Mining Quantitative Frequent Itemsets Using Adaptive Density-based Subspace Clustering

Takashi Washio, Yuki Mitsunaga and Hiroshi Motoda
The Institute for Scientific and Industrial Research, Osaka University
8-1, Mihogaoka, Ibaraki City, Osaka, 567-0047, Japan
washio@ar.sanken.osaka-u.ac.jp

Abstract

A novel approach to subspace clustering is proposed to exhaustively and efficiently mine quantitative frequent itemsets (QFIs) from massive transaction data for quantitative association rule mining. The numeric part of a QFI is an axis-parallel and hyper-rectangular cluster of transactions in an attribute subspace formed by numeric items. For the computational tractability, our approach introduces adaptive density-based and Apriori-like subspace clustering. Its outstanding performance is demonstrated through the comparison with the past subspace clustering approaches and the application to practical and massive data.

1. Introduction

An important extension of association rule mining is to mine "*Quantitative Association Rules* (QARs)" covering the relations among both numeric and categorical items in transaction data [12]. The rules have the form "$\{< p_1 : q_1 >, ..., < p_i : q_i >\} \Rightarrow \{< p_{i+1} : q_{i+1} >, ..., < p_m : q_m >\}$" where $< p : q >$ is an item, p an attribute and q its value. An example is "$\{< Age : [30, 39] >, < NumCars : [2, 2] >\} \Rightarrow \{< Married : Yes >\}$" stating "A person who is in his/her thirties and has two cars is married." A "*numeric item*" has a numeric interval value whereas a "*categorical item*" has a categorical value. Given a transaction t, if every item in the body and the head of a QAR is supported by the items in t, then this rule holds in t. Here, a numeric item $< p : q >$ in the QAR is supported by a numeric item $< p_t : q_t >$ in t if $p_t = p$ and $q_t \subseteq q$ where $\subseteq$ states that the range of q_t is within the range of q. Hence, "$t_1 = \{< Age : [35, 37] >, < Married : Yes >, < NumCars : [2, 2] >, < Child : [3, 3] >\}$" supports the aforementioned rule, whereas "$t_2 = \{< Age : [29, 31] >, < Married : Yes >, < NumCars : [2, 2] >, < Child : [3, 3] >\}$" does not, because $< Age : [29, 31] >$ is not within $< Age : [30, 39] >$. Given a transaction data set D, the union of the body and the head of a QAR is a "*frequent itemset*" if it is supported by D more frequently than a "*minimum support* ($minsup$)" threshold. A frequent itemset including numeric items is called as a "*Quantitative Frequent Itemset*," QFI in short. The numeric part of a QFI corresponds to an axis-parallel and hyper-rectangular region in a subspace of the entire attribute space of D.

A pioneering work to mine QARs was made by Srikant and Agrawal [12], where each numeric attribute is equi-depth partitioned, and the adjacent intervals are merged to a maximum support limit in preprocessing. The conventional levelwise algorithm to mine frequent itemsets is subsequently applied. Wang et al. proposed a more efficient approach to merge adjacent intervals under an interestingness measure [14]. Its time complexity $O(N \log N)$, with $N = |D|$, is feasible for practical applications. However, they often fail to discover the appropriate discretization, because each attribute is independently discretized from the others under greedy strategies. To overcome this difficulty, optimized approaches have been explored for some criteria such as to mine QARs having maximum confidence under a given minimum support [5, 10]. However, this optimization is known to be *NP-complete* and practically intractable [15]. The candidate upper and lower bounds of intervals needed for the discretization exponentially increase, if the number of numeric attributes in D increases.

To mine reasonably optimal QARs within tractable computational complexity, a natural extension of Basket Analysis is considered by introducing the subspace clustering which searches axis-parallel and hyper-rectangular clusters of the transactions where each cluster corresponds to the numeric part of a QFI. It efficiently derives QFIs, because a dense region of many transactions in an attribute subspace is supported by all transactions within the region. Moreover, the density measure has the (anti-)monotonicity property that the transactions in a dense cluster in an attribute space are always included in some dense clusters in its subspaces. An efficient algorithm to mine QFIs can be designed based on this property together with the (anti-)monotonicity property of the support of categorical itemsets.

Many studies addressed the density-based subspace clustering in a high dimensional and numeric attribute-value table. In CLIQUE, the original numeric attribute space is first discretized by an axis-parallel grid, and a maximal set of connected dense blocks in the grid are searched as a cluster by levelwisely merging the blocks [1]. ENCLUS [3], MAFIA [6] and SCHISM [11] are the successive extensions that respectively introduce entropy based interestingness, variable width of the grid and variable thresholds of density. DOC uses moving hypercube windows to measure the density of instances in attribute subspaces [9]. CLTree mines axis-parallel and hyper-rectangular subspace clusters under a hierarchical and greedy search strategy by using entropy-based density measure [8]. Though their computational complexities are low (around $O(N) \sim O(N \log N)$), they miss some clusters due to inadequate orientations, shapes and sizes of their grids/windows and due to the incompleteness of their search strategies. The recently developed SUBCLU searches subspace clusters under a rigid density measure proposed by DBSCAN [7, 4], where a dense region called a "*density-connected set*" is that for each object in the region the neighborhood of a given radius ϵ has to contain at least a minimum number of $MinPts$ objects. This approach exhaustively searches dense clusters in every attribute subspace by an Apriori-like levelwise algorithm with the (anti-)monotonicity property of the dense clusters, and does not miss any clusters except the cases having distorted distributions. However, because every pairwise distance between instances must be computed, the computational complexity is $O(N^2)$. A crucial limitation of these approaches other than the completeness and the complexity is that they are only dedicated to numeric instances.

