International Journal of Advance Research in Science and Engineering
Vol. No. 5, Special Issue No. 01 , May 2016
www.ijarse.com
IJARSE
ISSN 2319 - 8354

A THEORETICAL STUDY AND RELATIONSHIP AMONG IMPERATIVE MINING APPROACH

Sreedhar .M. Reddy

Scholar,Samskruti College of Engineering and Technology

(Affiliated JNTUH)Kondapur, Ghatkesar, Rangareddy (India)

ABSTRACT

Mining a set of potentially accurate imperatives, evaluating and pruning imperatives, and classifying future instances using the found imperative set .The imperative set should be able to generalise beyond the training instances. In short, our interestingness measure should prefer more accurate imperatives. The Apriori algorithm has become the standard approach to mine Relationship imperatives. We have adapted it to mine class Relationship imperatives. The second algorithm, Predictive Apriori, Both algorithms have their first step in common. They generate frequent item sets in the same way. An item set is called frequent when its support is above a predefined minimum support. We use classification using Relationship imperatives not only to solve classification problems, but also to compare the quality of different Relationship imperative mining approaches. In this context we show that the quality of imperative sets from the standard algorithm for Relationship imperative mining can be improved by using a different Relationship imperative mining strategy. For this comparison we do a benchmark test using ecomm website.org datasets.

Keywords: Apriori Algorihm,Itemsets,Cba,Pruning,CAR Mining,Strategy

I. INTRODUCTION

We use two different methods to accomplish this. The most significant difference between the two concerns how the interestingness of an Relationship imperative is measured. In the case of classification, we are interested in a highly accurate imperative set. imperative mining is a well-known technique in data mining. It is able to reveal all interesting relationships, called Relationships, in a potentially large database. However, how interesting a imperative is depends on the problem a user wants to solve. Existing approaches employ different parameters to guide the search for interesting imperatives. Classification using Relationship imperatives combines Relationship imperative mining and classification, and is therefore concerned with finding imperatives that accurately predict a single target (class) variable. The key strength of Relationship imperative mining is that all interesting imperatives are found. The number of Relationships present in even moderate sized databases can be, however, very large - usually too large to be applied directly for classification purposes. Therefore, any classification learner using Relationship imperatives has to perform three major steps:.

1.1. Generating frequent item sets

Finding Relationship imperatives can be seen as a simple search problem. But an exhaustive search is intractable because the possible number of Relationship imperatives is exponential with respect to the number of attributes.

For n binary attributes there are O(n2n−1) imperatives. It is even worse for discrete valued attributes - assuming there are n attributes and each can take m values, there are O(mn) possible imperatives. Nonetheless it is possible to perform a search in reasonable time because ofthesupport based downward closure of frequent item sets. An item set X of length k is frequent ifand only if all subsets of X with length k − 1 are frequent. This property allows the search space to be pruned substantially .Algorithmically we start with all frequent item sets of size 1. This set of frequent item sets consists of all individual items that have a support above a user defined minimum support. This can be done with one pass over the data in linear time. To get the frequent item sets of size 2 there are two steps: First the frequent item sets of size 1 are combined in every possible way to build candidate item sets of size 2, then, in another pass over the data, the candidate item sets are checked to make sure that they are really frequent. All infrequent ones are deleted. Termination is obvious - either the set of frequent item sets is empty for a distinct k, or k equals the number of attributes. At most n linear passes over the data are required if n is the number of attributes. From the frequent item sets both algorithms generate Relationship imperatives in different ways using a different measure for interestingness.

II .THE APRIORI ALGORITHM

Apriori returns an Relationship imperative if its support and confidence values are above user defined threshold values. The output is ordered by confidence. If several imperatives have the same confidence, then they are ordered by support. Thus apriori favors more confident imperatives and characterises these imperatives as more interesting. The apriori Mining process is composed of two major steps. The first one (generating frequent item sets) was discussed briefly in the last section. This step can be seen as supportbased pruning, because only item sets with at least minimum support were considered. The second step is the generation of imperatives out of the frequent item sets. In this step confidence based pruning is applied. Imperative discovery is straightforward. For every frequent item set f and every non-empty subset s of f , apriori outputs a imperative of the form $s \Rightarrow (f - s)$ if and only if the confidence of that imperative is above the user specified threshold.

