Discovery of Hidden Correlations in a Local Transaction Database based on Differences of Correlations

Tsuyoshi TANIGUCHI, Makoto HARAGUCHI, and Yoshiaki OKUBO

Division of Computer Science, Hokkaido University
N-14 W-9, Sapporo 060-0814, Japan
{tsuyoshi, makoto, yoshiaki}@kb.ist.hokudai.ac.jp

Abstract. Given a transaction database as a global set of transactions and its sub-database regarded as a local one, we consider a pair of itemsets whose degrees of correlations are higher in the local database than in the global one. If they show high correlation in the local database, they are detectable by some search methods of previous studies. On the other hand, there exist another kind of paired itemsets such that they are not regarded as characteristic and cannot be found by the methods of previous studies but that their degrees of correlations become drastically higher by the conditioning to the local database. We pay much attention to the latter kind of paired itemsets, as such pairs of itemsets can be an implicit and hidden evidence showing that something particular to the local database occurs even though they are not yet realized as characteristic ones. From this viewpoint, we measure paired itemsets by a difference of two correlations before and after the conditioning to the local database, and define a notion of DC pairs whose degrees of differences of correlations are high. As the measure is non-monotonic, we present an algorithm, searching for DC pairs, with some new pruning rules for cutting off hopeless itemsets. We show by an experimental result that potentially significant DC pairs can be actually found for a given database and the algorithm successfully detects such DC pairs.

1 Introduction

In the studies of data mining from transaction databases, many studies have been paying much attention to finding itemsets with high supports, paired itemsets appeared in association rules with high confidence [1], or paired itemsets with strong correlation [6–9]. These notions are considered useful for distinguishing characteristic itemsets from other ones in a single transaction database. A similar strategy based on the notion of change of supports, known as Emerging Patterns [2, 3], is successful even for finding itemsets characterizing either of two databases. All of the notions about itemsets are thus proposed to extract (paired) itemsets required to be characteristic in a given database or either of a given pair of databases.

However, as has been indicated in the study of Chance Discovery [10], some itemsets not characteristic in the above sense are also useful, as they are *potentially significant* under some condition. For instance, suppose we have a transaction database for supermarkets in a particular area and the database includes the information of ages of customers and goods on sale as items. We here consider the problem of capturing some correlations between some ages and some goods in the database. We regard the correlations as an interest of customers of some ages in some goods. For example, consider a case that degrees of the correlations are not high in a particular area but are very low in a global area including the particular area. The correlation cannnot be found by search methods of previous studies because the degrees of the correlations are not high in both global and particular areas. However, there is a possibility that the customers of the ages are interested in the goods in the particular area more than in the global area by some factor even if the correlations in the particular area are not regarded as characteristic. It may be worth remarking this specific phenomenon as an implicit and hidden evidence in order to consider a new strategy for sale. Moreover, consider a case that the database includes the information of time as item. In the particular area, we can find characteristic correlations with high degree of correlation in time t_1 or t_2 after t_1 by search methods of previous studies. But we may want to know an implicit correlation which may become a characteristic correlation in t_3 after t_2. In short, we want to know customers of some ages start to be interested in some goods. In the case, non-characteristic correlations in t_2 with high degrees of differences of correlations from t_1 to t_2 may be useful.

From the viewpoints mentioned in the above, for a given global database and its local database obtained by a certain conditioning, the purpose of this paper is to present an algorithm for finding pairs of itemsets such that (1) the paired itemsets are not necessarily characteristic, where we say that two itemsets are characteristic in a database if the correlation between them is high, (2) the degrees of correlation become much higher in the local database than ones in the global database. That is, we are going to observe the degrees of difference of correlations before and after the conditioning to the local database. Such a pair of itemsets with high degrees of difference of correlations is called a DC pair. We confirm by an experiment that potentially significant DC pairs can be actually found for a given database.

