International Journal of Computer Science Engineering & Technology
APPROVED BY
NATIONAL SCIENCE LIBRARY (NSL)
NATIONAL INSTITUTE OF SCIENCE COMMUNICATION AND INFORMATION RESOURCES (NISCAIR)
COUNCIL OF SCIENTIFIC AND INDUSTRIAL RESEARCH (CSIR)– NEW DELHI INDIA .
ISSN :2455-9091

EFFICIENT PROCESSING OF TOP–KDOMINATING QUERIESON INCOMPLETE DATA

[1]MR. R.GOWRI SHANKAR,[2] J.GOKUL,[3] J.SATHISH,[4] J.SRIDHAR RAJA,
[1]Assistant professor,[2,3,4]UG Scholar,
Nehru Institute Of Technology

ABSTRACT: Data mining is a powerful way to discover knowledge within the large amount of the data. Incomplete data is general, finding out and querying these type of data is important recently. The top-k dominating (TKD) queries return k objects that overrides maximum number of objects in a given dataset. It merges the advantages of top-k queries. Traditional query processing techniques that focus on the strict soundness of answer tuples often ignore tuples with critical missing attributes, even if they wind up being relevant to the user query. Ideally, the mediator is expected to retrieve such relevant uncertain answers and gauge their relevance by accessing their likelihood of being relevant answers to the query. The autonomous nature of the databases poses several challenges in realizing this idea. Such challenges include restricted access privileges, limited query patterns and cost sensitivity of database and network resource consumption in web environment. This thesis presents QPIAD – a framework for query processing over incomplete autonomous databases. QPIAD is able to retrieve relevant uncertain answers with high precision, high recall and manageable cost. Data integration over multiple autonomous data sources is an important task performed by a mediator. Extended experimental evaluation using both real and synthetic datasets shows the effectiveness of the developed pruning rules and confirms performance of algorithms.

1. INTRODUCTION

Data mining is a powerful new method to detect knowledge within the large amount of the data. Also data mining is the process of discovering meaningful new relationship, patterns and trends by passing large amounts of data stored in corpus, using pattern recognition technologies as well as statistical and mathematical techniques. Data mining sometimes called data or knowledge mining. Data are any facts, numbers, or sequence of characters that can be processed by a computer. Today, organizations are handling large and growing amounts of data in different structure and different databases.

This is a fundamental problem in data mining with diversified applications in many science and business fields, such as multimedia analysis (motion gesture/video sequence recognition), marketing analytics (buying path identification), and financial modelling (trend of stock prices). Given the overwhelming scale and the dynamic nature of the sequential data, new techniques for sequential pattern analysis are required to derive competitive advantages and unlock the power of the big data.

We need to develop techniques for predicting null values for autonomous databases. This becomes more complicated in autonomous

databases, as the mediator doesn't have access to the entire database and cannot write the predicted value back to the database. Since the mediator does not have access to the entire autonomous database, it has to use a sample of the entire database for predicting missing values. The techniques for predicting missing values have to be robust so that they can be applied for tuples not present in the sample. In other words, using only attribute value correlations from the tuples in the sample would not be effective for predicting missing values for autonomous database. Most autonomous databases do not support binding for null values in their web interfaces. Hence, we need to develop query rewriting techniques in order to retrieve tuples containing null values which might be relevant to a given user query. The mediator has to employ appropriate learning techniques based on the sample database in order to effectively predict null values for possible relevant tuples corresponding to a query. In order to improve the classification accuracy, the mediator can use certain results(without any null values) retrieved during query time to predict missing values for other tuples containing null values. In order to present ranked results to a user query, we need to develop techniques for ranking tuples containing missing values in terms of their relevance to the user query. Specifically, we need to develop appropriate ranking criteria as well as techniques to assign ranks to tuples containing null values. In order to achieve our goal of returning tuples with good precision, recall and manageable cost, the techniques developed should have high prediction accuracy and should minimize network traffic while retrieving possible relevant tuples containing null values.

2. LITERATURE REVIEW

In a real movie recommender system, it is very common that the ratings from some users are missing, because a user tends to only rate those movies he/she knows. As a result, each movie is denoted as a multi-dimensional object with some blank (i.e., incomplete) dimensions. Therefore, the set of movie ratings is incomplete.Although the TKD query over complete data or uncertain data has been well studied, TKD query processing on incomplete data still remains a big challenge. This is because existingtechniques cannot be applied to handle the TKD query over incomplete data efficiently.

In traditional and uncertain databases are not directly applicable to incomplete data.

