruning strategy, no matter what kind of classification heuristics are used. The average classification accuracy is higher around 1% to 2% percent.

Table 4.3: Comparison of the number of rules: C4.5, FOIL, CBA, CMAR, CPAR and SigDirect.

Dataset	C4.5	FOIL	CPAR	CBA		CMAR		SigDirect	
				before rule pruning	after rule pruning	before rule pruning	after rule pruning	before rule pruning	after rule pruning
adult	1176.5	229.4	84.6	87942.6	691.8	82694.5	2982.5	144.9	91.2
anneal	17.0	29.5	25.2	89101.9	27.3	54945.5	208.4	372.0	39.6
breast	8.8	13.9	6.0	2711.4	13.5	2372.1	69.4	25.1	10.9
cylBands	37.2	45.5	35.8	64194.3	135.4	21556.8	622.8	8610.5	149.0
flare	54.4	95.5	48.1	11910.2	115.1	4844.9	347.1	634.9	75.6
glass	14.8	47.1	34.8	10171.0	63.7	6901.0	274.5	279.6	52.2
heart	23.9	63.4	44.0	41899.4	78.4	18700.6	464.2	519.0	78.1
hepatitis	8.1	23.6	14.3	181441.1	2.3	46961.2	165.7	196.1	32.3
horseColic	25.6	64	19	178353.3	116.4	39923.9	499.9	324.8	84.4
ionosphere	18.3	24.9	22.8	95242.1	27.3	64631.9	272.7	1455.9	79.5
iris	8.4	7.9	7.4	171.0	12.3	152.7	63.4	14.1	5.9
led7	63.2	79.1	31.7	453.6	71.2	453.6	206.3	339.0	104.3
letRecog	1565.2	559.3	789.1	2402.9	151.4	2402.9	1132.5	42297.9	2343.9
mushroom	121.2	11.7	11.1	104666.3	2.0	102061.4	102.6	919.4	22.5
pageBlocks	16.3	43.6	29.9	1546.6	7.6	1403.3	80.6	240.6	29.1
penDigits	758.3	163.3	135.1	91125.5	657.6	91125.5	4501.5	8626.5	609.8
pima	24.4	58.7	21.7	1769.6	43.2	916.3	203.3	123.9	35.8
soybean	57.1	46.3	76.6	26912.0	65.8	26912.0	293.2	314701.3	75.3
wine	12.8	15.9	15.2	82120.6	4.7	56228.4	122.7	108.8	28.5
zoo	5.3	9.9	16.9	82616.7	2.0	68199.3	35.0	2282.6	9.4

Table 4.4: Classification results comparison with instance centric and database coverage pruning methods.

Dataset	S1		S2		S3	
	instance centric	database coverage	instance centric	database coverage	instance centric	database coverage
adult	**83.9**	**83.9**	**83.9**	83.2	**84.1**	83.6
anneal	**96.8**	96.1	**94.0**	88.0	**96.7**	94.4
breast	**91.4**	90.7	**91.7**	91.3	**91.6**	90.7
cylBands	**74.4**	73.3	**73.7**	72.0	**74.4**	73.9
flare	**83.0**	80.3	**84.2**	83.2	**84.2**	83.7
glass	**66.8**	66.4	69.6	**72.0**	**68.7**	67.8
heart	56.4	**57.1**	**58.1**	56.8	**57.4**	56.4
hepatitis	**83.2**	82.6	**85.2**	83.2	**82.6**	81.9
horseColic	**81.3**	80.2	**80.7**	76.7	**81.3**	80.7
ionosphere	87.2	**88.9**	**85.5**	85.0	87.2	**88.9**
iris	**94.0**	93.3	94.0	**94.7**	**93.3**	**93.3**
led7	**73.8**	73.5	**73.8**	73.5	**73.7**	72.7
letRecog	**48.2**	46.7	58.8	**61.8**	**52.6**	51.1
mushroom	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
pageBlocks	**91.2**	90.7	**91.2**	91.1	**91.2**	90.7
penDigits	**84.3**	81.5	88.4	**90.3**	**84.6**	81.5
pima	**74.6**	68.5	**75.1**	67.7	**74.6**	68.6
soybean	**89.5**	87.6	**90.0**	89.6	**89.8**	88.4
wine	92.1	**92.7**	**92.7**	88.2	92.1	**92.7**
zoo	**94.1**	93.1	**94.1**	93.1	**94.1**	**94.1**
Average	**82.3**	81.3	**83.2**	82.1	**82.7**	81.8

Next, in order to investigate the efficacy of the measure M ($ln(p_F)$, confidence or $ln(p_F)$.confidence) in the classification phase , to see if classifying by the sum of M can overcome the bias problem caused by classifying with only the best rule or by the average of M, we compare S1, S2 and S3 with their corresponding alternatives (B1, A1), (B2, A2) and (B3, A3). The compared classification heuristics B1, A1, B2, A2, B3 and A3 are listed below:

- B1: Select the matching rule with the lowest $ln(p_F)$ value, the class label of the new instance is determined by the selected rule
- A1: Calculate the average value of $ln(p_F)$ for matching rules in each class, the class label of the new instance is determined by the class of the lowest value
- B2: Select the matching rule with the highest confidence value, the class label of the new instance is determined by the selected rule
- A2: Calculate the average of confidence value for matching rules in each class, the class label of the new instance is determined by the class of the highest value
- B3: Select the matching rule with the lowest $ln(p_F)$.confidence value, the class label of the new instance is determined by the selected rule
- A3: Calculate the average of $ln(p_F)$.confidence value for matching rules in each class, the class label of the new instance is determined by the class of the lowest value

As shown in Table 4.5, S1, S2 and S3 have a better classification performance than their counterpart (B1, A1), (B2, A2), (B3, A3) with a higher average classification accuracy. It can be concluded that the classification heuristics in the "A" category are always the worst, "B" category heuristics are better than "A" category, but are still not as good as "S" category heuristics. Therefore, the classification heuristic that classifying a new instance by the sum of measure M ($ln(p_F)$, confidence or $ln(p_F)$.confidence) of all matching rules indeed helps to improve the

classification performance. When the measure M is the rule's confidence, the associative classifier is the best.