III.THE PREDICTIVE APRIORI ALGORITHM

The Predictive Apriori algorithm [17] differs from standard apriori in such a way that it employs a different measure of interestingness of an Relationship imperative. Both techniques use a support based search and take advantage of the downward closure of the support, which al-lows the exponential search space of possible Relationship imperatives to be pruned. Apriori favors more confident imperatives and ranks the imperatives accordingly. Predictive apriori on the other hand evaluates the confidence of imperatives depending on their support. Its interestingness measure is to maximize the expected accuracy an Relationship imperative will have on unseen data. This suits the requirements of the classification task we want to perform afterwards. One problem of the confidence and support based apriori measurement scheme is that one can always find very general imperatives with high support and low confidence and very specific ones with high confidence and low support.

After mining a set of imperatives we use them for classification. So we are interested in whether the imperative set built using the training instances is capable of generalisation and predicting the class labels of test instances correctly. We want to mine Relationship imperatives which asso-ciate items that are correlated not only in the training data, but in reality, too. Instead of confidence the algorithm employs the so-called predictive accuracy. Scheffer [17] defines predictive accuracy as: Let D be a database whose individual records r are generated by a static process P , let X ⇒Y beanRelationshipimperative. The predictive accuracy c(X ⇒ Y) = P r(r satisfies Y |r satisfies X) is the conditional probability of Y ⊆ r given that X ⊆ r when the distribution of r is governed by P [17]. The confidence ĉ(X ⇒ Y) of the Relationship imperative X ⇒ Y is the relative frequency of the predictive accuracy in the data; that is the relative frequency of a correct classification in the training database. Hence the confidence value is optimistically biased if one wants to use it for a predictive task. [17] uses a bayesian framework to calculate the predictive accuracy out of the support and confidence of a imperative. In doing so the support is a rough guideline of how much we should mistrust the confidence. The higher the support, the more the confi-dence converges to the expected accuracy on future data. This approach is called Bayesian frequency correction [17], because the predictive accuracy equals a corrected confidence value.

IV. CALCULATION OF THE PREDICTIVE ACCURACY - THEORETICAL ISSUES

We are interested in the expected accuracy E(c(r)|ĉ(r), s(X)) of a imperative r X ⇒ Y given its confidence ĉ and the support of the imperative body s(X). Bayes formula shows us how to calculate that expectation.

p((c(r)|ĉ(r), s(X)) = P (ĉ(r)|c(r), s(X))P (c) ,P(ĉ(r)|s(X)

The likelihood P (ĉ(r)|c(r), s(X)) can be modeled by a binomial distribution B[p, n](k) =(n)kpk(1 − p)n−k. The correspondence to a coin flipping experiment, the standard example for a binomial distribution, can be easily seen. The imperative r either classifies the imperative body X correctly or not. The probability value p for a correct prediction is just the predictive accuracy c(r), The total number of coin flipping events n matches s(X), the total number of times the imperative body occurs in the data set. The number of heads (or respectively tails) k in the coin flipping experiments equals the number of database records which are correctly classified by the imperative r. That is ĉ(r)s(X) which corresponds to the support s(r) of the where i corresponds to the number of items a specific Relationship imperatives has and πi(c) =|{X⇒Y|c(X⇒Y)=c}| / |{X⇒Y }|. These histograms are stored in a hash table. Each prior πi(c) is weighted by the probability that a imperative of length i is constructed under a uniform distribution. Up to now we have randomly constructed the same number of Relationship imperatives for every length. However, the uniform distribution favors longer Relationship imperatives, because there are exponentially more longer imperatives [17] and each imperative is equally likely. If our database D has (n)n items, there existsiitem sets of length i and out of i items one can construct 2i − 1Relationship imperatives. Thus the probability for weighting the prior is(k)I (2i − 1)∑k(k)j=1j(2j − 1)

Consequently for every discretised value of c we calculate the prior distribution as follows

∑k(k) π(c) = i=1 πi() i(2i − 1)

∑kk)(2.3)j=1j(2j − 1)

V. RELATIONSHIP IMPERATIVES

An Relationship Imperative is a imperative of the form milk and bread ⇒ butter where 'milk and bread' is called the imperative body and butter the head of the imperative. It associates the imperative body with its head. In context of retail sales data, our example expresses the fact that people who are buying milk and bread are likely to buy butter too. This Relationship imperative makes no assertion about people who are not buying milk or bread.