It is generally a hard problem to find DC pairs, as the degrees of difference of correlations are never monotonic w.r.t. the standard ordering of itemsets, namely the set inclusion. For this reason, we consider a restricted problem under given two parameters, ζ and ϵ. More precisely speaking, we evaluate the degrees of difference of correlations by a function defined with ζ and ϵ and restrict DC pairs we try to find. Then, we prove that a monotone property over itemsets can be observed in the mining of DC pairs depending on ζ or ϵ. Based on this monotonic property, we can design some pruning rules for cutting off hopeless itemsets X and Y not satisfying the constraints of DC pairs.

1.1 Related Works and Paper Organization

There exist many works in the field of data mining that are based on a strategy of contrasting two or more databases in order to extract significant properties or patterns from a huge data set. Particularly, data mining techniques, known as contrast-set mining [2–5], have been designed specifically to identify differences between databases to be contrasted.

For instance, in the study of Emerging Patterns [2, 3] for two transaction databases, itemsets whose supports are significantly higher in one database than in another one are considered significant, as they can be candidate patterns for distinguishing the former from the latter. A similar strategy is also used in the system STUCCO [4] in order to obtain characteristic itemsets in one database based on χ^2 test. In addition, the system, Magnum Opus [5], examines relations between itemsets and a database among several databases. On the other hand, what this paper tries to find are paired itemsets whose correlations drastically increase in one database. Thus we can say that the subject of this paper is a kind of "contrast-set mining of correlations between itemsets".

Secondly, many methodologies have been proposed to detect characteristic correlations in a single database [6–8]. In these studies, using some function measuring the degree of correlation between itemsets, strongly correlated itemsets in a given database or in one database from given two databases are examined. Thus, these methods are also used to discover itemsets or family of itemsets that are characteristic in one database. On the other hand, the algorithm presented in this paper is designed so as to find even paired itemsets whose correlation in one database is not significantly high but is significantly higher than correlation in another database. Our algorithm may find the characteristic paired itemsets as special cases, but is never supposed to find only characteristic ones. To find these paired itemsets, we present in this paper some new pruning rules so that the algorithm successfully detects even non-characteristic paired itemsets.

Finally, several notions about correlations have been proposed and used in the above previous studies from information theoretic or statistical viewpoints, then we describe our standpoint that we use a measure to evaluate correlations. If we need to consider even negative events that itemsets do not appear in transactions, the notion of correlations based on χ^2-test shall be taken into account. But this paper is based on the notion of self mutual information without taking log to measure positive relationships between events that itemsets occur.

The rest of this paper is organized as follows. The next section defines some terminologies used throughout this paper. In Section 3, we introduce the notion of DC pairs and define our problem of mining DC pairs. An algorithm for finding DC pairs is described in Section 4. Section 5 presents our experimental results. In the final section, we summarize our study and discuss future work.

2 Preliminaries

Let $\mathcal{I} = \{i_1, i_2, \cdots, i_n\}$ be a set of *items*. An *itemset* is a subset of $\mathcal{I}$. A *transaction database* $\mathcal{D}$ is a set of transactions, where a transaction is an itemset.

We say that a transaction t *contains* an itemset X, if $X \subseteq t$. For a transaction database $\mathcal{D}$ and an itemset X, the *occurrence* of X over $\mathcal{D}$, denoted by $O(X, \mathcal{D})$, is defined as $O(X, \mathcal{D}) = \{t | t \in \mathcal{D} \wedge X \subseteq t\}$, and the *probability* of X over $\mathcal{D}$, denoted by $P(X)$, is defined as $P(X) = |O(X, \mathcal{D})|/|\mathcal{D}|$.

For an itemset C, a *sub-database* of $\mathcal{D}$ w.r.t. C, denoted by $\mathcal{D}_C$, is defined as the set of transactions containing C in $\mathcal{D}$, that is, $\mathcal{D}_C = O(C, \mathcal{D})$. The *complement* of $\mathcal{D}_C$ w.r.t. $\mathcal{D}$ is denoted by $\overline{\mathcal{D}_C}$ and is defined as $\overline{\mathcal{D}_C} = \mathcal{D} - \mathcal{D}_C$.