4.5.5 Statistical Analysis

From Table 4.1, we can conclude that our SigDirect algorithm gets a competitive and even better classification performance compared to other methods and the confidence is a better measure when measured against $ln(p_F)$ and $ln(p_F)$.confidence in the classification phase. Table 4.3 shows that our method gets a small number of CARs both before and after rule pruning phase. Table 4.4 and Table 4.5 indicate the superiority of the instance-centric rule pruning strategy and the summation effect, respectively. These conclusions are obtained mainly by measuring average classification accuracies and winning times. Although it gives us some intuition about the lead of a certain classifier, a certain rule pruning or a classification strategy, the conclusion is not forceful since the dominance is unsurpassed over all 20 datasets.

To better validate the conclusions we get, we use Demsar's [25] method, conducting a set of non-parametric statistical tests to compare different classifiers over multiple datasets.

In the first step, Friedman test is applied to measure if there is a significant difference between different classification models on Table 4.1. We first rank different classifiers on each dataset separately, r_i^j denotes the j-th of k classifiers on i-th of N datasets. Then the average rank of j-th classifier is computed as $R_j = \frac{1}{N}\sum_i r_i^j$. In the null hypothesis, the average ranks of different classifiers are equivalent, and the Friedman statistic is:

$$\chi_F^2 = \frac{12N}{k(k+1)}(\sum_j R_j^2 - \frac{k(k+1)^2}{4})$$

with $k-1$ degrees of freedom, when $N > 10$ and $k > 5$. If the Friedman statistic exceeds a critical value, the null hypothesis is rejected and we conduct post-hoc tests to make pairwise comparisons between classifiers, otherwise, there is no statistical significance among the k classifiers over these N datasets.

The Friedman statistics of 8 classification methods from Table 4.1 exceeds the critical value, so we continue to use Wilcoxon signed-ranks test to compare the

Table 4.5: Comparison of classification heuristics S1 with (B1, A1), S2 with (B2, A2) and S3 with (B3, A3).

Dataset	SigDirect								
	S1	B1	A1	S2	B2	A2	S3	B3	A3
adult	**83.9**	71.3	70.1	**83.9**	80.8	76.2	**84.1**	80.7	80.4
anneal	96.8	**97.4**	97.1	94.0	**94.2**	**94.2**	**96.7**	94.7	94.5
breas	**91.4**	89.7	88.1	**91.7**	90.0	90.0	**91.6**	91.4	90.7
cylBands	**74.4**	71.9	71.1	**73.7**	66.1	65.9	**74.4**	71.9	71.1
flare	**83.0**	82.9	81.6	**84.2**	83.9	83.9	**84.2**	83.9	84.0
glass	**66.8**	62.1	56.1	69.6	**70.6**	67.3	**68.7**	64.0	62.6
heart	56.4	56.4	**56.8**	**58.1**	51.5	52.1	**57.4**	55.4	56.4
hepatitis	**83.2**	75.4	73.5	85.2	**85.8**	83.2	**82.6**	81.3	80.0
horseColic	**81.3**	81.0	81.0	**80.7**	70.7	72.0	**81.3**	81.0	81.0
ionosphere	87.2	**88.3**	87.5	**85.5**	80.1	74.6	87.2	**88.3**	87.5
iris	**94.0**	93.3	**94.0**	94.0	94.7	**95.3**	93.3	93.3	**94.0**
led7	**73.8**	68.4	66.7	**73.8**	70.9	70.0	**73.7**	70.0	68.7
letRecog	**48.2**	28.6	16.4	**58.8**	54.2	48.5	**52.6**	42.7	38.6
mushroom	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
pageBlocks	**91.2**	90.6	90.6	**91.2**	90.7	90.7	**91.2**	90.7	90.7
penDigits	**84.3**	68.0	48.1	**88.4**	87.4	84.2	**84.6**	74.6	63.0
pima	**74.6**	74.5	74.5	**75.1**	**75.1**	**75.1**	**74.6**	74.5	74.5
soybean	**89.5**	85.6	81.7	90.0	90.3	**90.6**	**89.8**	87.3	87.1
wine	**92.1**	90.4	88.8	**92.7**	84.3	80.3	**92.1**	**92.1**	90.4
zoo	**94.1**	93.1	93.1	**94.1**	93.1	93.1	**94.1**	**94.1**	**94.1**
Average	**82.3**	78.4	75.8	**83.2**	80.7	79.4	**82.7**	80.6	79.5

differences between different methods pairwisely. In Wilcoxon signed-ranks test, d_i denotes the classification accuracy difference on the i-th of N datasets. We then rank the difference d_i according to their absolute values, if ties occur, average ranks are assigned. Next, the sum of ranks R^+, R^- are calculated on datasets which the second classifier outperforms the first classifier and the first classifier outperforms the second classifier, respectively:

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i)$$
$$R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i)$$

Let T be the smaller value of these two sums, when $N \geq 20$, Wilcoxon W statistic tends to form a normal distribution, then we can use z-value to evaluate the null hypothesis that there is no statistical difference between these two classifiers. The z-score is:

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}}$$

If $z < -1.96$ then the corresponding p-value is smaller than 0.05, therefore, the null hypothesis is rejected.