In this paper, we focus on the QFI mining as the basis to mine QARs, and propose a novel approach called "*QFIMiner*" having the following features:

1. The approach exhaustively mines all dense clusters supported by more than $minsup$ transactions in all subspaces formed by both numeric and categorical attributes of a given transaction data.
2. The clusters to be mined have axis-parallel and hyprectangular shapes in the numeric attribute subspaces.
3. Interval values of numeric items are allowed in the transactions for mining.
4. The approach is virtually $O(N \log N)$ and tractable.

The second section outlines QFIMiner we propose. The third section describes its details. Its outstanding performance is demonstrated in the fourth section.

2. Outline

QFIMiner searches QFIs from a data set D of transactions consisting of numeric and categorical items. The numeric part of a transaction represents an axis-parallel and hyper-rectangular region in a subspace S of the entire attribute space of D. QFIMiner assumes that dense clusters of the transactions exist with scattered outliers, i.e., background noise, in the subspace. Figure 1 depicts this example where every numeric item takes a point interval (unique) value in each transaction. Two dense clusters are in $S = \{p_1, p_2\}$ together with much background noise.

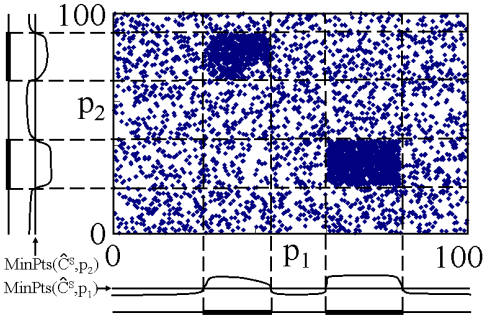

Figure 1. Clusters and their projections.

QFIMiner does not use any preset grids and windows for density evaluation, but uses a definition of density similar to DBSCAN. This approach significantly reduces the possibility to miss clusters under an appropriate density threshold. QFIMiner uses a levelwise algorithm where it starts from the clusters in one dimensional subspaces, and joins $(k-1)$ dimensional clusters into a candidate cluster $\hat{C}^S$ in k dimensional subspace S. While this is similar to SUBCLU, QFIMiner can derive clusters on both numeric and categorical items by embedding the levelwise subspace clustering into the standard Apriori algorithm. The clusters supported more than a minimum support ($minsup$) in numeric and categorical attribute subspaces are exhaustively mined.

To avoid $O(N^2)$ computational complexity, QFIMiner does not compute the pairwise distances among transactions. Instead, it projects transactions in a candidate dense cluster $\hat{C}^S$ onto each attribute axis of the subspace S. Fig. 1 shows a case that $\hat{C}^S$ is a $[0, 100] \times [0, 100]$ region. All maximal density-connected sets are searched in the transactions projected onto every axis, where a density-connected set on an attribute axis is such that for each transaction in the set the $\pm\Delta$ neighborhood on the axis has to contain at least a minimum number of $MinPts$ transactions, and a maximal density-connected set is not contained in any other density-connected set. An intersection of the maximal density-connected sets on all axes in the subspace becomes a new $\hat{C}^S$ due to the (anti-)monotonicity of the density. In Fig. 1, the four intersections are new $\hat{C}^S$. These projection and searching maximal density-connected sets are iterated until each $\hat{C}^S$ converges to a dense cluster C^S where its projection to every axis in S is dense. The two intersections containing the dense clusters in Fig. 1 are retained under this iteration and the rest pruned. Because the density on every axis is evaluated within a scan of sorted transactions, the complexity of this algorithm is expected to be $O(N \log N)$.

In the search of maximal density-connected sets on an axis, if $MinPts$ is lower than the background noise level, the projection of dense clusters may be buried in the background. If it is too high, the projection of dense clusters may be missed. Accordingly, $MinPts$ is adapted to the number of $MinPts(\hat{C}^S, p)$ transactions projected to the $\pm\Delta_p$ neighborhood on an axis p from each $\hat{C}^S$ assuming that $\hat{C}^S$ has the average density of the subspace S. $MinPts(\hat{C}^S, p)$ is always between the densities of the dense cluster and the background. In Fig. 1, $MinPts(\hat{C}^S, p)$ efficiently extracts the maximal density-connected sets reflecting the dense clusters. This adaptive density threshold further accelerates QFIMiner, because $MinPts(\hat{C}^S, p)$ is higher for a lower subspace dimension, and prunes more maximal density-connected sets below the noise level. The projected density and its adaptive thresholds are the keys to reduce the computation of QFIMiner while maintaining its output quality.

3. Methods and Algorithms

3.1. Levelwise Subspace Clustering

First we focus on the subspace clustering of transactions consisting of numeric items only. The density threshold of $MinPts$ is left constant without loss of generality, and its adaptation to $MinPts(\hat{C}^S, p)$ is explained later.

Definition 1 (Neighborhood) *Let p be a numeric attribute, and let t and t' be two transactions sharing an attribute p with interval values q and q' respectively. Let their distance on the axis of p, $Dist_p(q, q')$, be the minimum distance between a point in q and a point in q', i.e., $\min_{v \in q, v' \in q'} |v - v'|$. Furthermore, let Δ_p be a "permissible range" on the attribute p. The "Δ_p-neighborhood" $N_{\Delta_p}(t)$ on p is defined by*

$$N_{\Delta_p}(t) = \{t' \in D | Dist_p(q, q') \leq \Delta_p\}.$$

If intervals q and q' overlap, then $Dist_p(q, q') = 0$, otherwise $Dist_p(q, q')$ is the distance between their boundaries facing each other.