For itemsets X and Y, the *correlation* between X and Y over a transaction database $\mathcal{D}$, $correl(X, Y)$, is defined as $correl(X, Y) = P(X \cup Y)/P(X)P(Y)$. For a sub-database $\mathcal{D}_C$, the correlation between X and Y over $\mathcal{D}_C$, $correl_C(X, Y)$, is given by $correl_C(X, Y) = P(X \cup Y|C)/P(X|C)P(Y|C)$, where $P(X|C) = P(X \cup C)/P(C)$. Note here that correlations are defined for only itemsets X whose supports in $\mathcal{D}$ and $\mathcal{D}_C$ are non-zero. We regard a pair of X and Y such that $correl(X, Y) > 1$ as characteristic since $P(X|Y) > P(X)$ holds. Notice that $P(Y|X) > P(Y)$ holds, too. Similarly, we regard a pair of X and Y such that $correl(X, Y) \leq 1$ as non-characteristic.

3 DC Pair Mining Problem

In this section, we define a notion of DC pairs and our problem of mining them.

For a pair of itemsets X and Y, we especially focus on "difference of correlations observed by conditioning to the local database". The difference of correlations is measured by the following ratio:

$$change(X, Y; C) = \frac{correl_C(X, Y)}{correl(X, Y)} = \frac{P(C)P(C|X \cup Y)}{P(C|X)P(C|Y)}. \tag{1}$$

Let $\rho (> 1)$ be an admissible degree of difference of correlations. In our framework, a pair of itemsets X and Y is considered significant if $change(X, Y; C) \geq \rho$ holds. Since we assume C is given by users, $P(C)$ can be regarded as a constant. Therefore, the change is actually evaluated with the following function g:

$$g(X, Y; C) = \frac{P(C|X \cup Y)}{P(C|X)P(C|Y)}. \tag{2}$$

A pair of itemsets X and Y is called a *DC pair* if $g(X, Y; C) \geq \rho/P(C)$. We try to find all DC pairs efficiently. It should be noted here that the function g behaves *non-monotonically* according to expansion of itemsets X and Y. So we cannot apply a simple pruning method like one Apriori adopted [1]. Therefore, we approximate the above problem according to the following naive strategy:

> Find pairs of X and Y which give higher values of $P(C|X \cup Y)$, keeping the values of $P(C|X)$ and $P(C|Y)$ small.

With a new parameter ζ ($0 \leq \zeta \leq 1$), our approximated problem is precisely defined as follows:

Definition 1. DC Pairs Mining Problem

Let C be an itemset for conditioning. Given ρ and ζ, DC pair mining problem is to find any pairs of X and Y such that $P(C|X \cup Y) > \zeta$, $P(C|X) < \epsilon$ and $P(C|Y) < \epsilon$, where $\epsilon = \sqrt{\zeta \cdot P(C)/\rho}$.

4 Algorithm for Finding DC Pairs

In this section, we present an algorithm to solve the DC pair mining problem. In Section 3, by using parameters ζ and ϵ, we restrict DC pairs we try to find. But, $P(C|Z)$ behaves *non-monotonically* according to expansion of an itemset Z as well as g. This means that there is a possibility that we have to examine all itemsets in database since a simple pruning method cannot be used. Then, we prove some pruning rules by considering the problem of mining DC pairs in top-down manner. Therefore, in this paper, we explain an algorithm which candidates Z for compound itemsets of DC pairs such that $P(C|Z) > \zeta$ are found at first in top-down manner. In order to examine itemsets in top-down manner, we firstly enumerate maximal itemsets in the local database $\mathcal{D}_\mathcal{C}$ because $P(Z|C) > 0$ must hold. After all, the computation for mining DC pairs is divided into two phases:

Phase1: Identifying Candidates for Compound Itemsets
An itemset Z such that $P(C|Z) > \zeta$ is identified as a candidate itemset from which DC pairs X and Y are obtained as $Z = X \cup Y$.

Phase2: Dividing Compound Itemsets
Each candidate Z is divided into two itemsets X and Y such that $Z = X \cup Y$, $P(C|X) < \epsilon$ and $P(C|Y) < \epsilon$.

In the algorithm, there is a case that some candidate Z may not be decomposable. Therefore, we consider checking the possibility for Z to be divided into some DC pair. Then, we first describe a basic enumeration schema, and then introduce more refined one taking the decomposability into account.