A series of Wilcoxon signed-ranks test from Table 4.1, Table 4.3, Table 4.4 and Table 4.5 are listed in Table 4.6. It shows the count of wins, losses, ties and corresponding p-value for pairwise post-hoc comparisons. **Rows 2-6** show the difference between the proposed SigDirect algorithm with 5 other well-known rule based and associative classifiers. SigDirect is significant better than C4.5, FOIL, CBA and CPAR and is as good as CMAR. From **Rows 7-8**, we can see the difference between three different classification heuristics is not statistically significant, but since S2 gets a more higher average classification accuracy, we choose to use S2 in the classification phase. **Rows 9-12** list the number of CARs differences between SigDirect, CBA, CMAR before and after rule pruning phase. SigDirect gets a significantly smaller number of CARs in the rule generation phase when measured against CBA and CMAR, the number of CARs is still significantly smaller than CMAR even after the rule pruning phase. The effect of the instance-centric rule pruning strategy is shown in **Rows 13-15**, when classification heuristics S1 and S3

are used, the instance-centric method is significantly better than the database coverage method. Although the difference is not statistically significant with S2, the corresponding p-value is still very close to 0.05 and the instance-centric strategy wins 15 time and only loses 4 times. Therefore, the instance-centric rule pruning strategy is better than the database coverage method. The last 6 rows compares different classification heuristics, the "S" category is much better than the "B" and "A" category. In this way, to classify a new instance, we should choose to sum up the measure M of multiple matching rules to make a final prediction.

Table 4.6: Statistical analysis of Table 4.1, Table 4.3, Table 4.4 and Table 4.5 ; (*) indicates statistically significant difference.

row	comparisons	wins	losses	ties	p-value
2	SigDirect(S2) vs. C4.5*	17	2	1	0.001
3	SigDirect(S2) vs. FOIL*	13	6	1	0.040
4	SigDirect(S2) vs. CBA*	14	5	1	0.033
5	SigDirect(S2) vs. CMAR	12	5	3	0.136
6	SigDirect(S2) vs. CPAR*	13	6	1	0.010
7	SigDirect(S2) vs. SigDirect(S1)	10	4	6	0.158
8	SigDirect(S2) vs. SigDirect(S3)	11	5	4	0.214
9	#bef. prun: SigDirect vs. CBA*	18	2	0	0.004
10	#bef. prun: SigDirect vs. CMAR*	18	2	0	0.006
11	#aft. prun: SigDirect vs. CBA	9	11	0	0.435
12	#aft. prun: SigDirect vs. CMAR*	19	1	0	0.001
13	S1: instance-cetric vs. db coverage*	15	3	2	0.001
14	S2: instance-cetric vs. db coverage	15	4	1	0.056
15	S3: instance-cetric vs. db coverage*	15	2	3	0.008
16	SigDirect(S1) vs. SigDirect(B1)*	16	2	2	0.001
17	SigDirect(S1) vs. SigDirect(A1)*	15	3	2	0.001
18	SigDirect(S2) vs. SigDirect(B2)*	13	5	2	0.006
19	SigDirect(S2) vs. SigDirect(A2)*	15	3	2	0.001
20	SigDirect(S3) vs. SigDirect(B3)*	15	1	4	0.001
21	SigDirect(S3) vs. SigDirect(A3)*	16	2	2	0.001

4.6 Integrating Negative Classification Association Rules

In Definition 4, the significance level of the CAR $X \rightarrow c_k$ can be calculated as:

$$p_F(X \rightarrow c_k) = \sum_{i=0}^{min\{\sigma(X,\neg c_k)\sigma(\neg X,c_k)\}} \frac{\binom{\sigma(X)}{\sigma(X,c_k)+i}\binom{\sigma(\neg X)}{\sigma(\neg X,\neg c_k)+i}}{\binom{|\mathcal{D}|}{\sigma(c_k)}}$$

Similarly, the negative dependency between itemset X and class label c_k is measured by:

$$p_F(X \rightarrow \neg c_k) = \sum_{i=0}^{min\{\sigma(X,\neg c_k)\sigma(\neg X,c_k)\}} \frac{\binom{\sigma(X)}{\sigma(X,\neg c_k)+i}\binom{\sigma(\neg X)}{\sigma(\neg X,c_k)+i}}{\binom{|\mathcal{D}|}{\sigma(c_k)}} \tag{4.4}$$

where $\neg c_k$ indicates the absence of class label c_k. It can be observed that from the notion of p_F-value that $p_F(X \rightarrow \neg c_k) = p_F(\neg X \rightarrow c_k)$, therefore, it is enough to only consider the negative CARs in the form of $X \rightarrow \neg c_k$.

The statistically significant negative CARs generation phase is very similar to the phase for the positive CARs. What we only need to do is substituting all c_k to $\neg c_k$ in Algorithm 4, and all negative statistically significant CARs in the form of $X \rightarrow \neg c_k$ will be generated. It is obvious that the rule generation for the positive and negative CARs can also be integrated together in Algorithm 7.

Although the rule generation phase is similar when the negative CARs are considered, the rule pruning phase cannot be directly used for the negative CARs. In the rule pruning step, both instance centric and database coverage method try to find a matching relationship between a training instance o and a CAR r, the relationship holds only if $r.antecedents \subseteq o.antecedents$ and $r.class = o.class$. It is easy for a positive CAR to find a matching instance. In contrast, for the negative CARs in the form of $X \rightarrow \neg c_k$, it is impossible to find a matching instance in the training dataset. To reduce the number of negative CARs, we propose a simple but effective rule pruning strategy.

The idea is similar to database coverage, we first scan through the set of discovered negative CARs. For each negative CAR $X \rightarrow \neg c_k$, if it misclassifies at least one training instance, in other words, if we find an instance t in the training dataset such that $X \subseteq t.antecedent$ and $c_k = t.class$, the negative CAR $X \rightarrow \neg c_k$ is pruned, otherwise, it is kept.

For the positive CARs $X \rightarrow c_k$, we first rank them by their confidence values, and then use the database coverage method to select a subset of high quality. The database coverage method is used instead of the instance centric method. The reason is that database coverage takes a greedy way to remove instances covered by rules and generates a smaller number of rules compared to the instance centric method. Therefore, it gives more chances for the negative rules to pop up and affect the classification phase.

