Cluster Refinement Using Diversity in Frequent-Pattern based Iterative Projected Clustering

M. Kumara Swamy
International Institute of
Information Technology (IIIT-H)
Gachibowli, Hyderabad, India
kumaraswamy@
research.iiit.ac.in

P. Krishna Reddy
International Institute of
Information Technology (IIIT-H)
Gachibowli, Hyderabad, India
pkreddy@iiit.ac.in

Ankush Jain
International Institute of
Information Technology (IIIT-H)
Gachibowli, Hyderabad, India
ankush.jain@
students.iiit.ac.in

Mayank Gupta
International Institute of
Information Technology (IIIT-H)
Gachibowli, Hyderabad, India
mayank.g@students.iiit.ac.in

ABSTRACT

Clustering is one of the data mining technique to group the data objects into similar clusters. These clustering algorithms do not work efficiently in higher dimensional spaces because of the inherent sparsity of the data. Projected clustering algorithms have been proposed to find the clusters in hidden subspaces. The identification of more relevant attributes is a research issue in the projected clustering. In the literature, an approach has been proposed to find the relevant subspaces using the frequent pattern mining approaches. The frequent pattern can able to identify the item sets with high support. In frequent pattern based projected clustering approach, we observed that only frequent itemsets are considered. In this paper, we propose an approach to refine clusters generated by the frequent pattern based approach and produce more accurate clusters. In the proposed approach, we identify the irrelevant objects in a given cluster by exploiting the notion of diversity. Based on the notion of diversity, which captures the extent the items in the pattern belong to multiple categories of a concept hierarchy, the objects the membership of an object is identified. The membership of each object decides the existance of the object in the cluster. This process iteratively refines each cluster. We conduct the experiments on the real world datasets and show that the proposed approach improve the quality of the cluster.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Theory

Keywords

Data mining, clustering, high dimensional data, diversity

1. INTRODUCTION

Clustering is one of the data mining approachs to find the groups of objects such that the objects in a group will be similar to one another and different from the objects in other groups [17]. The objects are represented as a point, where each dimension corresponds to an attribute/feature and the feature value of each object determines its coefficient in the corresponding dimension. The data points are grouped into clusters based on the some similarity metric. The clustering algorithms have large number of applications such as marketing (e.g., customer segmentation), image analysis, bioinformatics, document classification, indexing, etc.

Most clustering algorithms do not work efficiently in higher dimensional space because of the inherent sparsity of the data [3]. In high dimensional space, it is likely that the distance of any two points is almost the same for a large class of common distributions [5]. So, a clustering algorithm employ feature selection based approach know as subspaces clustering [11]. The goal of the subspace clustering is to find the particular dimensions on which the data points are correlated and pruning away the remaining dimensions that reduces the noise in the data. The problem in these algorithms is that picking certain dimensions in advance can lead to a loss of information. Furthermore, in many real data, some points are correlated with respect to a given set of dimensions and others are correlated with respect to different dimensions. Thus, it may not always be feasible to prune off too many dimensions without considering the data at the same time incurring a substantial loss of information. Alternatively, the effects of dimensionality can be reduced by a dimensionality reduction technique [6], however, information from all dimensions is uniformly transformed and relevant information for some clusters may be reduced. Also, the clusters may hard to understand.

The projected clustering [1] alogirthms overcome some of the issues of the subspace clustering. In projected clustering, a set of data points with an associated set of relevant dimensions are employed such that the data points are similar to each other in the subspace formed by the relevant dimensions, but dissimilar to data point outside the cluster. The widely used distance measures are more meaningful in projections of the high-dimensional space, where the object values are dense [9]. In other words, it is more likely for the data to form dense, meaningful clusters in a hih-dimensional subspace. CLIQUE [3], PROCLUS [1], ORCLUS [2], and DOC [12] are some efforts in the projected clustering approaches. These approaches suffer from the quality of the clusters and consumes lot of time. An effort is made to exploit the frequent pattern mining in the area of projected clustering to address the issue of quality clusters [18, 19]. However, the frequent pattern mining [4] approach identify the itemsets (features/attributes) with high support. In frequent pattern based projected clustering approach, we observed that only frequent itemsets are considered and the infrequent itemsets are completely eliminated from the clustering process.

