AN EFFECTIVE APPROACH TO COMPUTE DISTANCE OF UNCERTAIN DATA BY USING K-NEAREST NEIGHBOR

Suman Bhakar [#1], **Priya.S** [*2]

[#] *Final year, M.Tech/CSE, SRM University*
104,Pearl Excellency Gandhi Nagar ,Jaipur – 302014, Rajasthan,
India.
[1] bhakar2010@gmail.com
[*] *Assistant Professor, SRM University*
Department of Computer Science and Engineering, SRM University, Kattankulathur, Kancheepuram DT – 603203, India
[2] priya.sn@ktr.srmuniv.ac.in

Abstract — **Clustering uncertain data has been well recognized as an important issue. Generally, an uncertain data object can be represented by a probability distribution. In Existing system, Kullback-Leibler (KL) divergence approach from information theory is used to measure the similarity of uncertain data. This approach work is based on probability mass function calculation, where continuous and discrete values of uncertain data are calculated. By using the probability mass function, distance value of continuous and discrete cases are calculated individually. The probabilistic ratio of both the cases is used to calculate the similarity. Then, the Density based clustering approach is used to cluster the uncertain data. However, choosing the nearest neighbor is a drawback in the existing system. To overcome this issue proposed system introduces K-nearest-neighbor algorithm to calculate the nearest neighbor. The K-nearest-neighbor (KNN) algorithm measures the distance between a query scenario and a set of scenarios in the data set. Here, distance is calculated for both continuous and discrete cases by using Probability mass function. Then, nearest neighbor is calculated by applying KNN approach. Thus, proposed system overcomes the existing drawback and produce effective result.**

Keywords— **Clustering, Uncertain Data, density estimation, Probability mass function**

I. INTRODUCTION

Clustering is the process of making group of abstract objects into classes of similar objects. A cluster of data objects can be treated as a one group. While doing the cluster analysis, first partition the set of data into groups based on data similarity and then assigns the label to the groups. Clustering is the problem of partitioning a given set of objects into subsets of similar objects. It has application in various areas of computer science such as machine learning, data compression, data mining, or pattern recognition. Depending on the application we want to cluster such diverse objects as text documents, probability distributions, feature vectors, etc. Obviously, different objects and different applications also require different notions of dissimilarity of objects. As a consequence, there are numerous different formulations of clustering.

First step towards understanding clustering problems with non-metric dissimilarity measures, like Kullback-Leibler divergence. A problem that is relatively well understood in the case of Euclidean and metric distances: k-median clustering. In k-median clustering we have a representative (sometimes called prototype) for each cluster. In the geometric version of the problem this is the cluster center. Minimizing the sum of error of the clustering, i.e. the error that is made by representing each input object by its corresponding representative.

Since non-metric dissimilarity measures, this version of k-median also captures other variants like the well known Euclidean k-means clustering, where the goal is to minimize the sum of squared errors (with respect to Euclidean distance). For instance, sensor measurements may be imprecise at a certain degree due to the presence of various noisy factors (e.g., signal noise, instrumental errors, and wireless transmission). at a certain degree due to the presence of various noisy factors (e.g., signal noise, instrumental errors, and wireless transmission).

Clustering Analysis is broadly used in many applications such as market research, pattern recognition, data analysis, and image processing. Clustering can also help marketers discover distinct groups in their customer basis. And they can characterize their customer groups based on purchasing patterns. In field of biology it can be used to derive plant and animal taxonomies, categorize genes with similar functionality and gain insight into structures inherent in populations. Clustering also helps in identification of areas of similar land use in an earth observation database. It also helps in the identification of groups of houses in a city according house type, value, and geographic location. Clustering also helps in classifying documents on the web for information discovery. Clustering is also used in outlier detection applications such as detection of credit card fraud.

As a data mining function, Cluster Analysis serve as a tool to gain insight into the distribution of data to observe characteristics of each cluster.