Here a problem arises, in some datasets, the number of left negative CARs may be much larger than the number of positive CARs. In the extreme case if only negative CARs $X \rightarrow \neg c_k$ are left, it is still hard to make a prediction for a new instance like XY, the only information obtained is that class label c_k is not correct. Therefore, we still wish the positive CARs dominate the classification decision phase, while taking negative CARs as a complement. In this case, we adjust the number of negative CARs, making it at most as large as the number of positive CARs. To be more specific, let n_{neg} and n_{pos} denote the number of pruned positive and negative CARs respectively, if $n_{neg} > n_{pos}$, only the first n_{pos} negative CARs and all positive CARs are chosen as the actual classifier. The whole process is illustrated in Algorithm 8.

The set of statistically significant positive and negative CARs left from the previous rule pruning phase represents the actual associative classifier. Given a new unlabeled object, the classification process searches for the set of CARs that are relevant to this object, and makes the prediction according to the label information of all these relevant rules. Here we discuss how to make the predictions for new objects based on the set of rules in the classifier. There are two types of CARs in our classifier: positive CARs in the form of $X \rightarrow c_k$ and negative CARs in the form of $X \rightarrow \neg c_k$. These two types of CARs are both considered in our classification phase. According to the results from previous section, the classifier works best when the classification heuristic is S2, classifying an unlabeled instance by the summation of confidence values of multiple matching rules. Therefore, we still use the S2 method when the negative CARs are integrated. It is obvious that the confidence values of positive CARs $X \rightarrow c_k$ are added to the class c_k. However, the

negative CARs $X \rightarrow \neg c_k$ is treated differently, we choose to subtract their confidence values from the total confidence of the corresponding class c_k. The detailed description of the classification phase is presented in Algorithm 9.

4.7 Experiments with Negative Classification Association Rules

We evaluate the proposed associative classifier which integrates the negative CARs on the same 20 datasets as Section 4.5.1. To have a fair comparison, we list the classification results of 5 rule based and associative classifiers C4.5 [47], FOIL [46], CBA [40], CMAR [38] and CPAR [62] as before, we also show the classification accuracy of ARC-PAN [8], which is also built on positive and negative CARs. In ARC-PAN, the minimum support is 1%, the minimum confidence is 50% as other associative classifiers. The confidence margin and correlation threshold is 0.1 and 0.5 in ARC-PAN, respectively. The parameter setting of other 5 classifiers follow the settings in Section 4.5.1.

4.7.1 Classification Accuracy

The experimental results are shown in Table 4.7. **Columns 2-3** show the number of classes and number of records for each dataset. **Columns 4-9** list the classification accuracies of C4.5, FOIL, CBA, CMAR, CPAR and ARC-PAN. **Columns 10** shows the performance when only positive CARs are used, while **Columns 11** lists the classification result when the negative CARs are also taken into account.

As can be observed, rules+- gets the best overall classification performance (82.7%) and wins 5 out of 20 datasets, followed by rules+ (82.1%). Both of their average classification accuracy outperform that of C4.5, FOIL, CBA, CMAR, CPAR and ARC-PAN.

4.7.2 Effect of Negative Classification Association Rules

To validate the effect of negative CARs in the associative classifier, we compare rules+- with rules+. The average classification accuracy is higher when negative

CARs are included. We also compare the count of wins and losses for rules+- when compared with rules+, rules+- wins 12 times and only loses twice. It indicates the power of negative CARs. They indeed help us get more reliable and more accurate classification results on most datasets.

4.7.3 Effect of Negative Rule Pruning Method

In the rule pruning phase, due to the absence of negative rule pruning strategies in literature, we propose a simple but effective way, it prunes positive and negative CARs separately. We compare the classification performance of three different scenarios: prune both positive and negative CARs, prune only positive CARs and without rule pruning. In Table 4.8, **Columns 2-4** show the classification results of these three scenarios. The average accuracy of **Column 2** (prune both positive and negative CARs) is the highest and it wins 18 out of 20 datasets. Therefore, the simple but effective rule pruning method not only reduces the number of CARs in the classifier, but also improves the classification performance compared to the associative classifier pruning only positive CARs and the associative classifier without rule pruning phase.

4.7.4 Statistical Analysis

We use the same method as Section 4.5.5 to analysis the results from Table 4.7 and Table 4.8. First, the Frideman test is applied on Table 4.7 to compare the differences between 8 different classifiers. The null hypothesis is rejected which indicates there is a significant difference between these 8 classifiers. Then, we conduct pairwise Wilcoxon signed-ranks test to compare the difference between these different classifiers. The Wilcoxon signed-ranks test is also applied on Table 4.8 to show the difference between different rule pruning strategies. The results are listed on Table 4.9. **Rows 2-7** show the comparisons of our associative classifier rules+- with the other 6 well established classifiers. Our associative classifier always wins more than half of all 20 datasets, but the only strong conclusion we draw is that our method is significantly better than C4.5 and ARC-PAN. ARC-PAN is an associative classifier most similar to our method which also uses the negative CARs, however,

it fails to consider the statistical dependency of the discovered CARs. The statistically significant difference between our method and ARC-PAN is very appealing. It shows the power of introducing statistical dependency in the associative classification problem. Through **Row 8**, we can find that when the negative CARs are included, the associative classifier is significantly better than that with only positive CARs. **Rows 9-10** indicate the effect of the proposed rule pruning strategy, the difference between pruning only positive CARs and without pruning is not statistically significant although pruning only positive rules wins 14 times. But when we also prune negative CARs, the classification performance is greatly improved, the p-value is very small ($p = 0.001$). Thus, by applying this rule pruning strategy, we get a much better classifier with higher classification accuracy and fewer CARs.