Definition 2 (Core transaction) *A transaction $t \in D$ is called a "core transaction" on p if its Δ_p-neighborhood $N_{\Delta_p}(t)$ contains at least $MinPts$ transactions, i.e.,*

$$|N_{\Delta_p}(t)| \geq MinPts.$$

Definition 3 (Direct Density-Reachability) *An transaction $t \in D$ is "directly density-reachable" from another transaction $t' \in D$ on p, if t' is a core transaction on p, and t is an element of $N_{\Delta_p}(t')$.*

Definition 4 (Density-Reachability) *A transaction $t \in D$ is "density-reachable" from another transaction $t' \in D$ on p, if there is a chain of transactions $t_1, t_2, \cdots, t_{n-1}, t_n$, $t_1 = t$ and $t_n = t'$ in D where t_{i+1} is directly density reachable from t_i on p.*

Definition 5 (Density-Connectivity) *A transaction $t \in D$ is "density-connected" to an transaction $t' \in D$ on p if there is a transaction t'' such that both t and t' are density-reachable from t'' on p.*

Table 1. An example of transaction data set D.

$t_1 = \{< Age : [20, 23] >, < Child : [2, 3] >, < NumCars : [2, 2] >\}$
$t_2 = \{< Age : [30, 30] >, < Child : [4, 5] >, < NumCars : [1, 1] >,$ $< Savings : [10K, 10K] >\}$
$t_3 = \{< Age : [30, 30] >, < Child : [2, 2] >, < NumCars : [5, 5] >,$ $< Savings : [11K, 11K] >\}$
$t_4 = \{< Age : [30, 35] >, < Child : [5, 5] >, < NumCars : [1, 1] >\}$
$t_5 = \{< Age : [35, 37] >, < Child : [2, 2] >, < NumCars : [2, 2] >,$ $< Savings : [5K, 5K] >\}$
$t_6 = \{< Age : [36, 39] >, < Child : [2, 2] >, < NumCars : [2, 3] >\}$

Definition 6 (Density-Connected Set) *A non-empty subset $C \subseteq D$ is a "density-connected set" on p if all transactions in C are density-connected on p.*

Definition 7 (Dense Cluster) *A "dense cluster" $C^S \subseteq D$ in a subspace formed by a set of numeric attributes S is defined as a non-empty set of transactions density-connected on every $p \in S$ which is maximal w.r.t. density-reachability on every $p \in S$ in D.*

Definition 8 (Quantitative Frequent Itemset) *Let $C^S \subseteq D$ be a dense cluster in a subspace S and $a(C^S) = \{< p : q > | p \in S, q = [\min_p(C^S), \max_p(C^S)]\}$ an itemset where $\min_p(C^S)$ and $\max_p(C^S)$ are the minimum and the maximum interval boundaries of transactions in C^S on p. If $|C^S| \geq minsup$, $a(C^S)$ is a "quantitative frequent itemset (QFI)." When the dimension of S is k, it is called a k-QFI.*

A QFI is a dense, axis-parallel and monotone hyperrectangular region having a maximal volume in the subspace. Similarly to the dense clusters of SUBCLU, the following (anti-)monotonicity property of QFIs holds.

Lemma 1 (Monotonicity) *$\forall T \subseteq S$, if $a(C^S)$ is a QFI in S, then a QFI $a(C^T)$ supported by $a(C^S)$, i.e., $a(C^S) \subseteq a(C^T)$, exists in T.*

Proof. Because all transactions in C^S are density-connected on every $p \in S$, they are density-connected on every $p \in T$, and hence $C^S \subseteq C^T$. Accordingly, for every $p \in T$, $[\min_p(C^S), \max_p(C^S)] \subseteq [\min_p(C^T), \max_p(C^T)]$, and thus $a(C^T)$ is supported by $a(C^S)$. ∎

Accordingly, a levelwise bottom up approach is applicable to search all QFIs. We exemplify its operation by using the dataset in Table 1 under $\Delta_{Age} = 5$, $\Delta_{Child} = 1$, $\Delta_{NumCars} = 1$, $\Delta_{Savings} = 1K$, $MinPts = 1$ and $minsup = 2$. First, the items in each t_i are lexicographically ordered by the attribute names. This has been already done in this table. Subsequently, 1-QFIs are searched, where the transactions are maximally density-connected on an attribute. For Age, a 1-QFI, $\{< Age : [30, 39] >\}$, exists since the items densely range from 30 to 39 under $\Delta_{Age} = 5$, and its support 5 is more than $minsup$. This 1-QFI with its "*transaction id list* (TID-List)" is indicated in Table 2. Each attribute has an 1-QFI in this example.

In the next step, the levelwise search for k-QFIs ($k > 1$) starts, where index lists named $TID-List$ are used to point transactions in D similarly to AprioriTid algorithm [2]. Assuming that all $(k-1)$-QFIs are known, the following "*Candidate-Generation*" derives all candidate k-QFIs.

Table 2. Levelwise subspace clustering of D.

1-QFIs
$(\{< Age : [30,39] >\}, \{t_2, t_3, t_4, t_5, t_6\})$,
$(\{< Child : [2,5] >\}, \{t_1, t_2, t_3, t_4, t_5, t_6\})$,
$(\{< NumCars : [1,3] >\}, \{t_1, t_2, t_4, t_5, t_6\})$,
$(\{< Savings : [10K, 11K] >\}, \{t_2, t_3\})$
2-QFIs
$(\{< Age : [30,39] >, < Child : [2,2] >\}, \{t_3, t_5, t_6\})$
$(\{< Age : [30,35] >, < Child : [4,5] >\}, \{t_2, t_4\})$
$(\{< Age : [30,39] >, < NumCars : [1,3] >\}, \{t_2, t_4, t_5, t_6\})$
$(\{< Age : [30,30] >, < Savings : [10K, 11K] >\}, \{t_2, t_3\})$
$(\{< Child : [2,5] >, < NumCars : [1,3] >\}, \{t_1, t_2, t_4, t_5, t_6\})$
3-QFIs
$(\{< Age : [35,39] >, < Child : [2,2] >,$ $< NumCars : [2,3] >\}, \{t_5, t_6\})$
$(\{< Age : [30,35] >, < Child : [4,5] >,$ $< NumCars : [1,1] >\}, \{t_2, t_4\})$