The remainder of this paper is organized as follows. Section II surveys related work, whereas Section III describes the KL Divergence Algorithm. Section IV describes the K-Nearest Neighbor (KNN) algorithm to find the nearest neighbour in uncertain data, whereas Section V describes the system architecture of the proposed system. Section VI and VII illustrates the design implementation of clustering uncertain data. Section VIII illustrates the performance evaluation of the proposed system. Finally, Section IX concludes the paper.

II. Related Work

Clustering is the main approach in data mining task. Clustering of uncertain data brings a new challenge of data uncertainity. Most studies of clustering uncertain data used density estimation, which are reviewed in the following section.

A. Fast Gauss Transform and Efficient Kernel Density Estimation

Evaluating sum of multivariate Gaussians is a common computational task in computer vision and pattern recognition, including in the general and powerful kernel density estimation technique. The quadratic computational complexity of the summation is a significant barrier to the scalability of this algorithm to practical applications. The fast Gauss transform FGT[4] has successfully accelerated the kernel density estimation to linear running time for low dimensional problems. By higher dimensions, mean dimensions up to ten. Such high dimensional spaces are commonly used in many applications such as in video sequence analysis and eigen space based approaches. It also shows how the IFGT can be applied to the kernel density estimation.

The proposed IFGT successfully reduced the computational complexity into linear time. The success in acceleration of the FGT comes from two innovations: the use of the farthest-point algorithm to adaptively subdivide the high dimensional space, and the use of a new multivariate Taylor expansion we developed to dramatically reduce the computational and storage cost of the fast Gauss transform. The recursive computation of the multivariate Taylor expansion further reduces the computational cost and necessary storage.

B. Text Indexing Multi-Dimensional Uncertain Data with Arbitrary Probability Density Functions.

In an “uncertain database”, an object *o* is associated with a multi-dimensional probability density function MPDF [7], which describes the likelihood that *appears* at each position in the data space. A fundamental operation is the “probabilistic range search” which, given a value *pq* and a rectangular area *rq*, retrieves the objects that appear in *rq* with probabilities at least *pq*. Our system presents the U-tree, a multi-dimensional access method on uncertain data with arbitrary pdfs. This structure minimizes the amount of appearance probability computation in prob-range search. Intuitively, it achieves this by pre-computing some “auxiliary information” for each object, which can be used to disqualify the object (in executing a query) or to validate it as a result without having to obtain its appearance probability. Such information is maintained at all levels of the tree to avoid accessing the sub trees that do not contain any results. It presented a careful study of the probabilistic range search problem on uncertain data. Our solutions can be applied to objects described by arbitrary pdfs, and process queries efficiently with small space.

C. Monte Carlo Database for uncertain data

Managing uncertain data has been explored in many ways. Different methodologies for data storage and query processing have been proposed. As the availability of management systems grows, the research on analytics of uncertain data is gaining in importance. While different approaches for uncertain data management have been proposed and much work has been done on uncertain data mining, no literature can be found in combining these two principles. To tackle this problem, here a new method is proposed to cluster uncertain data on the base of the previously motioned MCDB [8] approach in this project. By having the same computational principle in the mining algorithm and the database system, the already established tight integration of data mining algorithms and database system is continued for the uncertain data field.

In this project, the proposed method is similar to the Monte Carlo Database for uncertain data to be able to include the algorithm into the database management system. The method is divided into three steps. In the first step, multiple possible worlds are generated from a dataset according to their uncertainty definition. In the second step, a cluster model is built for each world. For a final clustering the local clustering, results are aggregated into one clustering.

III. KL-Divergence Algorithm

It is natural to quantify the similarity between two uncertain objects by KL divergence.

Given two uncertain objects P and Q and their corresponding probability distributions, D(P||Q) evaluates the relative uncertainty of Q given the distribution of P. We have

$$D(P||Q) = E\,\log\frac{P}{Q} \qquad (1)$$

which is the expected log-likelihood ratio of the two distributions and tells how similar they are. The KL divergence is always nonnegative, and satisfies Gibbs' inequality. That is, D(P||Q) >=0 with equality only if P=Q.