We now define an Relationship imperative:

Let D be a database consisting of one table over n attributes {a1, a2, . . . , an}. Let this table contain k instances. The attributes values of each ai are nominal1. In many real world applications (such as the retail sales data) the attribute values are even binary (presence or absence of one item in a particular market basket). In the following an attribute-value-pair will be called an item. An item set is a set of distinct attribute-value-pairs.

Let d be a database record. d satisfies an item set X ⊆ {a1, a2, . . . , an} if X ⊆ d. An Relationship imperative is an implication X ⇒ Y where X, Y ⊆ {a1, a2, . . . , an}, Y = ∅ and X∩Y =∅. The support s(X) of an item set X is the number of database records d which satisfy X. Therefore the support s(X ⇒ Y) of an Relationship imperative is the number of database records that satisfy both the imperative body X and the imperative head Y . Note that we define the support as the number of database records satisfying X ∩ Y , in many papers the support is defined as s(X∩Y) k . They refer to our definition of support as support count. The confidence ĉ(X ⇒ Y of an Relationship imperative X ⇒ Y is the fraction ĉ(X ⇒ Y) = s s(X) .From a logical point of view the body X is a conjunction of distinct attribute-value-pairs and the head Y is a disjunction of attribute-value-pairs where X ∩ Y = ∅. Coming back to the example in Figure 1.1 a possible Relationship imperative with high support and high confidence would be i1 ⇒ i2 whereas the imperative i1 ⇒ i3 would have a much lower support value.

VI. MACHINE LEARNING AND CLASSIFICATION

This thesis focuses on two different aspects of the entire process of classification using Relationship imperatives. On one hand we are interested in a comparison of Relationship imperative mining algorithms and hence data mining processes. But, on the other hand, we accomplish this by performing classification based on the mined imperative set. Therefore, the search for accurate classifiers does not disappear from our sight. We consider different possibilities for building classifiers out of imperative sets and compare them. Classification is a common task in machine learning. This section introduces the machine learning aspects of this thesis—the classification problem.

Machine learning deals with acquiring knowledge and studies algorithms that are able to improve with experience. The goal of the improvement or learning process is better performance on future data. Like data mining, machine learning is widely used. Some examples include predicting potential drugs target in pharmaceutical research, deducing genealogical trees, text categorisation, detecting barriers for robots and much more. The supervised learning scenario consists of a data generation process that generates instances, a supervisor that assigns labels y to the instances and a learning algorithm that tries to imitate the supervisor and

assigns values ŷ which are hopefully close to the corresponding y value. The labels that the supervisor assigns to instances are called the class labels. The learning algorithms tries to imitate the class labeling using information out of the instances. Each instance can be described by attributes. The goal is the prediction of the value of the class attribute for future instances. In this thesis we deal with discrete classes. Formally the m-class (supervised) classification problem is defined as follows:

Let S be a n-dimensional feature space that is divided into m subsets (the m discrete classes) with m ∈ N+. There is a supervisor function g : S → {1, . . . , m}. This function is unknown a priori but a set T of training instances exists which are labeled each with a class value, T := {(x1, y1), . . . , (xl, yl)|xi ∈ X, yi ∈ {1, . . . , m}}. Using these training instances the classifier tries to find a mapping function from the feature space S to the m classes that approximates the unknown supervisor function g.