Skyline Query Processing for Incomplete Data [1]we go beyond the completeness assumptionof multi-dimensional input data where we develop new algorithmsfor efficient computation of skyline queries overincomplete data sets. The main reason for the need of a newset of algorithms for incomplete data is that the transitivedominance relation no longer holds.A Review on Top-K Dominating Queries on Incomplete Data [2] We use an adaptive binning strategy with an efficient method for choosing the appropriate number of bins to minimize the space of bitmap index for IBIG. We propose the improved BIG (termed as IBIG) algorithm to efficiently address the storage issue by using the bitmap compression technique and the binning strategy.Top-k Dominating Queries in Uncertain Databases [3, 4] We propose an effective pruningapproach to reduce the PTD search space, and present an efficient query procedure toanswer PTD queries. Moreover, approximate PTD query processing and the case wherethe PTD query is issued from an uncertain query object are also discussed. Furthermore,we propose an important query type, that is, the PTD query in arbitrary subspaces (namelySUB-PTD), which is more challenging, and provide an effective pruning method to facilitatethe SUB-PTD query processing.Progressive Skyline Computation in Database Systems [5] proposes BBS, a novel algorithm that overcomes all these shortcomings since it is efficient for both progressive and complete skyline computation,independently of the data characteristics (dimensionality, distribution), it

can easily handle userpreferences and process numerous alternative skyline queries (e.g., ranked, constrained, approximate skylines), it does not require any pre-computation (besides building the R-tree), it can be used forany subset of the dimensions, and it has limited main-memory requirements.Efficient Processing of Top-k Dominating Queries on Multi-Dimensional Data [6] we proposedITD, which integrates the algorithm of with our optimizationtechniques (batch counting and Hilbert ordering).Multi-Dimensional Top-k Dominating Queries [7] This query is an important tool for decision supportsince it provides data analysts an intuitive way for findingsignificant objects.

3. PROPOSED WORK

We propose efficient algorithms for processing TKD queries on incomplete data, using several novel heuristics.We present an adaptive binning strategy with an efficient method for choosing the appropriate number of bins to minimize the space of bitmap index for IBIG.We conduct extensive experiments using both real and synthetic datasets to demonstrate the effectiveness of our developed pruning heuristics and the performance of our proposed algorithms.

3.1 Dataset Collection and preprocessing

We have collected huge amount of purchase event data for the customers of a big company. We normalize the original location traces for work- flow modeling. Specifically, we project each raw coordinate to a semantic location of the building, such as a room in the hospital, based on the floor maps of the building. This data preprocessing drastically reduces the computational cost, since we significantly reduce the number of records in the data after the projection. Also, this preprocessing step greatly smooths out the noise and alleviates the impact of errors on the workflow modeling tasks.

3.2 Prediction of buying stages

We plot the embedding of selected 503 events, and mark it with the clustering results. Each detected cluster, we extract dominant semantic keywords for the events in that cluster Semantic information is only used to summarize each temporal cluster for better understanding of our results. Temporal clusters could be partially consistent with attribute-based clusters, while meanwhile revealing more fine-grained structure by exploiting the temporal correlations. This is where the extra value comes from.

3.3 Findout Critical Buying Paths

Detected temporal clusters, we can transform the original event sequences to sequences of temporal clusters, and apply the KNN algorithms on the skeletonized sequences. Nearest Neighbor (KNN from now on) is one of those algorithms that are very simple to understand but works incredibly well in practice. Also it is surprisingly versatile and its applications range from vision to proteins to computational geometry to graphs and so on. Most people learn the algorithm and do not use it much which is a pity as a clever use of KNN can make things very simple.

3.4 Frequent Items mining

We apply the Apriori algorithms on the raw sequences and report for the patterns identification. With a very small support threshold , the maximum pattern length is only three, among which the top 10 with the largest support are listed. However, the resultant patterns mainly represent simple action sequences well expected by common sense.

Note that, there are thousands of patterns returned, which are difficult to be investigated individually.

4. METHODOLOGY

The first algorithm we shall investigate is the k-nearest neighbor algorithm, which is most often used for classification, although it can also be used for estimation and prediction. k-Nearest neighbor is an example of instance-based learning, in which the training data set is stored, so that a classification for a new

unclassified record may be found simply by comparing it to the most similar records in the training set.