Definition 9 (Candidate-Generation)
Join Phase: *For two $(k-1)$-QFIs sharing $k-2$ attributes,*

$$((k-1)-QFI = \{< p_1 : q_1 >, < p_2 : q_2 >, ..., < p_{k-2} : q_{k-2} >, < p_{k-1} : q_{k-1} >\}, TID-List),$$
$$((k-1)-QFI' = \{< p_1 : q'_1 >, < p_2 : q'_2 >, ..., < p_{k-2} : q'_{k-2} >, < p_k : q'_k >\}, TID-List'),$$

their join is derived as follows:

$$(candidate-k-QFI = \{< p_1 : q_1^c >, < p_2 : q_2^c >, ..., < p_{k-2} : q_{k-2}^c >, < p_{k-1} : q_{k-1} >, < p_k : q_k >\}, TID-List^c).$$

where $q_i^c = q_i \cap q'_i$ is the intersection of the two intervals and $TID-List^c = TID-List \cap TID-List'$. If some $q_i^c = \phi$, i.e., no intersection exists, or $TID-List^c = \phi$, the given two $(k-1)$-QFIs are not joined.

Prune Phase: *For all $(k-1)$-subsets s of this candidate-k-QFI, if the following $(k-1)$-QFI exists:*

$$\forall < p_i : q_i^c > \in s, \exists < p_i : q_i > \in (k-1)-QFI, \quad q_i^c \cap q_i \neq \phi, \tag{1}$$

the candidate-k-QFI is retained otherwise pruned. $TID-List^c$ is a candidate dense cluster $\hat{C}^S$ where $|S| = k$.

This prune phase is based on Lemma 1. As far as q_i^c intersects with q_i in Eq. (1), the possibility that s and $(k-1)-QFI$ shares transactions more than $minsup$ is not negligible. Thus the candidate k-QFI is retained under this condition. In Table 2, a candidate-2-QFI, $\{< Age : [30,39] >, < Child : [2,5] >\}$ with $TID-List^c = \{t_2, t_3, t_4, t_5, t_6\}$ is derived from two 1-QFIs, $\{< Age : [30,39] >\}$ and $\{< Child : [2,5] >\}$. This passes the prune phase.

From Lemma 1, a dense cluster C^S ($|S| = k$) corresponding to $k-QFI$ follows $C^S \subseteq C^T$ and $C^S \subseteq C^{T'}$ where C^T and $C^{T'}$ ($|T| = |T'| = k-1$) are dense clusters corresponding to $(k-1)-QFI$ and $(k-1)-QFI'$ respectively. Because $\hat{C}^S$ is an intersection of C^T and $C^{T'}$, $C^S \subseteq \hat{C}^S$. Accordingly, the dense cluster C^S and its k-QFI, if they exist, can be derived by assessing the density of transactions in $\hat{C}^S$. q_i and $TID-List$ of the k-QFI can be computed through Definitions 7 and 8. The algorithm of "*QFI-Count*" shown in Fig. 2 performs these computations. First, a candidate-k-QFI with its $TID-List^c = \hat{C}^S$

QFI-Count($candidate-k-QFI, TID-List^c$);
/* Notions of input arguments follow Definition 9.*/
(1) $k-QFIS = \phi, TIDLS = \phi$;
(2) If $|TID-List^c| < minsup$ return $k-QFIS$;
(3) $S = \{p | < p : q > \in candidate-k-QFI, p$ is numeric.$\}$;
(4) $TIDLS.temp = \{TID-List^c\}$;
(5) while $TIDLS \neq TIDLS.temp$ do begin
(6) $\quad TIDLS = TIDLS.temp$;
(7) $\quad$ forall $p \in S$ do begin
(8) $\quad\quad TIDLS.temp = MDCS(TIDLS.temp, p)$;
(9) $\quad$ end
(10) end
(11) forall $TID-List \in TIDLS$ do begin
(12) $\quad k-QFIS = k-QFIS + (QFI(S, TID-List), TID-List)$;
(13) end
(14) return $k-QFIS$;

Figure 2. Algorithm of QFI-Count.

generated in Candidate-Generation is given. If $|\hat{C}^S|$ is less than $minsup$, $k-QFIS = \phi$ is returned as the output at step (2). In the inside loop from step (7) to (9), a maximal density-connected set C is searched on p within $\hat{C}^S$ at first in a function $MDCS$ along with Definition 6 under given Δ_p and $MinPts$. Multiple C can be found in $MDCS$ when multiple dense clusters are included in $\hat{C}^S$. $MDCS$ repeats to update C on p from every maximal density-connected set derived and kept in $TIDLS.temp$ at the previous loop path. C having the size less than $minsup$ are discarded in $MDCS$. This update continues in the outer loop from step (5) to (10), until each C converges to dense clusters C^S where each C^S is derived independently of the convergence process due to the (anti-)monotonicity property. Each dense cluster is represented by a $TID-List$ in $TIDLS$. In the loop from step (11) to (13), each QFI corresponding to a $TID-List$ is computed by Definition 8 in a function QFI, and they are returned as the output. In the example, the candidate-2-QFI, $\{< Age : [30,39] >, < Child : [2,5] >\}$ with $TID-List^c = \{t_2, t_3, t_4, t_5, t_6\}$ is given to this QFI-Count. In the inside loop, $MDCS$ derives $TIDLS.temp = \{\{t_2, t_3, t_4, t_5, t_6\}\}$ as C on Age under $\Delta_{Age} = 5$. Next under this $TIDLS.temp$, it derives $TIDLS.temp = \{\{t_3, t_5, t_6\}, \{t_2, t_4\}\}$ on $Child$ under $\Delta_{Child} = 1$. Further applications of $MDCS$ do not change $TIDLS.temp$. Since the sizes of candidates are more or equal to $minsup = 2$, two 2-QFIs, $(\{< Age : [30,39] >, < Child : [2,2] >\}, \{t_3, t_5, t_6\})$ and $(\{< Age : [30,35] >, < Child : [4,5] >\}, \{t_2, t_4\})$, are derived.

