Indian Journal of Science and Technology, *Vol 9(8), DOI: 10.17485/ijst/2016/v9i8/87969, February 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Stage-by-Stage Pruning Method for Classifying Uncertain Data Streams

S. Subashini[1]* and S. Appavu alias Balamurugan[2]

[1]Department of Computer Science Engineering, Fatima Michael College of Engineering and Technology, Madurai – 625020, Tamil Nadu, India; jessuba30@gmail.com
[2]Department of Information Technology, K.L.N. College of Information and Technology, Madurai – 630612, Tamil Nadu, India; app_s@yahoo.com

Abstract

Background: We study an important problem of similarity grouping processing on stream data that inherently contain uncertainty. **Method:** In this paper SBSP – [Stage by Stage Pruning] a novel pruning method is proposed for fast, accurate clustering and classifying the data where the two stages were grouped into a single framework MYFRAME. **Findings:** The proposed approach group the data-by-data level pruning using Manhattan distance in first stage. In the second stage, the data is grouped by object level pruning in hyperspace**. Improvements:** Currently, this approach is applied in real time applications such as object detection, video retrieval, people detection and tracking, earth quake monitoring etc.

Keywords: Clustering, Data Pruning, Distance, Group Nearest Neighbor, Grouping Process, Similarity Search, Uncertain Data Streams

1. Introduction

The greatest common text solemnization for text cataloging is the special data model initiated on the bag of words/expressions representation. The foremost advantage of the special data prototypical is that it can voluntarily be employed by classification algorithms. However, the bag of words/expressions representation is matched to catching only word/expression frequency; organizational and semantic information is discounted. It has been recognized that organizational information plays a significant role in classification accuracy. The data clustering depends on the dimensionality and similarity features of the data[1,2]. It finds groups of data under the homogeneity of the data according to similarity measurement among the data. Generally, the given set of data is partition into subset of data. Using K–means, C-means or K-Medoids method which computes a representative point per group and assign each object to the group with closest representative, so that sum of the squared differences between the objects and their representatives is minimized.

Furthermore, many groups may be occurred in different subsets, comprised of varies combinations of features. Currently, some points are correlated with respect to a given set of dimensions and others are correlated with respect to various dimensions. Every dimension could be relevant to at least one of the groups. To do the group analysis, the high dimensionality data needs to be specified in a subset of dimension. The partitioning and grouping of the data is one of the most useful tasks in data mining for high dimensional data set. Thus, the aim of the grouping is to partition a data set into sub groups such a way that objects in each particular group are similar and object in various groups are dissimilar. In real world application, most of the algorithms first choose the number of groups. In this paper, the grouping of uncertain data streams is clustered by a two stage pruning method.

In computer vision, for instance, subspaces are frequently used to catch the presence of objects under different lighting[3,4], viewpoint[5,6], spatial transformations (e.g., utilizing the tangent distance[7], articulation[8,9], identity[10,11], classes of comparable items[12,13] and more. Regularly, given

*Author for correspondence

a query image (or images) of an object, represented as a point (or as a subspace) in high-dimensional space, a database of subspaces is scanned for the subspace closest to the query. A natural issue which emerges from this sort of search problems is: Can the nearest (or a near) subspace is discovered faster than a brute energy sequential search through the whole database.

In that capacity it has attracted in considerable attention in recent years, and various efficient algorithms for Approximated Nearest Neighbor (ANN) pursuit have been proposed[14–17]. These algorithms accomplish sublinear pursuit times when placing a near, not so much the nearest neighbor, suffices. The gain in query speed is accomplished at the price of preprocessing the database.

A lot of Spatial Access Methods (SAM) was proposed for multidimensional data. A comprehensive survey showing the evolution of SAM and their main concepts can be found in[18]. However, the majority of them cannot index data in metric domain and suffer from the dimensionality curse, being efficient to index only low-dimensional datasets. A lopsided R-tree called CUR-tree (Cost-Based Unbalanced R-tree) was proposed in[19] to optimize query executions. It uses promotion and demotion to move data objects and sub-trees around the tree considering a given query distribution and a cost model for their execution. Considering cost model shows, a great deal of work were also published regarding to SAM[20]. In any case they depend on data distribution in the space and other spatial properties, what turns them infeasible for MAM. The Techniques of recursive dividing of data in metric domains proposed[21] were the starting stage for the development of MAM.