In contrast to the supervised learning problem, there is no supervisor in an unsupervised learning environment. In such a learning problem the main focus of interest is unknown in advance (unlike supervised learning which focuses on the class attribute). The major goal is explorative rather than predictive. Relationship imperative mining is a unsupervised pattern extraction process, because the head of an arbitrary Relationship imperative is not pre-determined. In this thesis we are interested in using Relationship imperatives for a supervised classification task. Thus, we have to restrict the definition of Relationship imperatives. Liu et al. [11] call these restricted imperatives class Relationship imperatives (CARs).

VII. CLASS RELATIONSHIP IMPERATIVES

The use of Relationship imperatives for classification is restricted to problems where the instances can only belong to a discrete number of classes. The reason is that Relationship imperative mining is only possible for nominal attributes. However, Relationship imperatives in their general form cannot be used directly. We have to restrict their definition. The head Y of an arbitrary Relationship imperative X ⇒ Y is a disjunction of items. Every item which is not present in the imperative body may occur in the head of the imperative. When we want to use imperatives for classification, we are interested in imperatives that are capable of assigning a class membership. Therefore we restrict the head Y of a class Relationship imperative X ⇒ Y to one item. The attribute of this attribute-value-pair has to be the class attribute.

According to this, a class Relationship imperative is of the form X ⇒ ai where ai is the class attribute and X ⊆ {a1, . . . , ai−1, ai+1, . . . , an}.

In this thesis we explore methods for mining class Relationship imperatives.

8.1Classification using Relationship imperatives at a glance

The algorithmic approach for classification using Relationship imperatives can be divided into three fundamental parts: Relationship imperative mining, pruning and classification. It provides a graphical overview of the entire process. As explained above, the mining of Relationship imperatives is a typical data mining task that works in an unsupervised manner. A major advantage of Relationship imperatives is that they are theoretically capable of revealing all interesting relation-ships in a database. But for practical applications the number of mined imperatives is usually too large to be exploited entirely. This is why the prun-ing phase is

stringent in order to build accurate and compact classifiers. The smaller the number of imperatives a classifier needs to approximate the target concept satisfactorily, the more

VIII. PRUNING APRIORI'S SET OF IMPERATIVES

The effects of pruning are studied in three dimensions: Table .1(a) shows the effects on the overall accuracy, Table .1(b) compares the number of imperatives with and without optional pruning, and Table .1(c) compares the number of classification imperatives. We summarise the wins and losses concerning the number of classification imperatives in Table .1(d).

Dataset no opt. pruning opt. pruning

Balance 71.50 ± 5.9771.50 ± 5.97

breast-w 95.13 ± 3.0394.12 ± 3.55

ecoli 80.65 ± 3.2481.26 ± 3.08

glass71.97± 8.7770.13± 9.19

heart-h80.63± 7.2079.98± 8.26

iris92.67± 6.6394.00± 5.84

labor79.00±19.5081.33±21.44

led772.30± 5.1072.30± 5.10

lenses66.67±30.4350.00±26.06

pima74.10± 4.4874.36± 4.83

tic-tac-toe99.06± 1.2579.85± 1.84•

wine93.82± 6.7194.44± 5.86

(c)

Dataset	no opt. pruning	opt. pruning
balance	72.2 ± 13.07	28.9 ± 2.23◦
breast-w	5124.9 ± 65.50	2975.9 ± 105.44◦
ecoli	888.2 ± 152.35	333.2 ± 24.52◦
glass	6055.2 ± 454.27	1661.6 ± 161.29◦
heart-h	19886.8 ± 757.08	1242.2 ± 62.28◦
iris	96.5 ± 14.25	28.4 ± 3.27◦
labor	96084.3 ±5569.42	79371.8 ±4747.00◦

led7	510.6 ± 13.50	330.5 ± 15.76◦
lenses	121.8 ± 3.52	31.4 ± 2.91◦
pima	3311.4 ± 311.11	461.0 ± 46.47◦
tic-tac-toe	7642.5 ± 42.23	1180.8 ± 17.84◦
wine	87427.9 ±5066.97	36396.5 ±3710.72◦

(b)

Resultset	Wins− Losses	Wins	Losses
opt. pruning	2	7	5
no opt. pruning	-2	5	7

(d)