3.2. Deriving Quantitative Frequent Itemsets

Candidate-Generation is extended to derive QFIs consisting of numeric and categorical items. The values of categorical items in the joined itemset are given in the same way as in the standard AprioriTid algorithm.

(1) For each numeric attribute, create an index list sorted with the ascending order of D. Sort items in each $t \in D$ lexicographically.
(2) $L_1 = \{(1 - QFI, TID - List)\}$;
(3) for (k=2; $L_{k-1} \neq \phi$; $k++$) do begin
(4) $C_k = \{(candidate - k - QFI, TID - List^c)\} = Extended - Candidate - Generation(L_{k-1})$;
(5) forall $(candidate - k - QFI, TID - List^c) \in C_k$ do begin
(6) $L_k = L_k \cup QFI - Count(candidate - k - QFI, TID - List^c)$
(7) end
(8) end
(9) Answer $L = \bigcup_k L_k$;

Figure 3. Entire algorithm.

Definition 10 (Extended-Candidate-Generation)
Join Phase*: For two $(k-1)$-QFIs sharing $k-2$ attributes,*
$((k-1) - QFI = \{< p_1 : q_1 >, < p_2 : q_2 >, ..., < p_{k-2} : q_{k-2} >, < p_{k-1} : q_{k-1} >\}, TID - List)$,
$((k-1) - QFI' = \{< p_1 : q'_1 >, < p_2 : q'_2 >, ..., < p_{k-2} : q'_{k-2} >, < p_k : q'_k >\}, TID - List')$,
their join is derived as follows:
$(candidate - k - QFI = \{< p_1 : q^c_1 >, < p_2 : q^c_2 >, ..., < p_{k-2} : q^c_{k-2} >, < p_{k-1} : q_{k-1} >, < p_k : q_k >\}, TID - List^c)$.
where $q^c_i = q_i \cap q'_i$ for a numeric item, $q^c_i = q_i = q'_i$ for a categorical item and $TID - List^c = TID - List \cap TID - List'$. If $q^c_i = \phi$ for some numeric item, $q_i \neq q'_i$ for some categorical item or $TID - List^c = \phi$, the given two $(k-1)$-QFIs are not joined.
Prune Phase*: For all $(k-1)$-subsets s of this candidate-k-QFI, if the following $(k-1)$-QFI exists;*

$$\forall < p_i : q^c_i > \in s, \exists < p_i : q_i > \in (k-1) - QFI,$$
$q^c_i \cap q_i \neq \phi$ for a numeric item,
and $q^c_i = q_i$ for a categorical item,

the candidate-k-QFI is retained otherwise pruned. $TID - List^c$ is a candidate dense cluster $\hat{C}^S$ where $|S| = k$.

The algorithm QFI-Count shown in Fig. 2 must be also altered. When the candidate k-QFI consists of categorical items only, the loop from step (5) to (10) is skipped, and $TIDLS = TIDLS.temp$ is applied. The function QFI at step (12) is also altered. For a categorical attribute p_i not covered by Definition 8, its value is set to be $q^c_i = q_i = q'_i$.

The entire algorithm to derive QFIs from D is indicated in Fig. 3. Required parameters are Δ_p for all numeric attributes, $MinPts$ and $minsup$. First, some index lists are created for the efficient processing in Extended-Candidate-Generation and QFI-Count. Subsequently, all QFIs are computed in L by the adaptation of the Apriori-Tid Algorithm. In the implementation, the inversed indexing $(t_i, \{candidate - k - QFI\})$ from each t_i to its containing candidate-k-QFIs is used instead of $(candidate - k - QFI, TID - List^c)$ similarly to the standard AprioriTid. The most expensive tasks are the sort at the first step which is $O(N \log N)$ and the derivation of the dense clusters in QFI-Count. The derivation of the maximal density-connected sets on every numeric attribute axis p is easily made in one scan of $TIDLS.temp$ in $MDCS$ in Fig. 2 by using the index list made at the first step in Fig. 3, and hence it is $O(N)$ at maximum. The iteration of the outer loop from step (5) to (10) in QFI-Count of Fig. 2 strongly depends on the distribution of transactions in attribute subspaces. In the worst case where a transaction is removed in each loop path, the total complexity of the loop is $O(N^2)$. However, in the most likely case, only a portion $0 < r < 1$ of the transactions in the average are retained in each loop path. The loop finishes when $r^m N$ becomes less than $minsup$ where m is the number of loop paths. From $minsup \simeq r^m N$, m is around $O(\log N)$. Accordingly, the expected time complexity of the entire algorithm is $O(N \log N)$.

3.3. Adaptive Density Threshold

The optimality of a density threshold to discriminate dense clusters from background noise is not generally defined, since the measures to estimate density and number of outliers are mostly subjective matters. Rather than the optimality, we introduce a robust density threshold under the following consideration. Unless the transactions are uniformly distributed in the space, dense clusters possibly exist in a region having relatively higher density than the other region of the space. This implies the following proposition.