4.1 Pruning Search Branches by Dropping Items

For each maximal itemset Z_{max} found in $\mathcal{D}_C$, we first examine Z_{max}, then its proper subsets are examined, and so on. During this search, we can prune useless branches (itemsets) based on the following observation.

Let Z be an itemset containing an item i. Suppose that there exists a subset Z' of Z such that $i \in Z' \subset Z$ and $P(C|Z') > \zeta$. Since $P(C|Z') = P(C)P(Z'|C)/P(Z') > \zeta$, $P(Z'|C) > \zeta \cdot P(Z')/P(C)$ holds. Therefore, $P(i|C) \geq P(Z'|C) > \zeta \cdot P(Z')/P(C) \geq \zeta \cdot P(Z)/P(C)$. As the result, we have $P(C \cup i) > \zeta \cdot P(Z)$. This means that if $P(C \cup i) \leq \zeta \cdot P(Z)$ holds, then we cannot obtain any subset Z' of Z containing i such that $P(C|Z') > \zeta$. That is, assuming Z as a search node in Phase1, if $P(C \cup i) \leq \zeta \cdot P(Z)$ holds, any immediate subset of Z containing i does not have to be examined. Therefore, we can safely drop i from Z.

Dropping Items:
For a search node (itemset) Z and an item $i \in Z$, if $P(C \cup i) \leq \zeta \cdot P(Z)$, any

subset Z' containing i never be a child node of Z in our top-down construction process. In other words, any child node consists of only items in Z that are not dropped.

As a special case, if any item $i \in Z$ is dropped, we do not need to examine any subset of Z.

Termination Condition :
For a search node (itemset) Z, if $\max\{P(C \cup i) | i \in Z\}/\zeta \leq P(Z)$ holds, then Z does not have to be expanded further.

The termination condition provides a theoretical lower bound of our search in Phase1. Since $i \in Z$ and $P(Z|C) > 0$, then $P(i|C) > 0$ holds. Therefore, we can obtain the following

Lower Bound of Search in Phase1 :
If a search node Z is visited in Phase1, then $P(Z) \leq maxp_\zeta$, where $maxp_\zeta = \max\{P(C \cup i) | P(i|C) > 0\}$. In other words, any search node Z whose probability exceeds $maxp_\zeta$ never be generated in Phase1.

4.2 Pruning Search Branches Based on Decomposability

The pruning mechanism just discussed above can become more powerful by taking some constraint in Phase2 into account. More concretely speaking, we can perform the operation of "Dropping Items" more frequently.

In Phase2, each candidate Z found in Phase1 is divided into two itemsets X and Y such that $P(C|X) < \epsilon$ and $P(C|Y) < \epsilon$. Similar to the above discussion, for any $i \in X \cup Y (= Z)$, $P(C \cup Z) < \epsilon \cdot P(i)$ holds. Therefore, if there exists an item $i \in Z$ such that $P(C \cup Z) \geq \epsilon \cdot P(i)$, Z cannot be divided into two parts satisfying the constraint on ϵ. In other words, such an item i never be a member of adequate two parts. Therefore, i can be dropped from Z. Thus, we can obtain a revised operation on search nodes which is more powerful.

Dropping Items (Revised):
For a search node Z and an item $i \in Z$, if $P(C \cup i) \leq \zeta \cdot P(Z)$ or $P(i) \leq P(C \cup Z)/\epsilon$, i can be dropped from Z.

According to it, a new termination condition and a new theoretical lower bound is given as follows:

Termination Condition (Revised):
For a search node Z, if $\max\{P(C \cup i) | i \in Z\}/\zeta \leq P(Z)$ or $\max\{P(i) | i \in Z\} \leq P(C \cup Z)/\epsilon$, then Z does not have to be expanded further.

Lower Bound of Search in Phase1 (Revised):
If a search node Z is visited in Phase1, then $P(Z) \leq maxp_\zeta$ and $P(C \cup Z) \leq \epsilon \cdot maxp_\epsilon$ holds, where $maxp_\epsilon = \max\{P(i) | P(i|C) > 0\}$.