In the discrete case, it is straightforward to evaluate (4) to calculate the KL divergence between two uncertain objects P and Q from their probability mass functions calculated as (2). In the continuous case, given the samples of P and Q, by the law of large numbers, we have

$$\lim_{s\to\infty} 1/s \sum_{i=1}^{s} \log P(p_i)/Q(p_i) = D(P||Q) \qquad (2)$$

where we assume the sample of P ={P1,P2, …Ps}. Hence, we estimate the KL divergence D(P||Q) as

$$D(P||Q) = \frac{1}{s}\sum_{i=1}^{s} \log P(p_i)/Q(p_i) \qquad (3)$$

It is important to note that the definition of KL divergence necessitates that for any x ϵ ID if P (x) > 0 then Q (x) > 0. To ensure that the KL divergence is defined between every pair of uncertain objects, we smooth the probability mass/density function of every uncertain object P so that it has a positive probability to take any possible value in the domain.

IV. KNN Approach

K-Nearest Neighbors (KNN) classification divides data into a test set and a training set. For each row of the test set, the K nearest (in Euclidean distance) training set objects are found, and the classification is determined by majority vote with ties broken at random. If there are ties for the Kth nearest vector, all candidates are included in the vote.

Suppose each sample in our data set has n attributes which we combine to form an n-dimensional vector: x = (x_1, x_2, . . x_n). These n attributes are considered to be the independent variables. Each sample also has another attribute, denoted by y (the dependent variable), whose value depends on the other n attributes x. We assume that y is a categoric variable, and there is a scalar function, f, which assigns a class, y = f(x) to every such vectors. We do not know anything about f (otherwise there is no need for data mining) except that we assume that it is smooth in some sense.

Suppose that a set of T such vectors are given together with their corresponding classes: x(i), y(i) for i = 1, 2, . . . , T. This set is referred to as the training set. The problem we want to solve is the following. Supposed we are given a new sample where x = u. We want to find the class that this sample belongs. If we knew the function f , we would simply compute v = f(u) to know how to classify this new sample, but of course we do not know anything about f except that it is sufficiently smooth.

The idea in k-Nearest Neighbor methods is to identify k samples in the training set whose independent variables x are similar to u, and to use these k samples to classify this new sample into a class, v. If all we are prepared to assume is that f is a smooth function, a reasonable idea is to look for samples in our training data that are near it (in terms of the independent variables) and then to compute v from the values of y for these samples. When we talk about neighbors we are implying that there is a distance or dissimilarity measure that we can compute between samples based on the independent variables. The Euclidean distance between the points x and u is

$$d(x,u) = \sqrt{\sum_{i=1}^{n}(x_i - u_i)^2} \quad (4)$$

The systems examine other ways to measure distance between points in the space of independent predictor variables when we discuss clustering methods.

The simplest case is k = 1 where we find the sample in the training set that is closest (the nearest neighbor) to u and set v = y where y is the class of the nearest neighboring sample. It is a remarkable fact that this simple, intuitive idea of using a single nearest neighbor to classify samples can be very powerful when we have a large number of samples in our training set. It is possible to prove that if we have a large amount of data and used an arbitrarily sophisticated classification rule, we would be able to reduce the misclassification error at best to half that of the simple 1-NN rule. For k-NN we extend the idea of 1-NN as follows. Find the nearest k neighbors of u and then use a majority decision rule to classify the new sample. The advantage is that higher values of k provide smoothing that reduces the risk of over-fitting due to noise in the training data. In typical applications k is in units or tens rather than in hundreds or thousands. Notice that if k = n, the number of samples in the training data set, we are merely predicting the class that has the majority in the training data for all samples irrespective of u. This is clearly a case of over-smoothing unless there is no information at all in the independent variables about the dependent variable.

V. System Architecture

Design is the only ways that can accurately translating a customer's requirements in to a finished software product. Fig 1 shows the process through which the requirements are translated in to a representation of the software i.e. the blue print for constructing software.