Proposition 1 (Average Density) *Most of dense clusters of the space are located within the region whose density is more than the average density of transactions in the space, whereas most of outliers in the space are located within the region whose density is less than the average.*

Let D^S be a set of transactions containing all attributes of a subspace S in D, *i.e.* a set of transactions lying in S. We consider the average density $\bar{d}^S$ of transactions of D^S over the region of S where are these transactions. Upon the above proposition, most of dense clusters are located within the region having density more than $\bar{d}^S$, while most of outliers are out of this region. On the other hand, a dense cluster C^S exists in its candidate dense cluster $\hat{C}^S$ as explained earlier. This implies that the region of C^S denser than $\bar{d}^S$ exists in the region of $\hat{C}^S$. When $MDCS$ in Fig. 2 searches the maximal density-connected sets on an axis p, the transactions in $\hat{C}^S$ are projected onto p, and the Δ_p-neighborhood $N_{\Delta_p}(t)$ in Definition 1 is computed for every projected t. If a dense cluster C^S exists in $\hat{C}^S$, $N_{\Delta_p}(t)$ of many t in $\hat{C}^S$ must be larger than the case that the density of $\hat{C}^S$ is equal to $\bar{d}^S$. Accordingly, we introduce the adaptive density threshold $MinPts(\hat{C}^S, p)$ which is the expected value of $N_{\Delta_p}(t)$ under $\hat{C}^S$ having its density $\bar{d}^S$.

Lemma 2 (Adaptive Density Threshold) *The adaptive density threshold is given as*

$$MinPts(\hat{C}^S, p) = |D^S| r_p,$$
$$\text{where } r_p = \frac{2\Delta_p}{R_p} \prod_{p' \in S, p' \neq p} \frac{C_{p'}}{R_{p'}},$$

C_p the width of $\hat{C}^S$ on p and R_p the range of the transactions in D^S located on p.

Proof. The volume of the region where the transactions are located in the subspace S is $\prod_{p' \in S} R_{p'}$, and thus its average density $\bar{d}^S$ is $|D^S| / \prod_{p' \in S} R_{p'}$. Since $\hat{C}^S$ is hyper-rectangular, its volume in S is $\prod_{p' \in S} C_{p'}$. Accordingly, the average number of transactions in $\hat{C}^S$ having $\bar{d}^S$ is $|D^S| \prod_{p' \in S} C_{p'} / R_{p'}$. Because these transactions range within C_p on p, the ratio $2\Delta_p / C_p$ of the transactions are captured in $\pm\Delta_p$ on p. Hence, the expected $N_{\Delta_p}(t)$ is;

$$MinPts(\hat{C}^S, p) = \frac{2\Delta_p}{C_p} |D^S| \prod_{p' \in S} \frac{C_{p'}}{R_{p'}} = |D^S| \frac{2\Delta_p}{R_p} \prod_{p' \in S, p' \neq p} \frac{C_{p'}}{R_{p'}}. \blacksquare$$

This $MinPts(\hat{C}^S, p)$ is applied to derive every maximal density-connected set in the function $MDCS$ in Fig. 2.

The input parameters of QFIMiner are Δ_p and $minsup$ where Δ_p is usually given by a unique relative width Δ over the range R_p of every p. In concert with $minsup$, $MinPts(\hat{C}^S, p)$ efficiently extracts even small dense clusters in much background noise. Suppose a two dimensional data of $|D| = 4100$, where 4000 transactions are background noise uniformly distributed in a region $[0, 100] \times [0, 100]$ of a space $\{p_1, p_2\}$, and the rest 100 are further added to form a dense cluster in $[40, 60] \times [40, 60]$. Under $\Delta = 2\%$ ($\Delta_{p_i} = \Delta R_{p_i}$), $MinPts(\hat{C}^{\{p_i\}}, p_i)$ is $2\Delta_{p_i} / R_{p_i} |D^{\{p_i\}}|$ for $i = 1, 2$ when $k = |S| = 1$, and is 164 where $R_{p_i} = 100$ and $|D^{\{p_i\}}| = |D| = 4100$. On the other hand, the number of noisy transactions in $[40, 60]$ is 800, and the additional 100 transactions are in $[40, 60]$. Thus, the expected $N_{\Delta_p}(t)$ in $[40, 60]$ is $2\Delta_{p_i} / (60 - 40) \times 900 = 180$ whereas it is 160 in the other region. Though the differences among $MinPts(\hat{C}^S, p)$ and these $N_{\Delta_p}(t)$ are not significant, most of the transactions in the dense cluster become core transactions, and form a maximal density-connected set in $[40, 60]$ on each p_i, whereas the majority in the other region do not, and form some noisy maximal density-connected sets mostly pruned by $minsup$.

4. Experimental Evaluation

The performance of QFIMiner has been evaluated in terms of efficiency, output quality, scalability and practical usability by using personal computers with a 2.7GHz Pentium 4 CPU and 2GB RAM throughout this section.

4.1. Comparison with Other Approaches

QFIMiner was compared with SUBCLU [7] and QAR mining of Srikant and Agrawal [12]. SUBCLU takes the parameters ϵ (Δ in our expression) and $MinPts$ for the density computation. Its performance is known to be superior to CLIQUE [7]. QAR mining applies the ordinary Apriori algorithm to mine QFIs after preprocessing numeric items. It takes the parameters $minsup$ and maximum support ($maxsup$) [1]. We obtained SUBCLU's code from its authors [7]. The program to derive QFIs by QAR mining has been rebuilt based on [12]. We also obtained data sets for the experiments from the authors of SUBCLU. They are artificially generated to locate some clusters having different densities in different dimensional subspaces. The size of each cluster is around 20% of the range R_p on each axis in the subspace. Substantial background noise has been uniformly added to the entire attribute space. The similar data sets were used to evaluate SUBCLU and CLIQUE in their paper [7]. The appropriate values of $MinPts$ and $maxsup$ which QFIMiner does not use have been predetermined by our insights on these data sets. $\Delta = 2\%$, $minsup = 2\%$, $MinPts = 8$ and $maxsup = 20\%$ were used for the three approaches throughout this subsection.