Data: Transaction Dataset $\mathcal{D}$, set of antecedent items $\mathcal{I}$, class labels C, significance level $\alpha = 0.05$.
Result: Statistically significant positive and negative CAR sets $\mathcal{R}_{pos}$ and $\mathcal{R}_{neg}$.
Prune impossible antecedent items I with Theorem 2;
$\mathcal{I}_{rest}$: the arranged and renamed antecedent item set;
Create root node and level-1 nodes;
Set $l = 2$;
for *each candidate 1-set CAR r* **do**
  if *r is* PSS **then**
    if $p_F(r) \leq \alpha$ **then**
      if *r is minimal* **then**
        r.minimal = true;
      end
      if *r.class is positive* **then**
        $\mathcal{R}_{pos}$.add(r);
      else
        $\mathcal{R}_{neg}$.add(r);
      end
    end
  else
    prune CAR r from the enumeration tree;
  end
end
while $l \leq |\mathcal{I}_{rest}|$ **do**
  for *each candidate l-set CAR r* **do**
    if *all parent rules of r are PSS and not minimal* **then**
      if $p_F(r) \leq \alpha$ **then**
        if *r is non-redundant* **then**
          if *r is minimal* **then**
            r.minimal = true;
          end
          if *r.class is positive* **then**
            $\mathcal{R}_{pos}$.add(r);
          else
            $\mathcal{R}_{neg}$.add(r);
          end
        end
      end
    else
      prune CAR r and all its decedent rules from the enumeration tree;
    end
  end
  $l = l + 1$;
end

Algorithm 7: Statistically significant positive and negative CARs generation.

Data: Set of positive and negative CARs $\mathcal{R}_{pos}$, $\mathcal{R}_{neg}$ from rule generation phase.
Result: Pruned CARs set $\mathcal{R}_{newpos}$ and $\mathcal{R}_{newneg}$.
Ranking $\mathcal{R}_{pos}$ and $\mathcal{R}_{neg}$ according to confidence values;
$\mathcal{R}_{newpos} = \emptyset$, $\mathcal{R}_{newneg} = \emptyset$;
// 1. Negative CARs pruning
for *each CAR r in* $\mathcal{R}_{neg}$ **do**
  for *each training instance* t *in training dataset* $\mathcal{D}$ **do**
    if r.*antecedent* $\subseteq$ t.*antecedent and* r.*class* = t.*class* **then**
      $\mathcal{R}_{neg}.remove(r)$;
      break;
    end
  end
end
Assign $\mathcal{R}_{neg}$ to $\mathcal{R}_{newneg}$;
// 2. Positive CARs pruning
for *each CAR r in* $\mathcal{R}_{positive}$ **do**
  for *each training instance* t *in training dataset* $\mathcal{D}$ **do**
    if r.*antecedent* $\subseteq$ t.*antecedent and* r.*class* = t.*class* **then**
      $\mathcal{R}_{newpos}.add(r)$;
      remove instances covered by r in $\mathcal{D}$;
    end
  end
end
// 3. Negative CARs set adjustment
if $|\mathcal{R}_{newneg}| > |\mathcal{R}_{newpos}|$ **then**
  $\mathcal{R}_{newneg}$ = first $|\mathcal{R}_{newpos}|$ rules in $\mathcal{R}_{newneg}$;
end

Algorithm 8: Positive and negative CARs pruning.

Data: A new instance o to be classified. Set of positive CARs $\mathcal{R}_{newpos}$ and negative CARs $\mathcal{R}_{newneg}$ from rule pruning phase.
Result: Class label of the new instance o.
$T_{pos} = \emptyset$; `// set of positive rules matching` o
$T_{neg} = \emptyset$; `// set of negative rules matching` o
for *each CAR r in* $\mathcal{R}_{newpos}$ **do**
  if r.*antecedent* $\subseteq$ o.*antecedent* **then**
    T_{pos}.add(r);
  end
end
for *each CAR r in* $\mathcal{R}_{newneg}$ **do**
  if r.*antecedent* $\subseteq$ o.*antecedent* **then**
    $r.confidence = -r.confidence$;
    T_{neg}.add(r);
  end
end
Divide T into n subsets by class labels: $T_1, T_2, ..., T_n$;
for *each subset* $T_1, T_2, ..., T_n$ **do**
  sum up the confidence values of matching CARs in each class
end
Assign the class with the highest sum of confidence value to o;

Algorithm 9: Classification phase with negative CARs integrated.

Table 4.7: Classification accuracy comparison when negative CARS are integrated.

Dataset	#cls	#rec	C4.5	FOIL	CBA	CMAR	CPAR	PAN	rules+	rules+-
adult	2	48842	78.8	**84.6**	84.2	81.3	77.3	83.1	83.2	83.2
anneal	6	898	76.7	**98.8**	94.5	90.7	95.1	86.9	88.0	91.9
breast	2	699	91.5	89.3	**94.1**	89.9	93.0	89.4	91.3	91.3
cylBands	2	540	69.1	74.1	76.1	**76.5**	70.0	42.2	72.0	72.2
flare	9	1389	82.1	83.8	84.2	**84.3**	63.9	83.4	83.2	83.2
glass	7	214	65.9	66.5	68.4	71.1	64.9	48.5	72.0	**72.4**
heart	5	303	**61.5**	55.2	57.8	56.2	53.8	58.8	56.8	57.4
hepatitis	2	155	**84.1**	77.8	42.2	79.6	75.5	39.8	83.2	83.9
horseColic	2	368	70.9	**83.4**	78.8	82.3	81.2	81.5	76.7	75.8
ionosphere	2	351	84.6	86.6	32.5	**91.5**	88.9	83.7	85.0	82.3
iris	3	150	91.3	94.0	93.3	94.0	94.7	94.6	94.7	**94.7**
led7	10	3200	**73.9**	60.5	73.1	73.2	71.3	59.5	73.3	73.3
letRecog	26	20000	50.4	50.0	32.5	28.3	58.2	26.9	61.8	**64.9**
mushroom	2	8124	92.8	99.5	46.7	**100.0**	98.5	98.9	**100.0**	**100.0**
pageBlocks	5	5473	92.0	92.4	90.9	90.1	**92.5**	89.9	91.1	91.2
penDigits	10	10992	70.5	84.1	**92.3**	87.4	80.5	79.5	90.3	91.3
pima	2	768	71.7	71.9	**74.6**	74.4	74.0	74.2	67.7	68.0
soybean	19	683	60.3	88.0	89.2	88.1	83.1	81.8	89.6	**90.5**
wine	3	178	75.8	88.2	49.6	**92.7**	88.2	89.3	88.2	91.6
zoo	7	101	91.0	93.1	40.7	93.0	**94.1**	86.1	93.1	**94.1**
Average			76.7	81.1	69.8	81.3	79.9	73.9	82.1	**82.7**

Table 4.8: Comparison of rule pruning strategies when negative CARs are integrated.