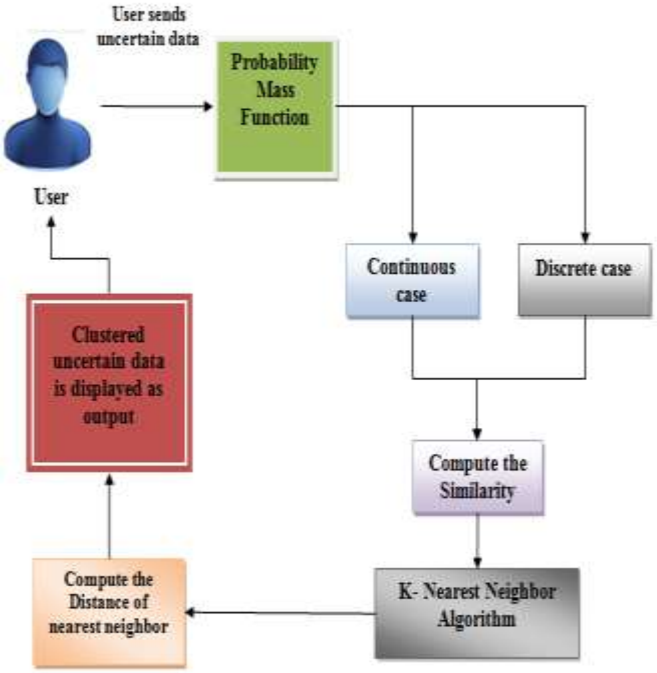
User sends uncertain data
User
Probability Mass Function
Continuous case
Discrete case
Compute the Similarity
K- Nearest Neighbor Algorithm
Compute the Distance of nearest neighbor
Clustered uncertain data is displayed as output

Fig. 1 System Architecture

VI. Design for Clustering of Uncertain Data

A. User Authentication

B. KL Divergence approach

C. KNN approach

D. Data Transmission

VII. Design Implementation

A. *User Authentication*

In Fig 2 contains authentication details between client and server. First users make registration by entering the required details and sent to server. Server validates the user details and sent authentication details.

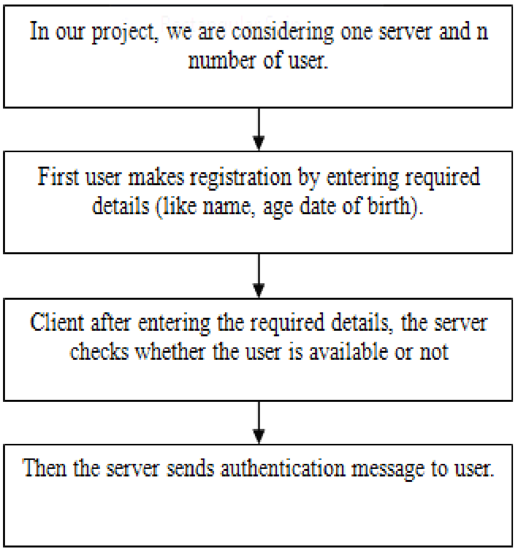
In our project, we are considering one server and n number of user.
First user makes registration by entering required details (like name, age date of birth).
Client after entering the required details, the server checks whether the user is available or not
Then the server sends authentication message to user.

Fig. 2 User Authentication

B. *KL divergence approach*

KL divergence is an important approach. This approach finds the finds the similarity of uncertain data. In given Fig 3, the similarity is more helpful to cluster the data. Then cluster the uncertain data according to the similarity.