We have seen above how, for a new record, the k-nearest neighbor algorithm assigns the classification of the most similar record or records. A distance metric or distance function is a real-valued function d, such that for any coordinates x, y, and z:

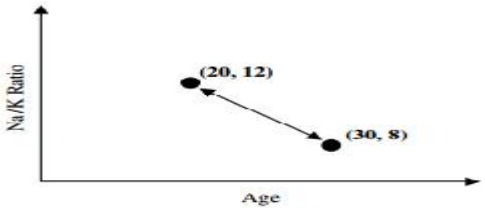
1. $d(x,y) \geq 0$, and $d(x,y) = 0$ if and only if $x = y$
2. $d(x,y) = d(y,x)$
3. $d(x,z) \leq d(x,y) + d(y,z)$

The most common distance function is Euclidean distance, which represents the usual manner in which humans think of distance in the real world:

$$d_{\text{Euclidean}}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

where x = x1, x2, . . . , xm, and y = y1, y2, . . . , ym represent the m attribute values of two records. For example, suppose that patient A is x1 = 20 years old and has a Na/K ratio of x2 = 12, while patient B is y1 = 30 years old and has a Na/K ratio of y2 = 8. Then the Euclidean distance between these points.

$$d_{\text{Euclidean}}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2} = \sqrt{(20-30)^2 + (12 -$$
$$= \sqrt{100 + 16} = 10.77$$



When measuring distance, however, certain attributes that have large values, such as income, can overwhelm the influence of other attributes which are measured on a smaller scale, such as years of service. To avoid this, the data analyst should make sure to normalize the attribute values. For continuous variables, the min–max normalization or Z-score standardization,

Min–max normalization

$$X^* = \frac{X - \min(X)}{\text{range}(X)} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Z-score standardization:

$$X^* = \frac{X - \text{mean}(X)}{\text{SD}(X)}$$

For categorical variables, the Euclidean distance metric is not appropriate. Instead, we may define a function, "different from," used to compare the ith attribute values of a pair of records, as follows:

$$\text{different}(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{otherwise} \end{cases}$$

where xi and yi are categorical values. We may then substitute different (x, i yi) for the ith term in the Euclidean distance metric.

For instance-based learning methods such as the k-nearest neighbor algorithm, it is vitally important to have access to a rich database full of as many different combinations of attribute values as possible. It is especially important that rare classifications be represented sufficiently, so that the algorithm does not only predict common classifications. Therefore, the data set would need to be balanced, with a sufficiently large percentage of the less common classifications. One method to perform balancing is to reduce the proportion of records with more common classifications.

Apriori Algorithm(Frequent Mining)

The Apriori algorithm is the most well known association rule algorithm and is used in most commercial products. The use of support for pruning candidate itemsets is guided by the following principles.

Property 1: If an itemset is sequential, then all of its subsets must also be sequential.

Property 2: If an itemset is insequential, then all of its supersets must also be insequential.

The algorithm initially scans the database to count the support of each item. Upon completion of this step, the set of all sequential 1-itemsets, F1, will be known. Next, the algorithm will iteratively generate new candidate kitemsets using the sequential (k-1)-itemsets found in the previous iteration.

Candidate generation is implemented using a function called Apriori-gen.

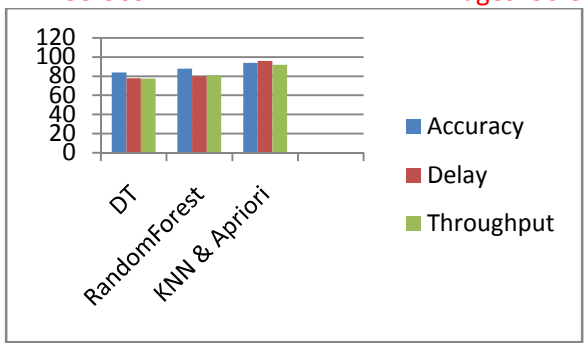
5. Experimental Result and Discussion

In order to verify the performance of the proposed algorithm, we compare it with Apriori algorithm. These algorithms are performed on a computer with a 2.00GHz processor and 512MB memory, running windows vista. The program is developed by Java with Mysql. We present experimental results using the database. The experimental result is showed in Figures. As shown in the Figure, proposed algorithm is more super than apriori ,because it dosen't need to generate 2-candidate itemsets and reduce the search space, and proposed algorithm dosen't need to much extra spaces on the mining process, so proposed algorithm has a better space scalability

Compared to existing algorithms our performance is increased. The below tables represent the accurate values of current process and existing values.

Technique	Accuracy	Delay	Throughput
Decision Tree	84%	78%	77.5%
Random Forest	88%	80%	81%
KNN & Apriori	94%	96%	92%

Table 1- Performance Table The accuracy rate obtained by applying the classification algorithms on the data sets.



The individual accuracy rates obtained from different feature selection methods on the classifier. Different feature selection metrics are applied on the classifier.

CONCLUSION

In this paper, a algorithm is proposed which combined Apriori algorithm and the KNN. The experimental results shows that this new algorithm works much faster than Apriori. The future work is to optimize the technique for counting the support of the candidates and expand it for mining more larger database. In the future, to cope with different applications, it is interesting to generalize these works with an unified probabilistic framework which simultaneously identifies the pattern granularity levels and estimates the model parameters.