Dataset	rules+-		
	prune +-	prune +	w/o prune
adult	**83.2**	**82.3**	81.9
anneal	**91.9**	65.7	86.7
breast	**91.3**	87.0	81.0
cylBands	**72.2**	63.7	63.7
flare	**83.2**	78.0	76.6
glass	**72.4**	60.7	69.6
heart	57.4	60.1	**59.7**
hepatitis	**83.9**	82.6	81.3
horseColic	**75.8**	73.1	72.0
ionosphere	**82.3**	75.7	75.5
iris	**94.7**	94.0	**94.7**
led7	73.3	**74.2**	73.9
letRecog	**64.9**	55.1	52.6
mushroom	**100.0**	97.9	97.7
pageBlocks	**91.2**	90.4	89.8
penDigits	**91.3**	83.8	86.8
pima	**68.0**	65.6	65.1
soybean	**90.5**	78.2	61.6
wine	**91.6**	**91.6**	**91.6**
zoo	**94.1**	77.2	**94.1**
Average	**82.7**	76.9	77.8

Table 4.9: Statistical analysis of Table 4.7 and Table 4.8; (*) indicates statistically significant difference.

row ID	comparisons	wins	losses	ties	p-value
2	rules+- vs. C4.5 *	13	7	0	0.011
3	rules+- vs. FOIL	12	8	0	0.33
4	rules+- vs. CBA	11	9	0	0.24
5	rules+- vs. CMAR	12	7	1	0.42
6	rules+- vs. CPAR	12	6	2	0.058
7	rules+- vs. PAN *	15	5	0	0.014
8	rules+- vs. rules+ *	12	2	6	0.033
9	prune+- vs. prune+ *	17	2	1	0.001
10	prune+- vs. w/o rule pruning *	15	2	3	0.001

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this study, we investigate the problem of using statistically significant dependencies to mine spatial co-location patterns and discover classification association rules for classification. Co-location pattern mining and associative classification are two well-studied problems in the data mining community, most of current methods rely on confounding user defined thresholds to discover frequent patterns and strong rules. However, these set of thresholds are difficult to determine limiting the usage in many applications. Moreover, we show that the spatial co-location patterns and classification association rules discovered by support-confidence framework easily suffer from *type 1* error and *type 2* error which lead to the omission of many interesting and detection of meaningless patterns and rules. To address the limitations of support-confidence framework to make the algorithms less affected by *type 1* error and *type 2* error, we propose to use statistically significant methods in the knowledge discovery process.

By fully exploiting the property of statistical significance, we propose two algorithms CMCStatApriori and SigDirect to address the spatial co-location rule mining and associative classification problem, respectively.

The CMCStatApriori algorithm that we propose is a novel co-location mining algorithm that is able to detect statistically significant co-location rules in datasets with extended spatial objects. A motivation for the development of this algorithm is the problem of mining co-location rules between chemical pollutants and chil-

dren cancer cases in Alberta and Manitoba, Canada. To solve this problem, firstly, we build buffers around each spatial object and then impose grids over the geographic space. From the grid points that intersect with multiple spatial features, we then derive a spatial transaction dataset. In this way, the co-location pattern mining problem is transformed to the problem of mining statistically significant association rules in the spatial transaction dataset. In CMCStatApriori, z-score is used to measure those statistical significance of the co-location rules with a fixed consequent spatial feature, it provides an upper bound for the binomial distribution and its monotonically increasing property makes it possible to mine statistically significant co-location rules in an iterative way, iterating between candidate generation and pruning phase. In this way, we do not have to limit the number of antecedent features up to a certain number which is considered a major limitation of some previous methods. Therefore, the co-location rules we find are more general and the algorithm scales up very well. Different statistically significant co-location mining algorithms usually lead to different sets of co-location rules. The set of co-location rules is difficult to evaluate even for some domain experts, therefore, we also propose to use a classifier to help evaluate the detected co-location rules. Experimental results demonstrate that our algorithm seems to get more meaningful co-location rules than some other alternatives. In conclusion, in addition to its good generalization and scalability, CMCStatApriori is considered to be a very effective algorithm for statistically significant co-location rule mining problem.

The novel associative classifier, SigDirect, can be viewed as a strong competitor or even slightly winner to some existing rule based and associative classifiers. Unlike existing associative classifiers which require minimum support and minimum confidence thresholds, SigDirect does not need to set any confounding parameters. It extends the recent proposed Kingfisher algorithm, to directly mine CARs that show statistically significant dependencies. It is well known that searching for statistically significant CARs is a very expensive and demanding task, because the number of CARs grows exponentially with the number of antecedent items. We make this search phase possible by using some effective pruning strategies. Since the mined CARs are statistically significant, it alleviates the *type 1* error and *type*

2 error that normally appear in other classification association rule mining methods. Apart from the promising classification performance, the number of CARs before and after rule pruning are both very small, making SigDirect more appealing than other methods when there is little difference in classification performance. The number of CARs before rule pruning is even an order of magnitude smaller than that by CBA and CMAR. After rule pruning, the number of rules are almost always below 100. The small set of rules in both phases makes it possible and practical for users to sift through them to edit and update according to their own needs, which can be very important in many applications. The experiments also show that in the rule pruning phase, the instance centric pruning strategy is more effective than the database coverage method, and in the classification phase, under a measure M, summation effect performs better than average effect and classifying just based on the best matching rule. When the measure M is the confidence, the classification performance is the best. We also investigate the problem of integrating statistically significant negative CARs into our SigDirect algorithm, the experimental results seem very encouraging. A novel rule pruning strategy is also proposed to reduce the number of positive and negative CARs separately. The classification performance of the new classifier can be seen as a competitor to some well known classification methods. Besides, by considering the negative CARs, the classification accuracy indeed increases compared to the classifier built with only positive classification association rules. A two step statistical test are used to validate all these conclusions.