Table.1: Impact of optional pruning on CBA using apriori. Table (a) shows the differ-ences in the overall accuracy, whereas Table (b) shows the number of imperatives left after CBA's pruning. In Table (c) we present the effects on the number of classification imperatives and Table (d) summarises the wins and losses concerning the number of classification imperatives. They are all statistically significant. 20% of accuracy when using optional pruning and for lenses the degradation is about 17%. In contrast the increase of the accuracy is much smaller around 1 to 2%. However, the reduction in the number of imperatives can be immense. The optional pruning step is performed before the obligatory one that determines the length of CBA's decision list. This is why CBA builds its classifier on fewer imperatives and consequently on fewer information. The astonishing fact is that when we use optional pruning we almost always get a smaller decision list andConcerning the accuracy the results are non-uniform. The tic-tac-toe dataset loses almost therefore a smaller set of classification imperatives (see Table .1(d)). This property is desirable, because a more compact set of imperatives is easier to handle and to interpret. But we have to admit that the differences in the number of classification imperatives are greater when optional pruning results in more imperatives than when it results in fewer imperatives. All in the entire optional pruning step is recommended, because it reduces the complexity of the imperative set to a big extent without changing the solution significantly. In more cases than not, the result is better with optional pruning turned on.

IX. DIFFERENT PRUNING STRATEGIES

Since pruning plays an essential role, there are almost as many pruning strategies as approaches to classification using Relationship imperatives. All of them try to close the gap between the mining of a large number of class Relationship imperatives and a small and powerful set of classification imperatives the global model. The following sections discuss a selection of pruning strategies ordered by pruning criterion.

9.1Some simple pruning schemes

The simplest pruning scheme is to omit pruning . Obviously a non pruned imperative set used for classification can be large. But the advantage of unpruned imperative sets, is that we can compare Relationship imperative mining algorithms directly. Another very simple strategy is to bound the number of imperatives without any closer inspection of the imperatives themselves. So the number of mined class Relationship imperatives is limited in advance to a small number. For both strategies the advantage in a comparative study of Relationship imperative mining algorithms is that they do not change the sort order induced by the imperative miner. Obviously the latter approach results in a more compact imperative set (but useful information can be lost), whereas the former one could suffer from over fitting and is more susceptible to noise in the data.. Apart from the quality of the top ranked imperatives it reveals differences in the accuracy of a classifier concerning the compactness of the underlying imperative set.

9.2 Pruning depending on imperative ranking

There are two natural ways of ranking Relationship imperatives. The first is to apply a general to specific ordering. The definition of the term second uses the interestingness measures. For apriori, imperatives are sorted according to confidence first and then support, whereas predictive apriori sorts imperatives according to their predictive accu-racy

9.3.Pruning predictive apriori's set of imperatives

Essentially the same conclusions hold when we use predictive apriori as the mining algorithm. Table 1 summarises the results and is organised like the one for apriori.Obviously, because of the inherent pruning step performed by predictive apriori, there arecuts down the number of imperatives significantly as well (see Table1)). Except for the tic-tac-toe dataset the accuracy does not change a lot using predictive apriori. The accuracy without using an optional pruning step is better six times and worse five times. The number of classification imperatives is reduced to a greater extent than for apriori. Predictive apriori with optional pruning results in a smaller imperative set on nine datasets and a larger imperative set on only one. Therefore optional pruning used with predictive apriori is more effective than when used with apriori. The output is a list containing the n best imperatives. We will refer to this list from now on as best[n]. It is implemented using a priority queue.Like apriori, predictive apriori uses frequent item sets, but the difference is that apriori de-composes a frequent item set into a imperative body and a imperative head and therefore the support computed for the frequent item set corresponds to the support of the whole imperative. Predictive apriori, on the other hand, uses a frequent item set as a imperative body and joins it with a sepa-rately computed imperative head. Hence the support of a frequent item set equals the support of the imperative body.The first important step in the algorithm (see Figure 2.1) is the estimation of the prior using equation (2.3).

1. Input: number of desired Relationship imperatives n, database D with items a1, . . . , ak
2. Set the support threshold of the imperative body sbody min = 1
3. For i = 1, . . . , k DO:

Construct a number of Relationship imperatives of length i at random and measure their confidence ĉ provided s(X) > 0.