First, we assessed the computational efficiency against the size of the data set, the dimensionality of the data set and the maximum dimensionality of the hidden subspace clusters as depicted in Figs. 4, 5 and 6. For Fig. 4, data sets having 20 dimensions and containing five subspace clusters of $4 \sim 7$ dimensions were used. The increase of the computation time of QFIMiner against the data size N is less than SUBCLU and similar to QAR mining. This is consistent with $O(N \log N)$ of QFIMiner, $O(N^2)$ of SUBCLU and $O(N)$ of QAR Mining. For Fig. 5, data sets having 3967 transactions and containing a 7 dimensional subspace cluster were used. The increase of QFIMiner is larger than the others, while we observed its significant decrease when we slightly increased $MinPts(\hat{C}^S, p_i)$ above the value of Lemma 2. This implies that the enumeration of noisy candidate dense clusters in lower dimensional subspaces increases under our adaptive density threshold along with the increase of the combinations of attributes. For Fig. 6, data sets having 4000 transactions and 15 dimensions were used. The computation times gently increase in all approaches. This indicates that the increase of computational amount in high dimensional search is relatively small, and the efficient and accurate pruning of candidates in the low dimensional subspaces is a key for the speed. Every figure shows that QFIMiner is faster than SUBCLU and slower than QAR mining in the order of 2 and 1, respectively.

Table 3 shows the accuracy of the clustering in terms of the number of clusters correctly mined by the three approaches. Six data sets having different subspace clusters were used for the evaluation. QFIMiner could successfully mine the clusters for almost all data sets, while the other

[1] A parameter called partial completeness $K = 0.1$ is also used, but it is not very influencial to mine QFIs.

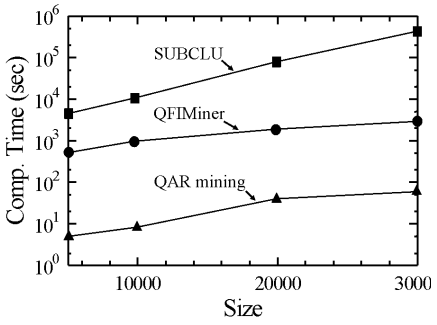

Figure 4. Time vs. data size.

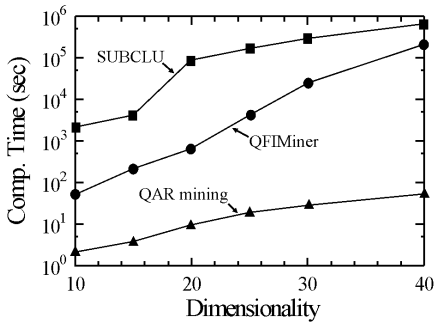

Figure 5. Time vs. dimensionality.

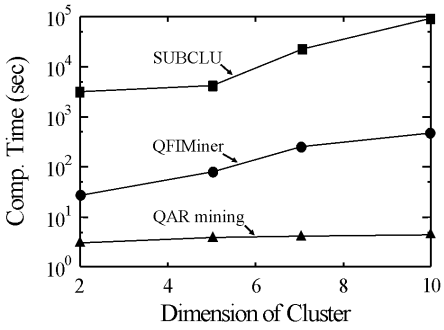

Figure 6. Time vs. dim. of clusters.

Table 3. Comparative evaluation.

N	dim. of cluster (# of trans-sactions)	# of clusters	QFIMiner	SUBCLU	QAR mining
4324	3(711) 5(256) 7(92)	3	3	2	1
4057	5(256) 5(256) 5(256)	3	3	3	2
4469	5(256) 5(256) 5(256) 5(256) 5(256)	5	4	2	1
4045	2(3673)	1	1	1	1
3967	7(1024)	1	1	1	1
3986	10(49)	1	0	0	0

The dimensionality of all data sets is 15.

two failed in many data sets. All approaches failed in the last data set, because the density of the cluster is very low. The reason why the performance of QFIMiner is better than SUBCLU is the evaluation of the density by the projection to every axis. The projection reduces the statistical errors on the density by collecting the transactions of $\hat{C}^S$ onto an axis, and this increases the accuracy of the search. In contrast, the error of the density directly evaluated in a hyper-ϵ-neighborhood increases due to the sparse distribution of the instances in higher dimensional subspaces. In summary, the performance of QFIMiner is superior to the other approaches in accuracy and efficiency.

4.2. Scalability

The scalability of QFIMiner w.r.t. computation time and memory consumption has been evaluated through the application to large artificial data consisting of both numeric and categorical items. A set of transaction data is generated where each transaction is made by randomly selecting a QFI from a set of predefined seed QFIs and further adding extra random items. The values of numeric items in each transaction are distorted by introducing Gaussian noise having $\pm 5\%$ amplitude to give the 10% diameter of subspace clusters and some background noise. The number of generated item types is 1000 where half of them are numeric. The number of seed QFIs is 10, and the average size of a transaction is 24. $\Delta = 5\%$ and $minsup = 5\%$ were used in this analysis. The dependency of the computation time and the memory consumption on N is shown in Fig. 7. As known by the reference line, the computation time is almost $O(N \log N)$ up to 1 million transactions. The memory consumption is almost $O(N)$. This is reasonable, since the memory is mainly consumed for $TID-List$. These results indicate the tractability of QFIMiner for massive data.

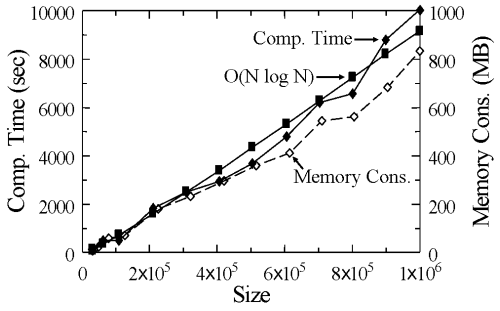

Figure 7. Time and Memory vs. data size.