In existing similarity looks about whether series databases, the time series is changed from its original form (Xm, tm || m = 1, 2, … L) into a more compact representation. The search algorithm leverages on two steps: dimensionality reduction[22–27] and data representation in the transformed space.

Different dimensionality reduction strategies have been proposed for time arrangement data transformation. This incorporates: Discrete Fourier Transform (DFT), Singular Value Decomposition (SVD)[1,2]. In summary, none of the above methods can address the challenges of effective and efficient SBSP comparability search. The existing methods lack a compact, powerful and measurable representation for the patterns and the natural relations that are inherent in the SBSPs. In addition, it is hard to recognize a generic relation descriptor that can be applied to different application domains.

2. Problem Statement

In data mining, the accuracy and speed of the mining process depends on the data model and structure of the data. The model, structure of the data can be obtained by pre-processing and normalization with clustering and classification. After classification, the mining process becomes fast and mine accurate, with relevant data from the data model for the query object. It is necessary to change the raw data into a data model which improve efficiency of data mining. In this paper, a stage by stage pruning method is used for pre-processing and normalizing an uncertain data. The overall functionality of the proposed approach in this paper is shown in the System Model.

3. System Model

The proposed approach of this paper is performed as a sequence of steps in such a manner that the uncertain data stream is clustered and classified using the similar feature based grouping. Initially, the data is read and analyzed. Then the data is preprocessed and normalized. The preprocessing step checks the complete data for any portion of data missed or not. If any data is missing in the whole data set, then missed data is filled by 0 or null according to the data type. In case of data in unreadable format, it will be replaced by 0 or null.

Once the data is preprocessed, generally, it will be normalized by arranging the whole data set in such a manner that the will be arranged in ascending order or in descending order. But in this paper, the proposed approach the whole preprocessed data is divided into small size by applying windowing method by representing a window size. According to the window size the entire data set is changed into small size of sub-data. This sub data is compared among the data sets using Euclidean distance formula and gather similar data into groups. Each group of data is grouped due to the similarities.

One more methodology for group the data is by converting the data into objects. The data object for uncertain data is obtained by applying random sampling method.

The complete set of data stream DS is divided into two equal portions like DS1 and DS2. Select a query point q from the DS initially finds a data d1 from DS1 and d2 from DS2 where obtained by Equation [1].

$$d = dist\left(q, data\left(DS_i\right)\right) \quad (1)$$

The dist function is finding the similarity distance on the features of the data, where the dist may be a Euclidian distance or Manhattan, Mahalanobis. The results of the distance function are compared with a constant value □ and classify the points into groups. In the same way complete DS is classified into groups according the feature similarity. Instead of taking the whole data, divide the data into sub portions for fast and accurate, easy classification using data level pruning and object level pruning. For object level pruning, the data is converted into objects by taking random samples in range. The stage by stage pruning data level, object level is depicted clearly in Figure 2 and Figure 3 respectively.

3.1 Data Level Pruning

Assume the data is arranged in a square format for dividing the data into equal sub divisions. The complete data is divided into M x N matrix form. Where, each row and column is divided according to the window size [w]. Each window data is having a series of data as the time data. The square format data is divided into rows and columns and declared as SW [1,1], SW [1,2] … SW [1,N] ….. SW [1,1], SW [2,1], SW [3,1] ……, SW [M,1] until SW [M,N]. The same process is applied for the next data set also.

And the data SW1 [1,1] from first data set DS1 is compared with SW2 [1,1] in the second data set DS2. After completing the comparison, the similarity data are grouped as pair which satisfies the inequality constraints Equation [5] and Equation [9].