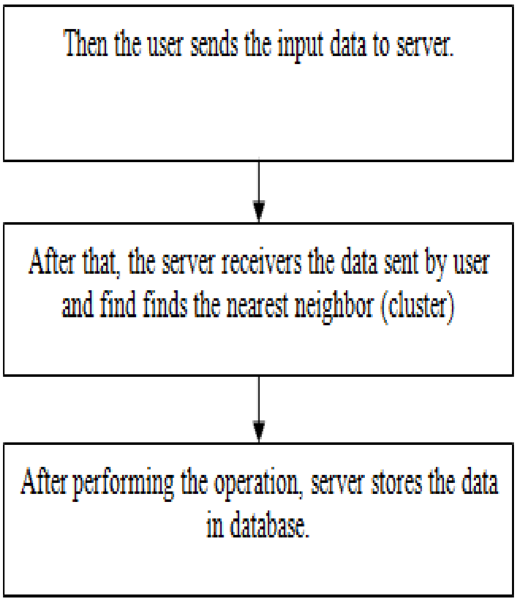
Then the user sends the input data to server.
After that, the server receivers the data sent by user and find finds the nearest neighbor (cluster)
After performing the operation, server stores the data in database.

Fig. 3 KL Divergence Method

C. *KNN approach*

After partitioning the datasets, the partitioned data sets are taken as input for KNN approach. This KNN approach finds the distance between every two node. Then if the distance is nearest than other node then that node is considered as nearest node which is shown in Fig. 4.

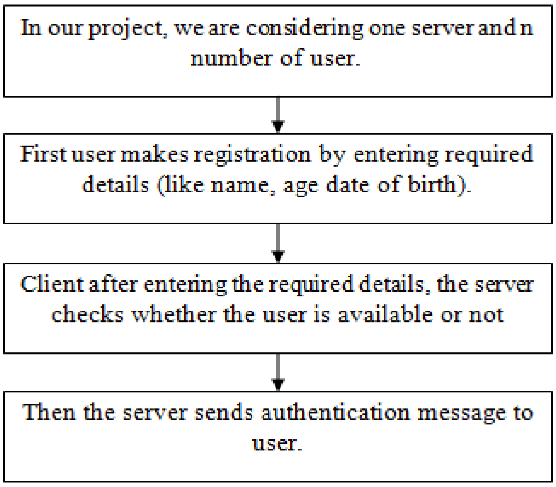
In our project, we are considering one server and n number of user.
First user makes registration by entering required details (like name, age date of birth).
Client after entering the required details, the server checks whether the user is available or not
Then the server sends authentication message to user.

Fig. 4 KNN Approach

D. *Data transmission to user*

After clustering the dataset the server sends the dataset to requested user. Then the user receives the dataset and views. This module contains details about transmitting the data to user after clustering the data is shown in Fig. 5.

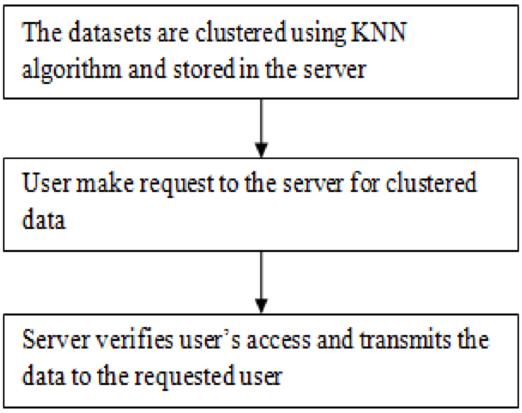
The datasets are clustered using KNN algorithm and stored in the server
User make request to the server for clustered data
Server verifies user's access and transmits the data to the requested user

Fig. 5 Data Transmission To User

VIII. Evaluation

In the existing system, KL Divergence method and density method is used for clustering, that does not provide proper clustering between dataset. KNN Algorithm is used along with the KL Divergence algorithm for proper clustering that provides more performance and efficiency with lesser computation for probability distribution than the existing approach.

IX. Conclusion and Future Work

A new mechanism is proposed for clustering uncertain data. First the user is registered with server and the server verifies the user's details with the database. After verification user send the uncertain data to the server. The server uses KLL divergence mechanism for classifying discrete and continuous case data and computes the similarity of the data. Finally apply K-NN algorithm to compute the distance between the nearest nodes and cluster the data. This method provides efficient clustering of uncertain data compared to other clustering methods. The future enhancement of this work is to implement an advanced clustering algorithm for clustering uncertain data.