In this paper, we propose an approach to refine the clusters generated by the frequent pattern based projected clustering approach and produce more qualitative clusters. We observed that the high support pattern include few items and eliminate the non-frequent items. As a result of elimination the some of the items, the influence of other items are not considered in the process of clusters. Each data has some business value. Elimination such valuable data may lost some useful information. Due to elimination of some data values, the the quality of clusters got effected. In the proposed approach, for a given cluster, we identify the membership of each data point employing the notion called diversity. The diversity identify the non-members of the given cluster and remove the objects from the cluster. We conduct the experiment on the real world datasets and show that the proposed approach improve the quality of the cluster.

The rest of the paper is organized as follows. In the next section, we present the related work. In Section 3, we explain an overview of diversity computation of patterns. In Section 4, we explain the proposed approach. In Section 5, we present experimental results. The last section contains summary and conclusions.

2. RELATED WORK

In this section, we first explain the approaches in project clustering, frequent pattern based projected clustering and diversity based approaches.

One of the first algorithms in the area of projected clustering algorithms is CLIQUE [3]. CLIQUE finds the dense regions (clusters) in a level-wise manner, based on the Apriori principle. However, this algorithm does not scale well with data dimensionality. In addition, the formed clusters have large overlap, and this may not generate clear disjoint partitions. PROCLUS [1] and ORCLUS [2] employ alternative techniques. They are much faster than CLIQUE and they can discover disjoint clusters. In PROCLUS, the dimensions relevant to each cluster are selected from the original set of attributes. ORCLUS is more general and can select relevant attributes from the set of arbitrary directed orthogonal vectors. PROCLUS fails to identify clusters with large difference in size and requires their dimensionality to be in a predefined range. ORCLUS may discover clusters that are hard to interpret.

DOC [12] is a density-based algorithm that iteratively discovers projected clusters in a data set. DOC discovers one cluster at a time. At each step, it tries to guess a good medoid for the next cluster to be discovered. It repeatedly picks a random point from the database and attempts to discover the cluster centered at random point. Among all discovered clusters, the cluster with the highest quality is selected. The process repeated for number of random points.

The frequent pattern mining [7] is one of the interesting algorithms in data mining. The frequent itemset mining problem was first proposed in [4]. The frequent patterns are generated from the transactional databases for a given *minsup*. Subsequently, an efficient approach was proposed in FP-growth [8] to generate the frequent patterns without generating the candidate items. In [18, 19], a projected clustering algorithm was proposed. The algorithm iteratively produce one cluster at a time by exploiting the frequent pattern mining approach. The top supported frequent patterns are employed to find the item sets to generate the clusters. A strength function is defined to arrange the clusters in to strong and weak clusters. The strong clusters are produced as an output and the objects in the weak clusters are added by to the database. The process is iterated till all the objects are grouped into the clusters. The FP-growth approach has employed to generate the frequent patterns.

The diversity of patterns in the transactional databases has been studied in [14, 15]. In several applications, it may be useful to distinguish between the pattern having items belonging to multiple categories and the pattern having items belonging to one or a few categories. The notion of diversity captures the extent the items in the pattern belong to multiple categories. The items and the categories form a concept hierarchy. The studies show that the diverse patterns project different knowledge which is not discovered by the frequent patterns. The concept hierarchies were used to find the categories of the items of a pattern.

3. OVERVIEW OF DIVERSITY COMPUTATION OF PATTERNS

In this section, we first explain about the MinClus Algorithm, next we explain the concept hierarchies and finally, we explain the method to compute the diversity of patterns.