The functionality of the data level pruning and the object level pruning of the proposed approach is depicted in Figure 1and Figure 2 respectively. Since the data size is huge in size, in the data level pruning the overall data is divided into multiple sub sets. The complete data DS is divided into DS1 and DS2.

In existing kNN query processing, the search bound is determined by the farthest data point in the result set, i.e., the Kth nearest neighbor NN@k and the search bound can be described as a circle centered at query point q with radius of:

$$\mathrm{Dist}(\mathrm{q},\ \mathrm{NN@k})$$

We use □ to represent such a bound. Similarly but more roughly, a GNN query can be regarded as the equivalent of a corresponding range query, i.e., for the Kth distance value.

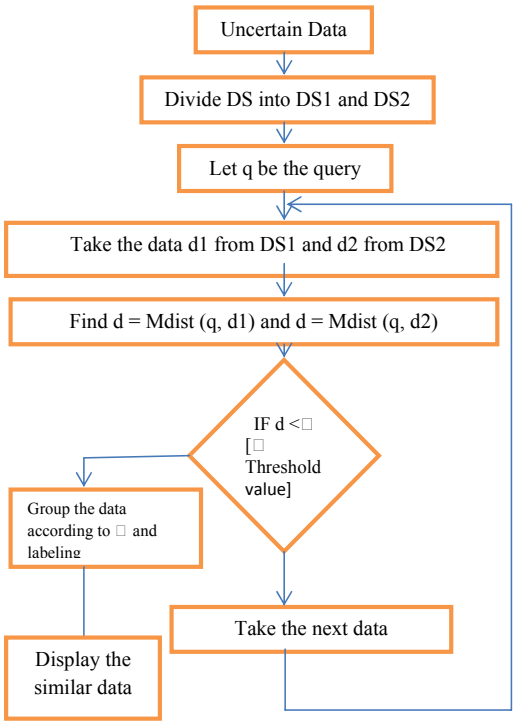

Figure 1. Proposed system model.

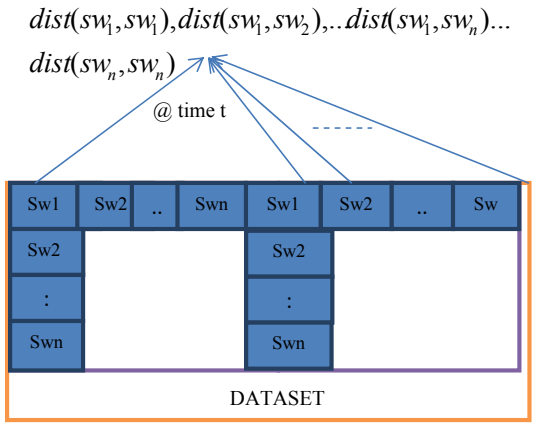

Figure 2. Data level pruning.

$$\varepsilon_k = \max\{dist(p,Q) \mid p \varepsilon GNN_Q^K\}$$

Both queries return the same result set and retrieve all objects in P that have a distance from Q not greater than ε_k. In other words

$$GNN_Q^k = range_Q(\varepsilon_k) = \{p_i \varepsilon P \mid dist(p_i,Q) \le \varepsilon_k,\ 1 \le i \le k\}$$

However, in a GNN query, it is difficult to determine and describe the search bound because of the number of

query points and their arbitrary distribution. Nevertheless, we can still get some inspiration from the search bound of a *k*NN query.

DS1 has split into K number of sub windows and the same manner DS2 has split into K number of sub windows. The first element of the first window from DS1 is compared with the second element of the first window and third element of the second window so on.

Once the first window of the DS2 is over, the first window of the DS1 is compared with the second window of the DS2, third window of the DS2 and so on. The same manner all the windows from DS1 is compared with the all the windows with the DS2. The comparison is finding the similarity distance between the data and it can be obtained by:

$RS = mdist\left(q, D0\right) \le \varepsilon$ Where ☐ is the threshold value.

RS= result set retrieved by the above constraint.