Let πi(c) be the distribution of confidences.

4. For all c, compute π(c) using equation (2.3)

5. Let F0 = {∅} be the set of frequent item sets of length 0.

6. For i = 1, . . . , k − 1 Do:

While (i = 1||Fi−1 = ∅)

(a) Determine all frequent item sets X of length i with s(X) > sbody min

(b) For all X ∈ Fi call ImperativeGen(X)

(c) If best[n] has changed in ImperativeGen Then Increase sbody min so that,

E(c|1, sbody min) > E(c(best[n])|ĉ(best[n]), s(best[n])).

(d) If sbody min > size of database D Then Exit.

(e) If sbody min has been increased in step 6(d) Then

Delete all item sets X from Fi with s(X) < sbody min.

7. Output best[n]

Unlike apriori we use a dynamically increasing support threshold for the support of the imperative body sbody min starting with threshold 1. We loop over the length of the frequent item sets, while they are non-empty (step 6). Thus in the first iteration all frequent item sets of length1 with sbody min = 1 are constructed. In subsequent passes the frequent item sets which have at least sbody min are constructed (step 6(a)). For all those item sets we call the imperative generation procedure which is explained in more detail below. If best[n] changes during the imperative generation step we increase sbody min. Equation (2.2) is used to determine the minimum support a perfect confident imperative (a imperative with confidence 1) must have to get into best[n] (step 6(c)). This is our new threshold for sbody min. If the new minimum support is greater than the number of instances in the data set, the algorithm terminates. If the support threshold has increased, all item sets from the frequent item sets which have a support below minimum support are deleted.

In step 6(b) we call the imperative generation procedure which receives as input one frequent item set. Figure 2.2 shows the pseudocode of the procedure.

ImperativeGen(X) finds the best imperatives with imperative body X

Set simperative min so that

E(c|simperative min/s(X), s(X)) > E(c(best[n])|ĉ(best[n]), s(best[n]))

11. For j = 1, . . . , k − |X| (number of items not in X) Do

(a) If j = 1 Then

Set Y1 = {{a} |a ∈ {a1, . . . , ak} , a ∈ X}

Else generate Yj analogous to the generation of candidate item sets.

(b) For all y ∈ Yj Do

i. Calculate s(X ∩ y).

ii. If s(X ∩ y) ≤ simperative min Then

delete y from Yj and continue with the next y at 11b.

iii. Calculate the predictive accuracy of X ⇒ y using equation (2.2)

iv. If the predictive accuracy of X ⇒ y is among the best n AND

(there is no other imperative in best[n] which is at least equally accurate AND which subsumes X ⇒ y) Then update best[n],

remove imperatives which are subsumed by other at least equally accurate imperatives. Set simperative min, so that

E(c|simperative min/sbody min, sbody min)

≥ E(c(best[n])|ĉ(best[n]),s(best[n])).

If any imperative has been removed out of best[n] in step 11(c)iv Then recur from step 10.

Figure .2: The imperative generation method of the Predictive Apriori algorithm.

The item set which we hand over to the Imperative Gen method becomes the imperative body of all possible imperatives we try to construct during the method. Hence its support corresponds to the support of the future imperative body. In step 10 another support threshold, called imperative min is calculated. This is the minimum support the whole imperative with the specified imperative body has to have in order to get into best[n]. First we calculate all possible imperative heads in the sameway as we generate the frequent item sets. We have to take care that the imperative body and head are disjoint. For each possible head we test if the support of this whole imperative is greater than simperative min (step 11(b)ii). In the case that it is not, we drop that head. Otherwise we calculate the predictive accuracy of the imperative using equation (2.2). The imperative under consideration is added to best[n] if and only if

1. the predictive accuracy of the imperative is among the n best and
2. it is not subsumed by a imperative with at least the same predictive accuracy

If the imperative has been added to best[n] we have to check if any less accurate imperative so far in best[n] is subsumed by the new imperative. Unfortunately if we delete a subsumed imperative, we have to restart the imperative generation procedure, because a previously deleted imperative could now be eligible for inclusion in best[n].