REFERENCES

[1] A. Banerjee, S. Merugu, I.S. Dhillon, and J. Ghosh, “Clustering with Bregman Divergences,” J. Machine Learning Research, vol. 6, pp. 1705-1749, 2005.

[2] B. Kao, S.D. Lee, D.W. Cheung, W.-S. Ho, and K.F. Chan, “Clustering Uncertain Data Using Voronoi Diagrams,” Proc. IEEE Int’l Conf. Data Mining (ICDM), 2008.

[3] B. Kao, S.D. Lee, D.W. Cheung, W.-S. Ho, and K.F. Chan,“Clustering Uncertain Data Using Voronoi Diagrams,” Proc. IEEE Int’l Conf. Data Mining (ICDM), 2008.

[4] C. Yang, R. Duraiswami, N.A. Gumerov, and L.S. Davis, “Improved Fast Gauss Transform and Efficient Kernel Density Estimation,” Proc. (IEEE Int’l Conf. Computer Vision (ICCV), 2003.)

[5] D.M. Blei, A.Y. Ng, and M.I. Jordan, “Latent Dirichlet Allocation,”J. Machine Learning Research, vol. 3, pp. 993-1022, 2003.

[6] F. Song and W.B. Croft, “A General Language Model for Information Retrieval,” Proc. Int’l Conf. Information and Knowledge Management (CIKM), 1999.

[7] Y. Tao, R. Cheng, X. Xiao, W.K. Ngai, B. Kao, and S. Prabhakar, “Indexing Multi-Dimensional Uncertain Data with Arbitrary Probability Density Functions,” Proc. (Int’l Conf. Very Large Data Bases (VLDB), 2005.

[8] R. Jampani, F. Xu, M. Wu, L.L. Perez, C.M. Jermaine, and P.J. Haas, “Mcdb: A Monte Carlo Approach to Managing Uncertain Data,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD), 2008.

[9] M.R. Ackermann, J. Blo¨mer, and C. Sohler, “Clustering for Metric and Non-Metric Distance Measures,” Proc. (Ann. ACM-SIAM Symp. Discrete Algorithms (SODA), 2008)

[10] M.R. Ackermann, J. Blo¨mer, and C. Sohler, “Clustering for Metric and Non-Metric Distance Measures,” Proceeding in ACM-SIAM Symposuim. Discrete Algorithms (SODA), 2008.

[11] R. Cheng, D.V. Kalashnikov, and S. Prabhakar, “Evaluating Probabilistic Queries over Imprecise Data,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD), 2003.

[12] N.N. Dalvi and D. Suciu, “Management of Probabilistic Data:Foundations and Challenges,” Proc. ACM SIGMOD-SIGACTSIGART Symp. Principles of Database Systems (PODS), 2007.

[13] P.B. Volk, F. Rosenthal, M. Hahmann, D. Habich, and W. Lehner, “Clustering Uncertain Data with Possible Worlds,” Proc. IEEE Int’l Conf. Data Eng. (ICDE), 2009

[14] T. Feder and D.H. Greene, “Optimal Algorithms for Approximate Clustering,” Proc. Ann. ACM Symp. Theory of Computing (STOC),1988.

[15] T.F. Gonzalez, “Clustering to Minimize the Maximum Intercluster Distance,” Theoretical Computer Science, vol. 38, pp. 293-306, 1985.

[16] T. Imielinski and W.L. Lipski Jr., “Incomplete Information in Relational Databases,” J. ACM, vol. 31, pp. 761-791, 1984.

[17] V. Cerny, “A Thermodynamical Approach to the TravellingSalesman Problem: An Efficient Simulation Algorithm,”J. Optimization Theory and Applications, vol. 45, pp. 41-51, 1985.

[18] W.K. Ngai, B. Kao, C.K. Chui, R. Cheng, M. Chau, and K.Y. Yip, “Efficient Clustering of Uncertain Data,” Proc. (Sixth Int’l Conf. Data Mining (ICDM), 2006.)