In this paper the two stage pruning method concentrates data level pruning as well as object level pruning. Where in the first stage is grouping on uncertain Data streams is data level. There are n numbers uncertain data streams available in a data pool, from that without loss of generality, full of our experiments we consider two uncertain data streams. A complete two uncertain data streams DataSet1 and DataSet2 are taken for our MYFRAME problem, where both data stream consists of a sequence of continuously occurring uncertain objects in different time interval, are denoted as:

$$\text{DataSet1} = \{x[1], x[2], \ldots x[t], \ldots\} \quad (1)$$

$$\text{DataSet2} = \{y[1], y[2], \ldots y[t], \ldots\} \quad (2)$$

Where x [i] or y [i] is a k-dimensional uncertain objects at the time interval i and t is the current time interval. According to group nearest neighbor, the objects should retrieve a close pairs of objects within a period. Thus a sub-window window concept is adapted for uncertain stream group operator. From Figure-2, a MYFRAME operator always considers the most recent SUBWIN uncertain data in stream, that is:

$$SUBWIN\left(Dataset1\right) = \begin{Bmatrix} x[t - SubWin+], \\ x[t - SubWin + 2], \ldots . x[t] \end{Bmatrix} \quad (3)$$

$$SUBWIN(DataSet\ 2) = \begin{Bmatrix} y[t - SubWin + 1], \\ y[t - SubWin + 2], \ldots . . y[t] \end{Bmatrix} \quad (4)$$

At the time intervals t, it can be say in other words, when a new certain object x [t+1] (y [t+1]) comes in at the next time interval (t+1), the new object x [t+1] (y [t+1]) is appended to DS1(DS2). In that particular time the old object x [t-SubWin+1] (y [t-SubWin+1]) expires and is evicted from the memory. Thus, MYFRAME at time interval (t+1) is conducted on a new sub-window window {x [t-SubWin+2], ……x [t+1]} (y [t-w+2],….,y [t+1]}) of size SubWin.

For Grouping the uncertain Data Streams, the two data streams DS1 and DS2, a distance threshold value ε and a probabilistic threshold $\alpha \in [0, 1]$, a group on uncertain data streams continuously monitors pairs of uncertain objects x [i] and y [i] within compartment windows SUBWIN (DS1) and SUBWIN (DS2), respectively, of size SubWin at the current period of clock interval t, and it is:

$$\Pr\left\{dist\left(x[i], y[i]\right) \le \varepsilon \ge \alpha\right\} \quad (5)$$

Holds, where t-SubWin+1 ≤ i, j ≤ t, and dist (., .) is a Euclidean distance function between two objects. To perform a MYFRAME Equation (5), users need to register two parameters, distance threshold ☐ and probabilistic threshold α. Since each uncertain object at a time stamp consists of R samples, the grouping probability.

$$P\,|\,r\left\{dist(x[i], y[i] \le \varepsilon\right\}$$

In Inequality (5) can be rewritten via samples as:

$$\Pr\left\{dist\left(x[i], y[i]\right) \le \varepsilon\right\} = \sum_{k1=1}^{R} \sum_{k2=1}^{R} \begin{Bmatrix} xk1[i].p.yk2[j].p, \\ 0\ otherwise \end{Bmatrix}$$

$$\text{if } dist\left(xk1[i], yk2[j] \le \varepsilon\right) \quad (6)$$

Note that one straight forward method to directly perform MYFRAME over sub-window windows is to follow the MYFRAME definition. That is, for every object pair <X[i],Y[i]> from sub-window windows SUBWIN (DataSet1) and SUBWIN (DATASet2) respectively, we compute the grouping probability that X [i] is within ☐ distance from Y [i] (via samples) based on (6). If the resulting probability is greater than or equal to probabilistic threshold α, then this pair <X[i],Y[i]> is reported as the MYFRAME answer; otherwise, it is a false alarm and can be safely eliminating.