4.3 Another Termination Condition in Phase1

In this section, we show another lower bound in Phase1 by taking the complement $\overline{\mathcal{D}_\mathcal{C}} = \mathcal{D} - \mathcal{D}_C$ into account. We expect this lower bound stop expanding search nodes before Dropping Items start to work.

Suppose a DC pair of X and Y is obtained from an itemset Z, that is, $Z = X \cup Y$. Then, $P(C|Z) = |O(Z, \mathcal{D}_{\mathcal{C}})|/(|O(Z, \mathcal{D}_{\mathcal{C}})| + |O(Z, \overline{\mathcal{D}_{\mathcal{C}}})|) > \zeta$ and $P(C|X) = |O(X, \mathcal{D}_{\mathcal{C}})|/(|O(X, \mathcal{D}_{\mathcal{C}})| + |O(X, \overline{\mathcal{D}_{\mathcal{C}}})|) < \epsilon$. Then it follows that $|O(Z, \mathcal{D}_{\mathcal{C}})| > \frac{\zeta}{1-\zeta}|O(Z, \overline{\mathcal{D}_{\mathcal{C}}})|$ and $|O(X, \mathcal{D}_{\mathcal{C}})| < \frac{\epsilon}{1-\epsilon}|O(X, \overline{\mathcal{D}_{\mathcal{C}}})|$. Therefore, $\frac{\zeta}{1-\zeta}|O(Z, \overline{\mathcal{D}_C})| < |O(Z, \mathcal{D}_C)| \leq |O(X, \mathcal{D}_C)| < \frac{\epsilon}{1-\epsilon}|O(X, \overline{\mathcal{D}_C})|$. As a result, we have $|O(Z, \overline{\mathcal{D}_C})| < k(\zeta, \epsilon)|O(X, \overline{\mathcal{D}_C})|$, where $k(\zeta, \epsilon) = \frac{(1-\zeta)\epsilon}{\zeta(1-\epsilon)}$. Furthermore, as $|O(Z, \overline{\mathcal{D}_C})| \leq |\overline{\mathcal{D}_C}| \leq |\mathcal{D}|$, we have $|O(Z, \overline{\mathcal{D}_C})| < k(\zeta, \epsilon)|\mathcal{D}|$. Conversely, if $|O(Z, \overline{\mathcal{D}_C})| \geq k(\zeta, \epsilon)|\mathcal{D}|$, it follows that Z as well as any subset Z' of Z is never decomposable to obtain DC pairs, as $|O(Z', \overline{\mathcal{D}_C})| \geq |O(Z, \overline{\mathcal{D}_C})|$.

Termination Condition based on Complement

If $|O(Z, \overline{\mathcal{D}_{\mathcal{C}}})|/|\mathcal{D}| \geq k(\zeta, \epsilon)$, Z does not need to be expanded further.

In the top-down mining process of DC pairs, we firstly check the above termination condition for the present itemset Z. If the condition does not hold, then we make the next node Z' with the help of the rule of dropping items.

4.4 Dividing Compound Itemsets

In Phase 2, we divide a candidate compound itemset Z into itemsets X and Y such that $Z = X \cup Y$, $X \cap Y = \emptyset$, $P(C|X) < \epsilon$, and $P(C|Y) < \epsilon$. For this purpose, we consider a lattice of itemsets with Z as its greatest itemset, and enumerate $X \subset Z$ in a bottom-up manner, from a singleton itemset to Z, with the following pruning rule.

Dropping Items in Phase 2:

For a search node (itemset) X and an item $i \in X$, if $P(C \cup Z) \leq \epsilon P(i \cup X)$, any superset of X containing i does not need to be expanded further.

The above rule is exactly dual to the rule of Dropping Items in Phase1, and is therefore similarly proved and utilized for cutting off useless branches to next nodes including items that can be dropped.

5 An Experiment

In this section, we present some experimental results on the mining of DC pairs. The main purpose of experiments is to confirm that potentially significant DC pairs can be actually found for a given database.