3.1 About MinClus Algorithm

The approach works based on the given a random medoid $p \in S$, the appraoch transform best projected cluster containing p, to the trasnactional databases. If the attribute of a record is bounded by p with respect to the width w (here, w=2), an item for that attribute is added to the corresponding itemset.The appraoch observe that all frequent itemsets (i.e., combinations of dimensions) with respect to $minsup = \alpha \times |S|$ are candidate clusters for medoid p. The problem of finding the best projected cluster for a random medoid p can be transformed to the problem of finding the best itemset in a transformation of S, where goodness is defined by the μ function (refer Equation 1). Instead of discovering it in an non-deterministic way, a systematic data mining algorithm on S is applied. The association between the data items is identified by the frequent pattern mining alogirhtm. Due to association in the data items, the frequent pattern mining approach has been exploited in projected cluter approachs. Recently, there is a more efficient algorithm, the FP-growth method [8]. Here, the appraoch adopt frequent pattern mining for subspace clustering. However, the objective is to find the frequent itemset with maximum μ value, rather than finding all frequent subspaces. Assume that I_{best} is the itemset with the maximum μ value found so far and let $dim(I_{best})$ and $sup(I_{best})$ be its dimensionality and support, respectively. Let I_{cond} be the current conditional pattern of the FP-growth process. Its support $sup(I_{cond})$ gives an upper bound for the supports of all patterns containing it. Moreover, the dimensionality of the itemsets that contain I_{cond} is at most $dim(I_{cond}) + l$, where l is the number of items above the items in in the header table of the FP-Growth. The μ function helps to find the strong and weak clusters. Among the strong clusters, the one with highest μ score is prouced as the final cluster and the data in the other clusters are added back to the database for generation of the next clusters. The process is repeated till the now data point is left in the database. The final data points which are not part of any cluster is treated as outliers.

$$\mu(a, b) = a \times (1/\beta)^b \tag{1}$$

where, a is support of the frequent patter, $\beta \in (0, 1]$ reflects the importance of the projection, and b is the number of items in the frequent pattern.

3.2 About Concept Hierarchies

A pattern contains data items. A concept hierarchy is a tree in which the data items are organized in a hierarchical manner. In this tree, all the leaf nodes are the *items*, the internal nodes represent the *categories* and the top node represents the *root*. The *root* could be a virtual node. Figure 1 is an example of the concept hierarchy.

Let C be a concept hierarchy. A node in C may be an item, category or root. The height of the *root* node is 0. Let n be a node in C. The height of n, is denoted as $h(n)$, is equal to the number of edges on the path from *root* to n.

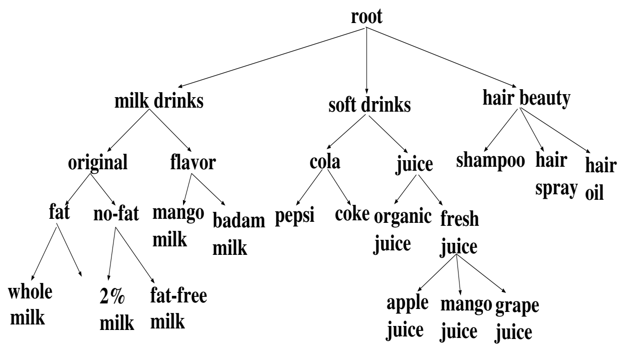

Figure 1: An example of concept hierarchy

The concept hierarchies can be balanced and unbalanced. In the balanced concept hierarchy having height h has the same number of levels. The items at the given height are said to be at the same level. In an unbalanced concept hierarchy, the height of at least one of the leaf level node is different from the height of other leaf level nodes. The height of unbalanced concept hierarchy is equal to the height of the leaf level node having maximum height. In concept hierarchy C, all the lower-level nodes, except the *root*, are mapped to the immediate higher level nodes. We consider the concept hierarchies in which a lower level node is mapped to only one higher level node.

3.3 Concept of diversity

The diversity of a pattern is based on the category of the items within it. If the items of a pattern are mapped to the same/few categories in a concept hierarchy, we consider that the pattern has low diversity. Relatively, if the items are mapped to multiple categories, we consider that the pattern has more diversity. We have developed an approach to assign the diversity for a given pattern based on the merging behavior in the corresponding concept hierarchy. If the pattern merges into few higher level categories quickly, it has low diversity. Otherwise, if the pattern merges into one or a few high level categories slowly, it has relatively high diversity value.