3.2 Object Level Pruning

Given a pair of uncertain objects X [t+1] and Y [j] and a distance threshold ❐, candidate pair <X[t+1],Y[j]> can be safely pruned if it holds that:

$$dist\left(Cx[t+1], Cy[j]\right) - rX[t+1] - rY[j] > \varepsilon \qquad (8)$$

Proof: From the Figure 3 spontaneously, LHS of the inequality (8) corresponds to the minimumpossible sample distance between objects X [t+1] and Y [j]. If this minimum distance is greater than the distance threshold. ❐, then inequality (5) in the MYFRAME definition will never hold [because $\alpha > 0$] and thus we discard this object pair.

To improve the efficient than the object level pruning one more inequality constraint is applied for the data object is:

$$dist\left(CX[t+1], CY[j]\right) \le \varepsilon + rX[t+1] + r\max(DS2) \qquad (9)$$

Then, instead of exhaustive computation, only those uncertain objects in grid cells satisfying Equation (9) are needed to be accessed, where rX [t+1] is the radius of object X [t+1] and rmax (DS2) is defined as the maximum radius among all objects in component windows SUBWIN (DS2).

```
Data_Level_Pruning_Algorithm( ).
{
Data set DS1, DS2 contains set of all data.
DS1 = {D1,D2,D3,…..DN}.
DS2 = {D1,D2,D3,……DM}.
Enter the value of window W.
For I = 1 to N step W
        For J = 1 to M step W
                Score[i] = ED(DS1(I,J),DS2(I,J);
End j
End i
For i=1 to N
For J=1 to M
If Score[i] satisfies [P|r{dist(x[i],y[i]) ≤ ❐] then pair1 =
 pair1 = DS1[I,j],DS2[I,j]
Next j
Next i
}
```

Object level pruning also uses the sliding window concepts. Randomly choose an uncertain object X [t+1] from the SW [1,1] and a number of uncertain objects Y [j] where [t-w +2 ≤ j ≤ t+1] from the sliding window SW2 [1, 1]. In this paper, object level pruning method

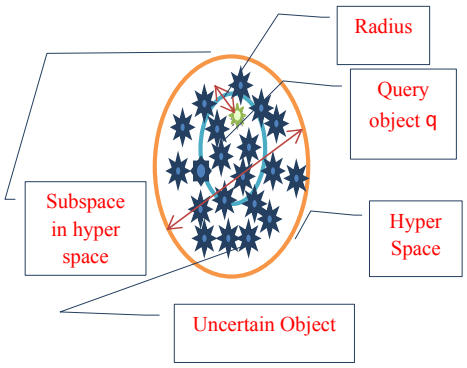

Figure 3. Object level pruning.

eliminate the candidate pairs such that the connecting probability Pr{dist(X[t+1],Y[j]) ≤ ❐b = 0}. To prune the pairs <x[t+1],Y[j]> such that the objects X [t+1] and Y [j] should have distance to each other greater than the distance threshold ❐.

```
Object_level_pruning_Algorithm ( )
{
*Obtain uncertain object X[t1] and Y[t1] from uncertain
data streams DS1 and DS2 respectively.
*And add new object (X[t+1],Y[t1]) to and obliterate the
expired object.
→ X[t - cw +1](Y[t- cw +1]) from CW(DS1) (CW(DS2)).
→Invoke the procedure getdata_pair() to find the data objects
Y[j] (X[i]) in CW(DS2)(CW(DS1)) such that inequality
(5) holds for pair <X[t+1],Y[j]> (<X[i],Y[t+1]>).
→ Insert the data pair <X[t1],Y[j]> (<X[i],Y[t1]>) into the
result RS and obliterate the expired pair in RS.
}
```

Once the data set is preprocessed and normalized then a query object will be fired in the data set. Then the query object will be matched with the RS data pair and retrieve the accurate or relevant data object from the pairs. If the data is not available in the RS, then it will display "Data not available".