5.2 Future Work

Future work can be focused in these possible directions:

1. In CMCStatApriori, we generate the complete set of statistically significant co-location rules. The number of co-location rules will be very large if a loose z-score threshold is used, and a large subset of them is considered to be redundant, i.e. adding no new information. In the future, we may consider to detect only non-redundant co-location rules to reduce the number of dis-

covered co-location rules. In addition, CMCStatApriori uses the z-score to search for statistically significant co-location rules because z-score provides an upper bound for the binomial distribution, more research can be done to find a more tight upper bound for a more accurate result. Another limitation of CMCStatApriori is we still need to set the z-score threshold. In this study, we present a possible way to select the threshold by evaluating the classification accuracy, how to automatically select the threshold more effectively for each dataset is another very challenging problem.

2. Possible future work on SigDirect can be focused in these two aspects: firstly, we expect to derive more effective pruning strategies in the rule generation phase; another interesting but challenging task is to find more effective rule pruning strategies to prune negative CARs post rule generation to improve the classification performance.

Bibliography

[1] AgroClimatic Information Service (ACIS). Live alberta weather station data. `http://www.agric.gov.ab.ca/app116/stationview.jsp`.

[2] Aibek Adilmagambetov, Osmar R Zaïane, and Alvaro Osornio-Vargas. Discovering co-location patterns in datasets with extended spatial objects. In *Proceedings of the 15th International Conference on Data Warehousing and Knowledge Discovery (DaWaK)*, pages 84–96, 2013.

[3] Charu C Aggarwal and Philip S Yu. A new framework for itemset generation. In *Proceedings of the 7th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 18–24, 1998.

[4] Charu C Aggarwal and Philip S. Yu. Mining associations with the collective strength approach. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):863–873, 2001.

[5] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (SIGMOD*, pages 207–216, 1993.

[6] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference Very Large Data Bases (VLDB)*, pages 487–499, 1994.

[7] M-L Antonie and Osmar R Zaïane. Text document categorization by term association. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM)*, pages 19–26, 2002.

[8] M-L Antonie and Osmar R Zaïane. An associative classifier based on positive and negative rules. In *Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD)*, pages 64–69, 2004.

[9] M-L Antonie, Osmar R Zaïane, and Robert C Holte. Learning to use a learned model: A two-stage approach to classification. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM)*, pages 33–42, 2006.

[10] Maria-Luiza Antonie and Osmar R Zaïane. Mining positive and negative association rules: an approach for confined rules. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 27–38, 2004.

[11] Bavani Arunasalam and Sanjay Chawla. Cccs: a top-down associative classifier for imbalanced class distribution. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 517–522, 2006.

[12] K. Bache and M. Lichman. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2013.

[13] Elena Baralis and Paolo Garza. A lazy approach to pruning classification rules. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM)*, pages 35–42, 2002.

[14] Sajib Barua and Jörg Sander. Sscp: mining statistically significant co-location patterns. In *Proceedings of the 12th International Symposium on Spatial and Temporal Databases (SSTD)*, pages 2–20, 2011.

[15] Sajib Barua and Jörg Sander. Mining statistically significant co-location and segregation patterns. *IEEE Transactions on Knowledge and Data Engineering*, 26(5):1185–1199, 2014.

[16] Yves Bastide, Nicolas Pasquier, Rafik Taouil, Gerd Stumme, and Lotfi Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. In *Proceedings of the 1st International Conference on Computational Logic (CL)*, pages 972–986, 2000.

[17] Roberto J Bayardo Jr. Brute-force mining of high-confidence classification rules. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 123–126, 1997.

[18] Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 265–276, 1997.

[19] Environment Canada. National Pollutant Release Inventory. Tracking Pollution in Canada. `http://www.ec.gc.ca/inrp-npri/`.

[20] Loïc Cerf, Dominique Gay, Nazha Selmaoui, and Jean-François Boulicaut. A parameter-free associative classification method. In *Proceedings of the 10th International Conference on Data Warehousing and Knowledge Discovery (DaWaK)*, pages 293–304, 2008.

[21] F. Coenen. The LUCS-KDD software library. `http://cgi.csc.liv.ac.uk/~frans/KDD/Software/`, 2004.

[22] William W Cohen. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning (ICML)*, pages 115–123, 1995.

[23] Chris Cornelis, Peng Yan, Xing Zhang, and Guoqing Chen. Mining positive and negative association rules from large databases. In *Proceedings of the 2006 IEEE Conference on Cybernetics and Intelligent Systems (CIS)*, pages 1–6, 2006.

[24] Environment Canada. National Climate Data and Information. Canadian climate normals or averages 1971-2000. `http://climate.weatheroffice.gc.ca/climate_normals/index_e.html`.

[25] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[26] Richard O Duda, Peter E Hart, et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[27] ESRI. ArcGIS Desktop: Release 10, 2011.

[28] W Hämäläinen and M Nykanen. Efficient discovery of statistically significant association rules. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*, pages 203–212, 2008.

[29] Wilhelmiina Hämäläinen. Efficient discovery of the top-k optimal dependency rules with fisher's exact test of significance. In *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM)*, pages 196–205, 2010.

[30] Wilhelmiina Hämäläinen. Statapriori: an efficient algorithm for searching statistically significant association rules. *Knowledge and Information Systems*, 23(3):373–399, 2010.

[31] Wilhelmiina Hämäläinen. Kingfisher: an efficient algorithm for searching for both positive and negative dependency rules with statistical significance measures. *Knowledge and Information Systems*, 32(2):383–414, 2012.

[32] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 1–12, 2000.