4.3. Mining Practical Data

We demonstrate a data mining example to see if QFIMiner can mine comprehensive relations to characterize objective data. The data is the adult database in UCI ML Repository [13]. We analysed its test data containing 100,000 instances of 40 attributes where 6180 instances are persons having more than 50K USD annual income, and the rest are persons having less than 50K USD. We splitted the data into two groups according to the 50K USD threshold for a comparative study. $\Delta = 2\%$ and high $minsup = 50\%$ are used, since the association among items is so strong that more than 2,000 QFIs are mined even with this $minsup$. QARs are derived from the QFIs mined by QFIMiner.

The followings two QARs are the examples of the comparative study.

>50K, support=69.6%, conf=95.9%

$\{< age : [29, 62] >, < birth - country : US >,$
$< veteran : Yes >\} => \{< dividends : [0.00, 6500.00] >\}.$

<50K, support=56.2%, conf=99.4%

$\{< age : [0, 45] >, < birth - country : US >,$
$< business : No >\} => \{< dividends : [0.00, 750.00] >\}.$

The first rule is for the persons having more than 50K USD income while the second is for the persons less than 50K USD. Under similar support and confidence, the former is for elders and veterans while the latter is for younger and unemployed persons. The upper bound of the dividends as stock owners is far larger in the former. The next two rules are for a comparative study within the higher income group.
>50K, support=65.6%, conf=85.5%

$\{< veteran : Yes >, < marital - stat : Married >\} =>$

$\{< weeks - worked : [50, 52] >, < dividends : [0.00, 6500.00] >\}.$

>50K, support=65.6%, conf=70.6%

$\{< veteran : Yes >, < dividends : [0.00, 6500.00] >\} =>$

$\{< weeks - worked : [50, 52] >, < marital - stat : Married >\}.$

The former indicates that most married veterans work long and rich, but the latter having less confidence tells that there are not so many rich veterans working long and married. This suggests that the veterans are rich but working hard for it and hence less married(?). The high granularity of the interval values yet maintaining the interpretability shows the high usability of QFIMiner to analyse massive data.

5. Discussion and Conclusion

CLIQUE generates a grid based on the density of instances projected onto every axis [1]. However, the generation is limited only to an equi-width grid at the initial stage of the clustering. MAFIA generates a variable grid by the merge of adjacent bins having similar density on every axis to improve the performance [6]. However, it is also limited to the initial stage. In contrast, QFIMiner evaluates the density in the neighborhood of each transaction projected onto every axis in each candidate dense cluster. This enables finer and more accurate detection of dense clusters while maintaining a good efficiency.

SCHISM applies a variable density threshold based on the average density in every subspace [11]. However, it defines the threshold based on the equi-width grid generated at the initial stage of the clustering similarly to CLIQUE. In contrast, QFIMiner computes the density threshold by the average density in each candidate dense cluster in every subspace. This enables more accurate detection of dense clusters while increasing its efficiency. An advantage of SCHISM is the acceleration of pruning by adding Chernoff-Hoeffding bound to the density threshold under some probability to miss dense clusters. More studies to provide safe and efficient thresholds are needed in the future.

In summary, the performance of QFIMiner is superior to the past subspace clustering approaches in terms of its accuracy and efficiency. It provides highly comprehensive notions of numeric and categorical clusters in form of QFIs. The study on a method to efficiently derive interesting QARs from QFIMiner's output is currently underway.

References

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *Proc. of the 1998 ACM SIGMOD international conference on Management of data*, pages 94–105, June 1998.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proc. of 20th Int. Conf. on Very Large Data Bases (VLDB)*, pages 487–499, 1994.

[3] C.-H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. *Proc. of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 84–93, August 1999.

[4] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *In Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*, pages 226–231, August 1996.

[5] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining with two-dimensional association rules. *ACM Transactions on Database Systems (TODS)*, 26(2):179–213, June 2001.

[6] S. Goil, H. Nagesh, and A. Choudhary. Mafia: Efficient and scalable subspace clustering for very large data sets. *Tech. Report No. CPDC-TR-9906-010, Center for Parallel and Distributed Computing, Dept. of Electrical and Computer Engineering, Northwestern University*, 1999.

[7] K. Kailing, H.-P. Kriegel, and P. Kroger. Density-connected subspace clustering for high-dimensional data. *Proc. Fourth SIAM International Conference on Data Mining (SDM'04)*, pages 246–257, 2004.

[8] B. Liu, Y. Xia, and P. S. Yu. Clustering through decision tree construction. *Proc. of the Ninth International Conference on Information and Knowledge Management*, pages 20–29, 2000.

[9] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. M. Murali. A monte carlo algorithm for fast projective clustering. *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 418–427, June 2002.

[10] R. Rastogi and K. Shim. Mining optimized association rules with categorical and numeric attributes. *Proc. of 14th Int. Conf. on Data Engineering, IEEE Computer Society*, pages 503–512, 1998.

[11] K. Sequeira and M. Zaki. Schism: A new approach for interesting subspace mining. *Proc. of Fourth IEEE International Conference on Data Mining*, pages 186–193, November 2004.

[12] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. *Proc. of 1996 ACM SIGMOD Int. Conf. on Management of Data*, pages 1–12, 1996.

[13] U. C. I. (UCI). *UCI Machine Learning Repository*. UCI, http://www.ics.uci.edu/ mlearn/MLRepository.html, 2004.

[14] K. Wang, S. Hock, W. Tay, and B. Liu. Interestingness-based interval merger for numeric association rules. *Proc. of 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 121–128, 1998.

[15] J. Wijsen and R. Meersman. On the complexity of mining quantitative association rules. *Data Mining and Knowledge Discovery*, 2(3):263–281, September 1998.