5.1 Datasets and Implementation

At first, we explain a database we use in our experiment. We carried out the experiments on Entree Chicago Recommendation Data, a family of databases from the UCI KDD Archive (http://kdd.ics.uci.edu). It consists of eight databases each of which contains restaurant features in a region, e.g. Atlanta, Los Angeles, New Orleans and so on in the USA. To examine DC pairs given a particular region to be compared with the whole regions, we consider a new item working as a name for each region, and assign it to every transaction of the corresponding database. By this operation, we have an integrated database of 4160 transactions and 265 items. The items represent various restaurant features as "Italian", "romantic", "parking" and so on. Given the integrated database, we have developed a system written in C for finding DC pairs. All experiments are conducted on 1.5 GHz PentiumIV PC with 896 MB memory.

As we have already explained in Section 4, our top-down search procedure enumerates compound itemsets Z such that $P(C|Z) > \zeta$, starting from maximal itemsets in $\mathcal{D}_{\mathcal{C}}$ and using the pruning rules based on Dropping Items (Revised) and two Termination Conditions.

We carried out a preliminary experiment before the experiment at first. So, we can know our pruning rules are difficult to work well when the size of an itemset examined is long. We describe the reason in 5.2. Therefore, let the purpose of the experiment be to confirm that potentially significant DC pairs can be actually found for a given database and the algorithm successfully detects such DC pairs and to examine a performance of our pruning rules when the size of an itemset examined is short. Moreover, based on the result of the experiment, we examine a possibility of an efficient search of DC pairs.

For the above purpose, we here assume that our search procedure starts from itemsets of shorter length than maximal itemsets in $\mathcal{D}_{\mathcal{C}}$. More precisely speaking, instead of maximal itemsets in $\mathcal{D}_{\mathcal{C}}$, we introduce a family of itemsets such that (1) the lengths are no more than a given size parameter (6 in our experiment) and that (2) they are maximal among all itemsets having non-zero support and satisfying (1), where the order to define the maximality is also based on the set inclusion.

5.2 Experimental Results

Our experimental results are summarized in Figure 1, where ρ is the ratio of $correl_C(X,Y)$ to $correl(X,Y)$, ζ is a parameter in our search strategy, and $|N_{full}|$ is the number of itemsets in $\mathcal{D}_{\mathcal{C}}$ whose sizes are no more than the size parameter. ρ, ζ and size parameter are set for the values 3.0, 0.4 and 6, respectively in our experiment. $|N_{drop}|$ is the number of itemsets actually examined in Phase1, $|P(C|Z) > \zeta|$ denotes the number of itemsets Z such that $P(C|Z) > \zeta$ in $\mathcal{D}_{\mathcal{C}}$ whose sizes are no more than 6, $|DC|$ is the number of detected DC pairs. Finally, $|DC_{NotCor}|$ is the number of DC pairs of itemsets whose degree of correlation is less than or equal to 1.

There exist various kinds of DC pairs in the experimental data. For instance, in New Orleans, a DC pair $X = \{$*Entertainment*, *Quirky*, *Up and Coming*$\}$ and $Y = \{$*\$ 15-\$ 30*, *Private Parties*, *Spanish*$\}$ is found. The pair shows high degree of difference of correlations by conditioning to New Orleans. But since the pair shows very high degree of correlation as a result of its conditioning, the pair can be found by search methods of previous studies. Also, in many cases, such DC pairs show high degrees of correlations in global database in the experiment. In short, such DC pairs may not be worth paying attention to by especially conditioning to New Orleans. On the other hand, there exists a pair $X = \{$*Quirky*$\}$ and $Y = \{$*Good Decor*, *Italian*, *\$15-\$30*, *Good Service*$\}$ in DC pairs in New

Orleans. The pair is not correlated in both global database and local database. Therefore, the pair cannot be found by search methods of previous studies. But the pair shows high degree of difference of correlations by conditioning to New Orleans. In short, the pair shows not high degree of correlation in New Orleans, on the other hand, the pair shows very low degree of correlation in a global database. We pay much attention to such DC pairs. We consider such DC pairs can be useful in some cases. For instance, people who look for a restaurant in New Orleans may be interested in a "quirky Italian restaurant" which is a hidden feature in New Orleans in contrast with a "quirky Spanish restaurant" which is a significant feature in both global and local database because there may be some factor of its high degrees of difference of correlations even if the pair doesn't show high degree of correlation. As we described the above, it is shown that potentially significant DC pairs can be actually found for the given database and our algorithm detects such DC pairs. In addition, various potentially significant DC pairs are found in the experimental data.