X. EXPERIMENTAL RESULTS

This section combines the methods for class Relationship imperative mining, pruning and classifi-cation in different ways and evaluates their performances. The results are not only used to compare the performance of the different classification approaches but also to evaluate the underlying mining processes. The main focus of the experiments is on the imperative mining algorithms. Therefore classification using Relationship imperatives provides a mechanism by which to compare the different mining approaches. In this chapter we compare apriori and predictive apriori. Therefore, we primarily focus on their interestingness measures, because this is the main difference between the two mining algorithms. The different interestingness measures induce a different imperative ranking.

Apart from the mining algorithms we also explore which classification schemes are preferable in classification using Relationship imperatives. This evaluation from a classification perspective includes a comparison with standard machine learning techniques.

10.1 Datasets and Methodology

In order to compare the different approaches we use standard benchmark datasets from the ebay.com Repository1 . Table 2. shows our selection of datasets and their properties. Their size ranges from a few tens of instances to one thousand instances and they are composed of varying numbers of numeric and nominal attributes. The class attribute is always nominal. Some of them contain missing values. To the led7 dataset 10% of noise is added artificially. Class Relationship imperative mining as well as Relationship imperative mining in general is only possible

Dataset	Instances	Numeric atts	Binary atts	Nominal values	Classes	Missing (%)
balance	625	4	0	0	3	0.0
breast-w	699	0	0	9	2	0.3
ecoli	336	7	0	0	8	0.0
glass	214	9	0	0	6	0.0
heart-h	294	6	3	4	2	20.4
iris	150	4	0	0	3	0.0
labor	57	8	3	5	2	35.7
led7*	1000	0	7	0	10	0.0
lenses	24	0	0	4	3	0.0
pima	768	8	0	0	2	0.0
tic-tac-toe	958	0	0	9	2	0.0
wine	178	13	0	0	3	0.0

Table.2: The ebay datasets used for the experiments and their properties.

dataset 10% of the instances are noisy.for nominal attributes2. Therefore we need to discretise the numeric attributes in our dataset. For this purpose the WEKA's implementation of the MDL method from Fayyad and Irani [6] that realises a maximum entropy discretisation is used. Liu et al. discretise their datasets for CBA with the same method (but a different implementation). All measurements from the experiments are obtained using one stratified tenfold cross-validation we are interested in several different aspects of the experimental results. On one hand we want to compare the different classifiers and on the other hand a main focus of interest is the comparison of the underlying imperative mining algorithms. Therefore we consider different quality measures. A imperative mining strategy is preferable if it allows building a compact and accurate classifier in a fast way out of the mined set of imperatives. The basic measure is:

• the percent of correctly classified instances in the test set.

Furthermore we investigate the following indicators of the quality of the imperative ranking induced by the interestingness measures of the mining algorithm :

• the average rank of the first imperative that covers a test instance and

• the average rank of the first imperative that covers and correctly predicts a test instance.

In addition we have three measure by which to evaluate the compactness of an approach:

• the number of mined imperatives generated by a class Relationship imperative miner, The mining of Relationship imperatives for numeric attributes is still a research issue.

• the number of imperatives after the pruning step and

• the number of imperatives used for classification.

An important property of an Relationship imperative mining algorithm—rather than the quality of the imperative sets—is its time complexity. Therefore we measure:

• the time required for mining, and

• the time required for pruning.