As an example, consider the concept hierarchy in Figure 1. For the pattern {apple juice, mango juice} the items *apple juice* and *mango juice* are mapped to the next level category *fresh juice*. In this case, the merging occurs quickly. For the pattern {whole milk, fat-free mlk}, the items *while milk* is mapped to the category *fat* and item *fat-free milk* is mapped to category *no-fat*. Further, both th ecategories *fat* and *no-fat* are mapped to the category *original*. We say that the pattern {whole milk, fat-free milk} is more diverse than the pattern {apple juice, mango juice} as the merging is relatively slow in case of {whole milk, fat-free milk} as compared to {apple juice, mango juice}. Consider the pattern {2 %milk, mango juice} which is relatively more diverse than the pattern {whole milk, fat-free milk} as both items merges at the *root*. The merging of {2 %milk, mango juice} occurs slowly as comapred to {whole milk, fat-free milk}.

3.4 Computing the Diversity of Patterns

We explain the process of calculating diverse rank of the pattern, called *DRank*, proposed in [14, 15]. We explain the balanced and unbalanced pattern as follows.

We extract the projection of the Y to compute the diversity. We explain the notion of "projection of a pattern" with concept hierarchy which is used to compute the diversity of the pattern.

Projection of Concept Hierarchy for Y **(**$P(Y/C)$**)**: Let Y be P and C be concept hierarchy. The $P(Y/C)$ is the projection of C for Y which contains the portion of C. All the nodes and edges exists in the paths of the items of Y to the *root*, along with the items and the *root*, are included in $P(Y/C)$. The projection $P(Y/C)$ is a tree which represents a concept hierarchy concerning to the pattern Y.

Given two patterns of the same length, different merging behavior can be realized, if we observe how the items in the patterns are mapped to higher level nodes. That is, one pattern may quickly merge to few higher level items within few levels and the other patterns may merge to few higher level items by crossing more number of levels. By capturing the process of merging, we define the notion of diverse rank (*drank*). So, *drank*(Y) is calculated by capturing how the items are merged from leaf-level to *root* in $P(Y/C)$. It can be observed that a given item set maps from the leaf level to the *root* level through a merging process by crossing intermediate levels. At a given level, several lower level items/categories are merged into the corresponding higher level categories.

Two notions are employed to compute the diversity of pattern: *Merging Factor (MF)*, *Level Factor (LF)* and Adjustment factor (AF).

We explain about MF after presenting the notion of generalized pattern.

Generalized Pattern ($GP(Y, l, P(Y/C))$): Let Y be a pattern, 'h' be the height of $P(Y/C)$ and 'l' be an integer. The $GP(Y, l, P(Y/C))$ indicates the GP of Y at level 'l' in $P(Y/C)$. Assume that the $GP(Y, l+1, P(Y/C))$ is given. The $GP(Y, l, P(Y/C))$ is calculated based on the GP of Y at level $(l + 1)$. The $GP(Y, l, P(Y/C))$ is obtained by replacing every item at level $(l+1)$ in $GP(Y, l+1, P(Y/C))$ with its corresponding parent at the level 'l' with duplicates removed, if any.

The notion of merging factor at level l is defined as follows.

Merging factor ($MF(Y, l, P(Y/C))$): Let Y be P and l be the height. The merging factor indicates how the items of a pattern merge from the level $l+1$ to the level l $(0 \leq l < h)$. If there is no change, the MF(Y,l) is 1. If all items merges to one node, the MF(X,l) value equals to 0. So, the MF value at the level l is denoted by $MF(Y, l, P(Y/C))$ which is equal to the ratio of the number of nodes in $(GP(Y, l, P(Y/C) - 1)$ to the number of nodes in $(GP(Y, l+1, P(Y/C) - 1)$.

$$MF(X, l, P(Y/C)) = \frac{|GP(Y,\ l,\ P(Y/C))| - 1}{|GP(Y,\ l+1,\ P(Y/C))| - 1} \qquad (2)$$

We now define the notion of level factor to determine the contribution of nodes at the given level.