As is shown in Figure 1, the number of compound itemsets examined is certainly reduced by the pruning rules in Section 4. Every pruning rule we have presented is theoretically safe in the sense that they cut off some branches only when it is proved that no solution can be reached through the branches. However, the degree of reduction does not seem sufficient to improve the efficiency. We consider the causes as follows.

The first cause is a low chance that our pruning rules can be applied to itemsets examined in our search. By a simple operation of our pruning rules, there is a possibility that we can turn out that many itemsets don't have to be examined. But our pruning rules cannot reduce so many itemsets examined in the experiment because there are not many opportunities that our pruning rules can be applied to the itemsets. So, we analyze a property of our pruning rules. And we can know our pruning rules are difficult to work well when a difference between a probability of an itemset Z and a probability of an item $i \in Z$ is large in a global or a local database. Note here that, in a sparse data which is often used in data mining, many itemsets whose size is long have a low probability and the difference is large in many cases. This is the cause that our pruning rules are difficult to be applied to the itemsets whose size is long. Therefore, in order to solve the problem and increase the chance of our pruning, we have to weaken conditions of our pruning rules and modify a procedure in our search.

The second cause is the large number of an itemset Z such that $P(C|Z) > \zeta$ in the experimental data. In Fig. 1, in Atlanta, it seems that our pruning rules cannot reduce only 100 thousands itemsets out of one million and 920 thousands all itemsets. But there are one million and 570 thousands itemsets Z such that $P(C|Z) > \zeta$ which we find in step 1. Therefore, there are only 350 thousands itemsets which don't have to be examined in Step 1. Notice here that, in Los Angeles, the number of itemsets actually examined is less than the number of itemsets Z such that $P(C|Z) > \zeta$. This phenomenon is influenced by decomposability of DC pair described in 4.2. Then, by taking decomposability of DC pairs into account more, there is a possibility that itemsets examined can

$\rho = 3.0, \zeta = 0.4$							
region	$sup(C)$	ϵ	$\|N_{full}\|$	$\|N_{drop}\|$	$\|P(C\|Z) > \zeta\|$	$\|DC\|$	$\|DC_{NotCor}\|$
Atlanta	0.064	0.0922	1922264	1826678	1575575	112877	269
Los Angeles	0.107	0.118	1857501	1760522	1769705	30404	97
New Orleans	0.079	0.102	1120224	1071306	1027241	39158	55
San Francisco	0.100	0.114	2154595	2113443	1735822	134520	312

Fig. 1. Experimental Results

be reduced. We describe the prosperity of the above problems in Concluding Remarks.

6 Concluding Remarks

Given a transaction database $\mathcal{D}$ and its sub-database $\mathcal{D}_{\mathcal{C}}$, we proposed the notion of DC pairs. A pair of itemsets X and Y is called a DC pair if the correlation between X and Y in $\mathcal{D}_{\mathcal{C}}$ is relatively high to one in the original $\mathcal{D}$ with some degree. It should be noted that the correlation is not always high in $\mathcal{D}_{\mathcal{C}}$ even though we can observe some degree of correlation change for $\mathcal{D}$ and $\mathcal{D}_{\mathcal{C}}$. In this sense, such a pair might not be characteristic in $\mathcal{D}_{\mathcal{C}}$. Thus, DC pairs are regarded as *potential characteristics* in the sub-database. Our experimental results showed that DC pairs which are potentially significant can be actually found for "Entree Chicago Recommendation Data" under conditioning by each region. On the other hand, it is turned out that our pruning rules have to be more powerful before we apply our algorithm to a problem in a real life. Then, in order to search DC pairs efficiently, we have some prosperities as follows.