[33] Yan Huang, Jian Pei, and Hui Xiong. Mining co-location patterns with rare events from spatial data sets. *Geoinformatica*, 10(3):239–260, 2006.

[34] Yan Huang, Shashi Shekhar, and Hui Xiong. Discovering colocation patterns from spatial data sets: a general approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(12):1472–1485, 2004.

[35] Yan Huang, Hui Xiong, Shashi Shekhar, and Jian Pei. Mining confident co-location rules without a support threshold. In *Proceedings of the 2003 ACM symposium on Applied Computing (SAC)*, pages 497–501, 2003.

[36] Yun Sing Koh and Russel Pears. Efficiently finding negative association rules without support threshold. In *Proceedings of the 20th Australian Joint Conference on Artificial Intelligence (AI)*, pages 710–714, 2007.

[37] Jiuyong Li. On optimal rule discovery. *IEEE Transactions on Knowledge and Data Engineering*, 18(4):460–471, 2006.

[38] Wenmin Li, Jiawei Han, and Jian Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM)*, pages 369–376, 2001.

[39] Tjen-Sien Lim, Wei-Yin Loh, and Yu-Shan Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine learning*, 40(3):203–228, 2000.

[40] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 80–86, 1998.

[41] Bing Liu, Wynne Hsu, and Yiming Ma. Pruning and summarizing the discovered associations. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 125–134, 1999.

[42] Shinichi Morishita and Akihiro Nakaya. Parallel branch-and-bound graph search for correlated association rules. In *Proceedings of the 5th ACM SIGKDD Workshop on Large-Scale Parallel KDD Systems (KDD Workshop)*, pages 127–144, 1999.

[43] Shinichi Morishita and Jun Sese. Transversing itemset lattices with statistical metric pruning. In *Proceedings of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 226–236, 2000.

[44] Siegfried Nijssen, Tias Guns, and Luc De Raedt. Correlated itemset mining in roc space: a constraint programming approach. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 647–656, 2009.

[45] Siegfried Nijssen and Joost N Kok. Multi-class correlated pattern mining. In *Proceedings of the 4th International Workshop on Knowledge Discovery in Inductive Databases (KDD Workshop)*, pages 165–187, 2005.

[46] J Ross Quinlan and R Mike Cameron-Jones. Foil: A midterm report. In *Proceedings of the 1993 European Conference on Machine Learning (ECML)*, pages 1–20, 1993.

[47] John Ross Quinlan. *C4.5: Programs for Machine Learning*, volume 1. Morgan kaufmann, 1993.

[48] Ashok Savasere, Edward Omiecinski, and Shamkant Navathe. Mining for strong negative associations in a large database of customer transactions. In *Proceedings of the 14th International Conference on Data Engineering (ICDE)*, pages 494–502, 1998.

[49] Shashi Shekhar and Yan Huang. Discovering spatial co-location patterns: A summary of results. In *Proceedings of the 7th International Symposium on Spatial and Temporal Databases (SSTD)*, pages 236–256, 2001.

[50] Wei-Guang Teng, Ming-Jyh Hsieh, and Ming-Syan Chen. On the mining of substitution rules for statistically dependent items. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM)*, pages 442–449, 2002.

[51] Wei-Guang Teng, Ming-Jyh Hsieh, and Ming-Syan Chen. A statistical framework for mining substitution rules. *Knowledge and Information Systems*, 7(2):158–178, 2005.

[52] Dhananjay R Thiruvady and Geoff I Webb. Mining negative rules using grd. In *Proceedings of the 8th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*, pages 161–165, 2004.

[53] Florian Verhein and Sanjay Chawla. Using significant, positively associated and relatively class correlated rules for associative classification of imbalanced datasets. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)*, pages 679–684, 2007.

[54] Hao Wang, Xing Zhang, and Guoqing Chen. Mining a complete set of both positive and negative association rules from large databases. In *Proceedings of the 12th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*, pages 777–784, 2008.

[55] Jianyong Wang and George Karypis. Harmony: Efficiently mining the best rules for classification. In *Proceedings of the 2005 SIAM International Conference on Data Mining (SDM)*, pages 205–216, 2005.

[56] Geoffrey I Webb. Discovering significant rules. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 434–443, 2006.

[57] Geoffrey I Webb. Discovering significant patterns. *Machine Learning*, 68(1):1–33, 2007.

[58] Geoffrey I Webb and Songmao Zhang. K-optimal rule discovery. *Data Mining and Knowledge Discovery*, 10(1):39–79, 2005.

[59] Xindong Wu, Chengqi Zhang, and Shichao Zhang. Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems*, 22(3):381–405, 2004.

[60] Xiangye Xiao, Xing Xie, Qiong Luo, and Wei-Ying Ma. Density based co-location pattern discovery. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS)*, page 29, 2008.

[61] Hui Xiong, Shashi Shekhar, Yan Huang, Vipin Kumar, Xiaobin Ma, and Jin Soung Yoo. A framework for discovering co-location patterns in data sets with extended spatial objects. In *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM)*, 2004.

[62] X Yin and J Han. Cpar: Classification based on predictive association rules. In *Proceedings of the 3rd SIAM International Conference on Data Mining (SDM)*, pages 331–335, 2003.

[63] Jin Soung Yoo and Shashi Shekhar. A joinless approach for mining spatial co-location patterns. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1323–1337, 2006.

[64] Jin Soung Yoo, Shashi Shekhar, and Mete Celik. A join-less approach for co-location pattern mining: A summary of results. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM)*, pages 813–816, 2005.

[65] Jin Soung Yoo, Shashi Shekhar, John Smith, and Julius P Kumquat. A partial join approach for mining co-location patterns. In *Proceedings of the 12th annual ACM International Workshop on Geographic Information Systems (GIS)*, pages 241–249. ACM, 2004.

[66] Mohammed J Zaki. Generating non-redundant association rules. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 34–43, 2000.

[67] Xin Zhang, Nikos Mamoulis, David W Cheung, and Yutao Shou. Fast mining of spatial collocations. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 384–393, 2004.