REFERENCE

[1]. W.-T. Balke, U. G¨untzer, and J. X. Zheng. Efficient DistributedSkylining for Web Information Systems. In EDBT, 2004.

[2]. S. B¨orzs¨onyi, D. Kossmann, and K. Stocker. The Skyline Operator.In ICDE, 2001.

[3]. A. R. Butz. Alternative Algorithm for Hilbert's Space-FillingCurve. IEEE Trans. Comput., C-20(4):424–426, 1971.

[4]. C.-Y. Chan, P.-K. Eng, and K.-L. Tan. Stratified Computation ofSkylines with Partially-Ordered Domains. In SIGMOD, 2005.

[5]. C.-Y. Chan, H. Jagadish, K.-L. Tan, A. Tung, and Z. Zhang. Findingk-Dominant Skylines in High Dimensional Space. In SIGMOD,2006.

[6]. C.-Y. Chan, H. Jagadish, K.-L. Tan, A. Tung, and Z. Zhang. OnHigh Dimensional Skylines. In EDBT, 2006.
[7]. S. Chaudhuri, N. Dalvi, and R. Kaushik. Robust Cardinality andCost Estimation for Skyline Operator. In ICDE, 2006.
[8]. J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. Skyline with Presorting.In ICDE, 2003.
[9]. R. Fagin, A. Lotem, and M. Naor. Optimal Aggregation Algorithmsfor Middleware. In PODS, 2001.
[10]. P. Godfrey. Skyline Cardinality for Relational Processing. InFoIKS, 2004.
[11]. P. Godfrey, R. Shipley, and J. Gryz. Maximal Vector Computationin Large Data Sets. In VLDB, 2005.
[12]. A. Guttman. R-Trees: A Dynamic Index Structure for SpatialSearching. In SIGMOD, 1984.
[13]. G. R. Hjaltason and H. Samet. Distance Browsing in Spatial Databases, TODS, 24(2):265–318, 1999.
[14]. V. Hristidis, N. Koudas, and Y. Papakonstantinou. PREFER: ASystem for the Efficient Execution of Multiparametric RankedQueries. In SIGMOD, 2001.
[15]. Z. Huang, C. S. Jensen, H. Lu, and B. C. Ooi. Skyline QueriesAgainst Mobile Lightweight Devices in MANETs. In ICDE, 2006.
[16]. D. Kossmann, F. Ramsak, and S. Rost. Shooting Stars in the Sky:An Online Algorithm for Skyline Queries. In VLDB, 2002.
[17]. I. Lazaridis and S. Mehrotra. Progressive Approximate AggregateQueries with a Multi-Resolution Tree Structure. In SIGMOD,2001.
[18]. S. T. Leutenegger, J. M. Edgington, and M. A. Lopez. STR: ASimple and Efficient Algorithm for R-Tree Packing. In ICDE,1997.
[19]. C. Li, K. C.-C. Chang, and I. F. Ilyas. Supporting Ad-hoc RankingAggregates. In SIGMOD, 2006.
[20]. C. Li, B. C. Ooi, A. Tung, and S. Wang. DADA: A Data Cube forDominant Relationship Analysis. In SIGMOD, 2006.
[21]. X. Lin, Y. Yuan, W. Wang, and H. Lu. Stabbing the Sky: EfficientSkyline Computation over Sliding Windows. In ICDE, 2005.
[22]. X. Lin, Y. Yuan, Q. Zhang, and Y. Zhang. Selecting Stars: The kMost Representative Skyline Operator. In ICDE, 2007.
[23]. D. Papadias, P. Kalnis, J. Zhang, and Y. Tao. Efficient OLAP Operationsin Spatial Data Warehouses. In SSTD, 2001.
[24]. D. Papadias, Y. Tao, G. Fu, and B. Seeger. Progressive SkylineComputation in Database Systems. TODS, 30(1):41–82, 2005.
[25]. J. Pei, A.W.-C. Fu, X. Lin, and H.Wang. Computing CompressedMultidimensional Skyline Cubes Efficiently. In ICDE, 2007.
[26]. J. Pei, W. Jin, M. Ester, and Y. Tao. Catching the Best Views ofSkyline: A Semantic Approach Based on Decisive Subspaces. InVLDB, 2005.
[27]. J. Pei, Y. Yuan, X. Lin, W. Jin, M. Ester, Q. Liu, W. Wang, Y. Tao,J. X. Yu, and Q. Zhang. Towards Multidimensional Subspace SkylineAnalysis. TODS, 31(4):1335–1381, 2006.