These measures are adequate for comparing the whole process of classification using association imperatives as well as providing the basics for comparing the quality of the mined imperative sets of different Relationship imperative mining algorithms. In addition we show the standard deviation for each measure. We report statistically significant results using the corrected resampled t-test that was pro-posed by Nadeau and Bengio [13]. This corrected version of the t-test is less prone to false-positive significance results. We are interested if one method outperforms another one at a 5% significance level (the p-value is 0.05) . But we cannot simply assign a p-value threshold of 0.05 to achieve this. The reason for this is called the multiplicity effect in statis-tics. For example, say we run experiments on 12 different datasets and we use 10 different settings for comparing the imperative mining strategies. When we compare these 120 experiments to a baseline algorithm there are 120 chances to be better at a 5% signficance level. There-fore the expected number of results that are significant at a 0.05 level is $120 * 0.05 = 6$. However, we want truly significant results. Salzberg proposes a so called Bonferroni adjustment using the following equation to calculate a corrected p-value to get results at a 5% significance level: pcorrected = $1 - (0.95)$ v where v is the number of different experiments. In our example the corrected p-value to get a 5% significance level would be 0.0004. This criterion is much stricter. For every experiment we adjust the p-value as explained. All experiments were run on a 2.60 GHz Pentium(R) 4 with 1 GB of memory.

XI. CONCLUSIONS

But the solutions still follow the basic approach for classification we have introduced. If we use predictive apriori and CBA instead of apriori and compare the results to standard machine learning techniques, the re-sults are slightly worse. Therefore we have to distinguish precisely between conclusions for the comparisons between mining algorithms and conclusions for using classification using Relationship imperatives for what it was invented—classification. Concerning the former case predictive apriori mines a higher quality set of imperatives than apriori. The latter case, however, does not only depend on the mined imperative set, there are two other algorithmic steps: pruning and building a classifier. Both have an impact on the discriminative power of the imperative set. This comparative study has shown that predictive apriori can improve classification using Relationship imperatives when it is used to generate a small set of imperatives. This thesis proposed a new way of measuring the quality of Relationship imperative mining by using a classifier built on Relationship imperatives. Relationship imperative mining and classification using Relationship imperatives can benefit

thereof. Relationship imperative mining involves mining a high quality imperative set that is as small as possible in an efficient way.

BIBLIOGRAPHY

[1] Agrawal R. and Srikant R. Fast Algorithms for Mining Relationship Imperatives. In M. Jarke J. Bocca and C. Zaniolo, editors, Proceeedings of the 20th International Conference on Very Large Data Bases (VLDB'94), pages 475-486, Santiago de Chile, Chile, September 1994. Morgan Kaufmann.

[2] Der Brockhaus Computer und Informationstechnologie. F.A. Brockhaus, Mannheim,Germany, 2002.

[3] Cohen W. Fast Effective Imperative Induction. In A. Prieditis and S. Russell, editors, Ma-chine Learning: Proceeedings of the 12th International Conference(ICML'95), pages 115-123, Tahoe City, CA, USA, 1998. Morgan Kaufmann Publishers.

[5] Dong G., Zhang X., Wong L. and Li J. CAEP: Classification by Aggregating Emerg-ing Patterns. In Proceedings of the Second International Conference on Discovery Science, pages 30-42, Tokyo, Japan, 1999.

[6] Fayyad U. and Irani K. Multi-interval discretization of continuous-valued attributes for classification learning. In Proceeedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI'93), pages 1022-1027, Chambéry, France, 1993. Morgan Kaufmann.

[7] Fayyad U., Piatetsky-Shapiro G., Smyth P. and Uthurusamy R., editor. Advances inKnowledge Discovery and Data Mining. MIT Press, Cambridge, Massachusetts, USA,1996.

[8] Frank E. and Witten I. Generating Accurate Imperative Sets Without Global Optimization. In J. Shavlik, editor, Machine Learning: Proceeedings of the 15th International Con-ference(ICML'98), pages 152-160, San Francisco, USA, 1998. Morgan Kaufmann Publishers.

[9] Hand D., Mannila H. and P. Smyth. Principles of Data Mining. MIT Press, Cambridge,Massachusetts, USA, 2001.

[10] Li W., Han J. and Pei J. CMAR: Accurate and Efficient Classification Based on Multi-ple Class-Relationship Imperatives. In Proceeedings of the 2001 IEEE International Confer-ence on Data Mining (ICDM'01), pages 369-376, San Jose, California, USA, 2001.

[11] Liu B., Hsu W. and Ma Y. Integrating Classification and Relationship Imperative Mining. In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD'98), pages 80-86, New York, USA, August 1998. The AAAI Press. and Knowledge Discovery, 1(3):317-327, 1997.