Level Factor ($LF(l, P(Y/C))$: For a given $P(Y/C)$, h be the height of $P(Y/C) \neq \{0, 1\}$. Let l be such that $1 \leq l \leq (h-1)$. The LF value of $P(Y/C)$ height l indicates the contribution of nodes at l to *DRank*. We can assign equal, linear or exponential weights to each level. Here, we provide a formula which assigns the weight to the level such that the weight is in proportion the level number.

$$LF(l, P(Y/C)) = \frac{2 * (h-l)}{h * (h-1)} \qquad (3)$$

Adjustment factor (AF): We define the AF at the given level. The *Adjustment Factor (AF)* at level l helps in reducing the *drank* by measuring the contribution of dummy edges/nodes relative to the original edges/nodes at the level l. The AF for a pattern Y at a level l should depend on the ratio of number of real edges formed with the children of the real nodes in $P(Y/E)$ versus total number of edges formed with the children of real and dummy nodes at l in $P(Y/E)$. The value of AF at a given height should lie between 0 and 1. If the number of real edges is equals to zero, AF is zero. If the pattern at the given level does not contain dummy nodes/edges, the value of AF becomes 1. Note that the AF value is not defined at the leaf level nodes as children do not exist. The AF for Y at height l is denoted as $AF(Y, l, P(Y/E))$ and is calculated by the following formula. The value for AF is less than 1 when the pattern is unbalanced otherwise it returns 1.

$$AF(Y, l, P(Y/C)) = \frac{\#\ of\ Real\ Edges\ of\ UP(Y, l, P(Y/E))}{\#\ of\ Total\ Edges\ of\ UP(Y, l, P(Y/E))} \qquad (4)$$

where *numerator* is the number of edges formed with the children of the real nodes and *denominator* is the number of edges formed with the children of both real and dummy nodes at the level l in $P(Y/E)$.

The approach to compute *drank* of P is as follows. We convert the concept hierarchy to extended concept hierarchy called, "extended concept hierarchy" by adding dummy nodes and edges. Next, we adjust the *drank* in accordance with the number of dummy nodes and edges using the notion called adjustment factor. So, the *drank* of P is relative to the *drank* of the same pattern computed by considering all of its items are at the leaf level of the extended concept hierarchy. The dummy nodes and edges are added when the pattern is unbalanced.

Computing the *DRank* for concept hierarchy

The *drank* of P is a function of MF, AF and LF.

Diverse rank of a frequent pattern Y (drank(Y)): Let Y be the pattern and C be the unbalanced concept hierarchy of height 'h'. The *drank* of Y, denoted by *drank*(Y), is given by the following equation.

$$drank(Y, C) = \sum_{l=h-1}^{l=0} [MF(Y, l, P(Y/E)) * AF(Y, l, P(Y/E))] * LF(l, P(Y/E)) \qquad (5)$$

where, h is the height of the $P(P/E)$, E is the extended unbalanced concept hierarchy, $MF(Y, l, P(Y/E))$ is the MF of Y at level l, $LF(l, P(Y/E))$ is the LF at level l and $AF(Y, l, P(Y/E))$ is the AF of Y at level l.

4. PROPOSED APPROACH

From the MinClus [18, 19] projected clustering algorithm, we observed that the strong associated itemsets are generated using the high supported frequent patterns. In the process of projected clustering, only the itemsets in frequent patterns are considered. However, the non-frequent itemsets are not considered in the process of clustering. Generally, the data is generated from the day to day operations of any business. Hence, the data has lot of value for business. However, part of the data is not considered while generating the clusters. This motivated us to investigate an approach to consider all the items that is frequent and non-frequent data items for refining the projected clusters.

In the proposed approach, we refine the projected clusters by considering all the items in the database. The basic idea of the proposed approach is as follows. Each data point membership is exist in one the cluster. If the membership not exist in the cluster that data point is said to be outlier. The memership of each data point in the given cluster is computed employing the diversity. The diversity captures the extent of the items in the pattern belong to multiple categories of a concept hierarchy. We explained more about the diversity in the section 3.3. We compute the diversity of all the data points for all the items (for which the data exist) that is both frequent and non-frequent itemsets. The diversity is able to identify how much diverse the data point from the other data points in the cluster. If the data point is more diverse from the other points, then we can consider the data points is non-member of the cluster, hence we can remove the data point from the cluster otherwise the data point is the member of the cluster. All the data points that are identified as non-members of the cluster are added back to the database for generating the next cluster.