At first, we try to weaken conditions of our pruning rules and modify a procedure. In the experiment, we can know our pruning rules are difficult to work well when a difference between $P(Z)$ and $P(i)$ or $P(C \cup Z)$ and $P(C \cup i)$ $(i \in Z)$ is large. Conversely, if the difference is small, our pruning rules can work well. So, in order to increase an opportunity of our pruning, a set of itemset whose probability is almost same can be useful. In order to make use of the set of itemsets, we have to weaken conditions of pruning rules. In short, for an itemset Z, our pruning rules need to be applied to itemsets $Z' \subset Z$ not items $i \in Z$. Moreover, in order to use the weaken rules, we have to modify a search procedure. Next, we try to take advantage of decomposability of DC pairs more. In 4.2, if an itemset Z examined doesn't contain $X(i \in X \subset Z)$ such that $P(C|X) < \epsilon$, we drop an item $i \in Z$ from Z because $Z'(i \in Z' \subset Z)$ cannot be divided into two itemsets X and Y such that $P(C|X) < \epsilon$ and $P(C|Y) < \epsilon$. Notice here that we can make use of the decomposability more. Briefly speaking, a DC pair is a pair of itemsets X and Y. Therefore, if X cannot hold $P(C|X) < \epsilon$ or Y cannot hold $P(C|Y) < \epsilon$, a pair of X and Y is not a DC pair. In addition, when an itemset Z is divided into two itemsets X and Y $(Z = X \cup Y, X \cap Y = \emptyset)$, X or Y contains an item $i \in Z$ necessarily. In short, Z cannot be divided into a

DC pair if $X(i \in X \subset Z)$ cannot hold $P(C|X) < \epsilon$. In a preliminary experiment, by taking the new decomposability into account, there is a possibility that the number of itemsets examined may become no more than the half number without using the new decomposability. We are trying to tackle the above problems.

Finally, we discuss our future work. As we described in Introduction, we consider our frame work can be applied to time series data. In this paper, we pay attention to a difference of correlation observed by conditioning to the local database. Based on the notion of the DC pair, if we pay attention to a difference of correlation from time t_1 to t_2 after t_1, our algorithm can be applied to time series data easily although we have to take the information particular to time series data into account. In this problem, characteristic correlation in t_1 or t_2 can be found by using search methods of previous studies. But there may be a case that we want to know an implicit correlation that may become characteristic in t_3 after t_2 although we have to consider an interval between t_1 and t_2 seriously. We can find such a correlation by capturing a difference of correlations from t_1 to t_2. We are considering the application of the notion of the DC pair to time series data.

References

1. R. Agrawal, R. Srikant. Fast algorithms for mining association rules. In *Proc. of the Int'l Conf. on Very Large Data Bases*, pages 487-99, 1994.
2. G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proc. of the 5th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 43-52, 1999.
3. H. Alhammady and K. Ramamohanarao. Using emerging patterns and decision trees in rare-class classification. In *Proc. of the 4th IEEE Int'l Conf. on Data Mining*, pages 315-18, 2004.
4. S. D. Bay and M. J. Pazzani. Detecting group differences: mining contrast sets. *Data Mining and Knowledge Discovery*, v 5, n 3, pages 213-46, 2001.
5. G. I. Webb, S. Butler and D. Newlands. On detecting differences between groups. In *Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 256-65, 2003.
6. S. Brin, R. Motwani and C. Silverstein. Beyond market baskets: generalizing association rules to correlations. In *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, v 26, n 2, pages 265-76, 1997.
7. S. Brin, R. Motwani, J. D. Ullman and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, v 26, n 2, pages 255-64, 1997.
8. C. C. Aggarwal, P. S. Yu. A new framework for itemset generation. In *Proc. of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1998)*, pages 18-24, 1998.
9. S. Morishita and J. Sese. Traversing itemset lattices with statistical metric pruning. In *Proc. of the ACM SIGACT-SIGMOD-SIGART Symposium on Database Systems (PODS)*. pages 226-36, 2000.
10. Y. Ohsawa and Y. Nara. Understanding internet users on double helical model of chance-discovery process. In *Proc. of the IEEE Int'l Symposium on Intelligent Control*, pages 844-9, 2002.