Let D be the database and the $|D|$ contains the n data points. Let C be a cluster contains m data points $\{o_1, o_2, \ldots, {}_m\}$. The membership of each o_i is computed with the notion of diversity. We propose two approaches to refine the clusters.

Diverse square root distance: In this approach, a data point is removed from the cluster whose membeship is does not exist in the current cluser, however the data point may be member of some other cluster. We employ the square root distnace for diversity is computed to find the membership. The diverse square root distnace is identify how much diverse the current point with other data points. If the distance is high that data point is not the member of the current cluster. The approach is as follows. The given cluster is refined by considering the diverse score. We check the membership of each data point with respect to the other data points in the cluster. We compute the diverse suqare root distance (refer Equation 6) for each data point with other data points. If the current data point is farthest from the other data points, then the data point is removed from the cluster. Otherwise, we check move on to the next data point. The process is repeated for all the data points in the cluster. The detail algorithm is present in 1.

$$dsrd = \sqrt{\sum_{j=1}^{n}(d_i - d_j)^2} \tag{6}$$

Diverse Bin: In this approach, the data points are divided into various bins. Generally, the diverse score provide the diversity score in high, medium and low scales. We would like to exploit this notion and find data points with the similar diversity. We assume that the data points with similar diverse score exist at one location. That means the membership of the data points defin by the bin. We explain the approach in the following. The data points are divided into bins based on the diverse score. A user defined threshold *delta* seperate the bins into differnt groups. We compare the diverse score of each successive data points to form the bins. If the difference is higher than *delta* then the data points are seperated into two bins. This process continues till all the data points are made into different bins. Now, we consider one bin where large number of data points exist they are the members of the current cluster. The meaning here is that the data points closely related are the members of the clusters and the others may be members of other clusters. Hence, remove the data points from the other bins.

```
Data: $C, D, \delta$
$C$ is a cluster contains $n$ data points
$D$ is a database
$\delta$ is a user specified threshold
Result: $C$
$d_i$ is the diversity of $i$ data point
for $i \leftarrow 1$ to $n$ do
    $sum = 0$ for $j \leftarrow 1$ to $n$ do
        $sum = sum + \sqrt{(d_i - d_j)^2}$
    end
    if $sum > \delta$ then
        Add C[i] to D
        Remove C[i]
    end
end
```

Algorithm 1: Diverse square root distance

```
Data: $C, D, \delta$
$C$ is a cluster contains $n$ data points order on diversity
$D$ is a database
$\delta$ is a user specified difference
Result: $C$
$d_i$ is the diversity of $i$ data point
$bin\{\}$ is a set of data points with $m$ bins
start = 1
m = 1
for $i \leftarrow 1$ to $n - 1$ do
    a = $d_i - d_{i+1}$
    if $(a > \delta)$ || $(i == (n-1))$ then
        for $j \leftarrow start$ to $i$ do
            Add C[j] to $bin_m$
            Remove C[j]
        end
        m = m + 1
        start = i + 1
    end
end
```

Algorithm 2: Diverse Bin

The basic steps of the approach is as follows.

1. Convert data into transactional database
2. Generate frequent patterns
3. Cluster the data employing the frequent patterns
4. Refine the cluster using the one of the approaches
5. Repeat the steps 2 and 4 until no more data points need to tested
6. Merging of clusters

We explain these steps in details in the following.

1. Convert data into transactional database: We convert the dataset into the transactional database by converting the continuous data into descrite data. These data items are considered as the features or dimensions for the projected clustering. The frequent patterns are generated using the transactional database. In MinClus approach, a random medoid and ω is used to transform the data into transactional data. We found that the approach generate very few patterns and all the data may not be included in the process of generation of frequent patterns. Hence, we convert the data into transactional database and used this database for further process.

2. Generate frequent patterns: There are different frequetn pattern mining algorithms exist in the literature such as Apriori [3], FP-Growth [8]. In this approach, we employed Apriori algorithm to generate the frequent patterns.

3. Cluster the data employing the frequent patterns: We use the μ function shown in the Equation 1 to generate the strong and weak clusters. Among the strong clusters, we output the top cluster as the selected cluster. We remove all the data points from the other clusters and added back for further process.

4. Refine the cluster using the one of the approaches: We employ one of the approaches proposed to refine the cluster generated from step 3. The data points removed from the cluster are added back to the database.

5. Repeat the steps 2 and 4 until no more data points need to tested: From the leftover data points, we generate the frequent patterns, generate the clusters and refine the clusters.

6. Merging of clusters: The merging step is applied only when the user wants at most k clusters in the result. In this case, the strong clusters are merged in an agglomerative way until k clusters remain. Given clusters $C_x$and C_y, the merged cluster is $C_x \cup C_y$, its subspace is $D_x \cap D_y$, its spread is $R(C_x \cup D_y)$ and its μ value is $\mu(|C_x \cup C_y|, |D_x \cap D_y|)$. A good cluster should have small spread and large μ value (i.e., large subspace), so we use both measures to determine the next pair to merge. We consider two rankings of the cluster pairs; one with respect to spread and one with respect to μ value. Then the pair with the highest sum of ranks in both orderings is merged.

Table 1: Data Description

Items	Iris	Zoo	Seeds	Water treatment
# of attributes in original data set	4	36	7	38
# of items in the transactional data set	12	36	21	114
# of levels in concept hierarchy	3	4	2	4

5. EXPERIMENT RESULTS

5.1 Data Description

We conduct the expertiments on realworld datasets taken from UCI machine learning repostory [16]. We have taken four datasets viz., Iris, Zoo, Seeds, and Water-treatment data sets. Concept hierarchies are generated for each data set with the consultation of the domain experts. We provide description about the data in the Table 1.

The performance measures used as accuracy, precision and recall. Accuracy in information-theoretic interpretation of clustering is to view it as a series of decisions, one for each of the $N(N-1)/2$ pairs of documents in the collection. We want to assign two documents to the same cluster if and only if they are similar. A true positive (TP) decision assigns two similar documents to the same cluster, a true negative (TN) decision assigns two dissimilar documents to different clusters. There are two types of errors we can commit. A (FP) decision assigns two dissimilar documents to the same cluster. A (FN) decision assigns two similar documents to different clusters. The *Rand index()* measures the percentage of decisions that are correct [13].

5.2 Experimental Methodology

We generated the transcational database from the datasets. We have not considered the transformation approach specified in [18, 19] without considering the mediod. The advantage is all the items are given equal priority for generation of the frequent patterns. The frequent patterns are generated using the using the Apriori algorithm at *minsup* as 0.2. The MinClus approch expects minimum number of data points per cluster. We provide the minimum number of data points as 5 per cluster. A strength function, μ function, is used to find the strong and weak clusters. Among the strong clusters the the top cluster is produced as a first cluster. On this cluster, we applied the refinement approaches proposed in the paper. The membership of each data point is identified using the diversity score. If the data point is not a member of the cluster then the data point is added back to the database. Similarly, the data in the rest of clusters are added back to the database and applied frequent pattern mining alogrithm for generation of the clusters. This process repeates until all the data points are grouped into the clusters. The unclustered data is greated as outliers. We applied merging process, if user spcify k number of clusters.

5.3 Results

We conducted the experiment on five approaches, existing MinClus, and proposed two approaches. The results are shown in the Tables from 2 to 4.

The experiment results are shown the Table .

Table 2: Accuracy

Data set	MinClus	Approach 1	Approach 2
Iris	81.63%	82.31%	78.46%
Zoo	87.04%	87.41%	86.20%
Seeds	79.08%	79.66%	75.39%
Water Treatmet	60.31%	60.70%	60.78%

Table 3: Precision

Data set	MinClus	Approach 1	Approach 2
Iris	98.33%	98.67%	98.29%
Zoo	93.81%	94.63%	96.17%
Seeds	90.26%	91.65%	90.66%
Water Treatmet	35.02%	36.49%	36.32%

We can observe that the approach 3 out performs the other approaches in accuracy, precision and recall. The other approaches also perform better than the existing MinClus approach.

6. SUMMARY AND CONCLUSIONS

Clustering is a data mining technique to group the data objects into similar clusters. The traditional clustering techniques are inappropriate for high dimensional databases, as the irrelevant attributes add noise. Projected clustering algorithms have been proposed to find the clusters in hidden subspaces. The identification of more relevant attributes is a research issue. In the literature, an approach has been proposed to find the relevant subspaces using the frequent pattern mining approaches. The frequent pattern mining approach identify the item sets with high support. In this paper, we propose an approach to refine clusters generated by the frequent pattern based approach and produce more accurate clusters. In the proposed approach, for a given cluster, we identify the irrelevant data points with the notion of diversity. Based on the notion of diversity, which captures the extent the items in the pattern belong to multiple categories of a concept hierarchy, the objects the membership of an object is identified. The membership of each object decides the existance of the object in the cluster. This process iteratively refine each cluster. The diversity score is able to find the members in the clusters generated using the frequent pattern based projeccted clusters and able refine the clusters.

As a part of the future work, we investigate the improvements to refine the projecte clutering. Explore the diversity in in social networks.

Table 4: Recall

Data set	MinClus	Approach 1	Approach 2
Iris	45.53%	46.43%	34.49%
Zoo	49.41%	49.63%	43.37%
Seeds	41.74%	42.33%	28.42%
Water Treatmet	16.99%	17.62%	17.59%

7. REFERENCES

[1] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. *SIGMOD Rec.*, 28(2):61–72, June 1999.

[2] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. *SIGMOD Rec.*, 29(2):70–81, May 2000.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec.*, 27(2):94–105, June 1998.

[4] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[5] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *Proceedings of the 7th International Conference on Database Theory*, ICDT '99, pages 217–235, London, UK, UK, 1999. Springer-Verlag.

[6] C. Faloutsos and K.-I. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. *SIGMOD Rec.*, 24(2):163–174, May 1995.

[7] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: Current status and future directions. *Data Min. Knowl. Discov.*, 15(1):55–86, 2007.

[8] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *SIGMOD Rec.*, 29(2):1–12, May 2000.

[9] A. Hinneburg, C. C. Aggarwal, and D. A. Keim. What is the nearest neighbor in high dimensional spaces? In *Proceedings of the 26th International Conference on Very Large Data Bases*, VLDB '00, pages 506–515, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[10] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[11] G. Moise, A. Zimek, P. Kr&#x00f6;ger, H.-P. Kriegel, and J. Sander. Subspace and projected clustering: Experimental evaluation and analysis. *Knowl. Inf. Syst.*, 21(3):299–326, 2009.

[12] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. M. Murali. A monte carlo algorithm for fast projective clustering. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, SIGMOD '02, pages 418–427, New York, NY, USA, 2002. ACM.

[13] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[14] S. Srivastava, R. U. Kiran, and P. K. Reddy. Discovering diverse-frequent patterns in transactional databases. In *Proceedings of the 17th International Conference on Management of Data*, COMAD '11, pages 14:1–14:10. Computer Society of India, 2011.

[15] M. K. Swamy, P. K. Reddy, and S. Srivastava. Extracting diverse patterns with unbalanced concept hierarchy. In *Advances in Knowledge Discovery and Data Mining - 18th Pacific-Asia Conference, PAKDD*

2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part I, pages 15–27. Springer-Verlag, 2014.

[16] UCI. *Machine Learning Repository, https://archive.ics.uci.edu/ml/index.html*, 2014 (accessed August, 2014).

[17] R. Xu and D. Wunsch, II. Survey of clustering algorithms. *Trans. Neur. Netw.*, 16(3):645–678, 2005.

[18] M. L. Yiu and N. Mamoulis. Frequent-pattern based iterative projected clustering. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, pages 689–692, Washington, DC, USA, 2003. IEEE Computer Society.

[19] M. L. Yiu and N. Mamoulis. Iterative projected clustering by subspace mining. *IEEE Trans. on Knowl. and Data Eng.*, 17(2):176–189, 2005.