4. Simulation Settings

The data is occupied from time series data of United States of America. The time series data is about trading information at every time interval for various cities. The total number of data taken for simulation is one year data size having 1781 records. The wide-ranging data is grouped conferring to the time. The similar data can be

obtained from each city, in particular date at particular time. The algorithm Data level pruning and Object level pruning are implemented in MATLAB-2012a software and the performance of the proposed approach is evaluated.

5. Results and Discussion

The input data is read and preprocessed for effective simulation in terms of readability. The data set taken is huge in size. Data is taken from United States of America based Trading company. This data is divided into region wise and time wise. One year data is taken as DS and it is divided into two equal portions DS1 and DS2 according to the proposed approach.

The original data DS is divided into DS1 and DS2 for easy process. The DS1 and DS2 size is 8000 each and shown in Figure 4. After splitting the DS as DS1 and DS2, the data will be preprocessed and normalized. Pre-processing removes the zeros and replace the unreadable character into 1 or Null according to the data type.

Once the data preprocessed then the data will be normalized by arranging the data in ascending or may be in descending order. To make the data mining easy, make the data model is clustered and classified one, the data will be normalized. The normalized data is shown in Figure 5.

Since the data size is big [in future also] and make the pruning process easier the windowing system is applied and split DS1, DS2 into W (DS1) and W (DS2). The size of W is a constant and it can be assigned by the developer due to the data size and number of iteration to be carried out for pruning. The W (DS1) and W (DS2) is shown in Figure 6.

The normalized data is pruned using data level pruning and object level pruning then the pair wise comparison is made among the data in data sets. The similarity data is obtained by collecting the data pairs satisfying the inequality constrained shown in Equation [5] and Equation [9] and the result is shown in Figure 7.

The object level pruning can be applied after converting the data into objects. To convert the data into subspace objects random sampling based nearest data generated within a radius. The Figure 8 shows the random sampled data created for DS1 and DS2.

The Random sampled data is normalized by arranging the data in ascending order. Similarity computation among data becomes much easier if the data is in a normalized form. The normalized hyperspace object is shown in Figure 9.

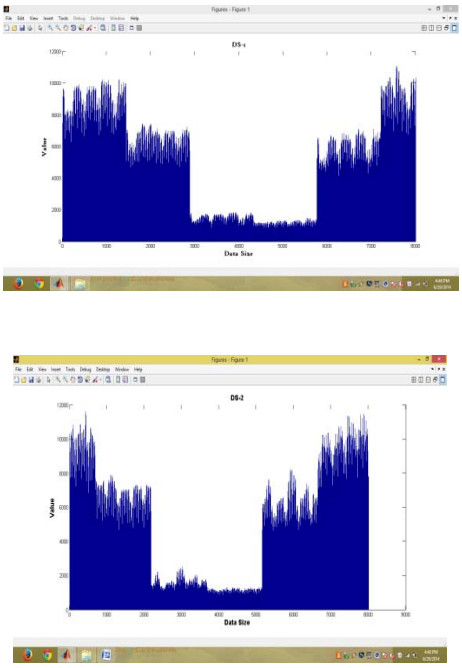

Figure 4. Original data divided into DS1 and DS2.

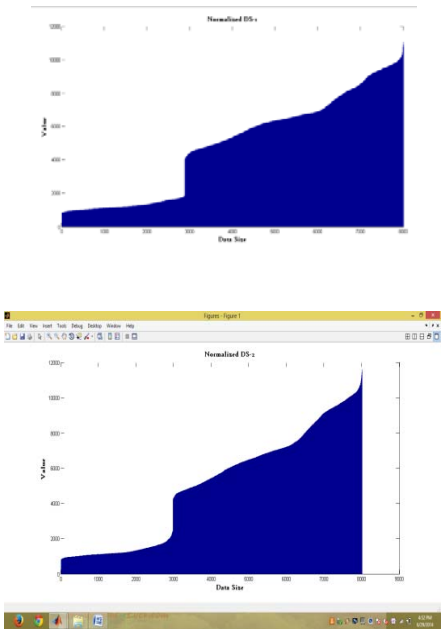

Figure 5. Normalized data for DS1 and DS2.

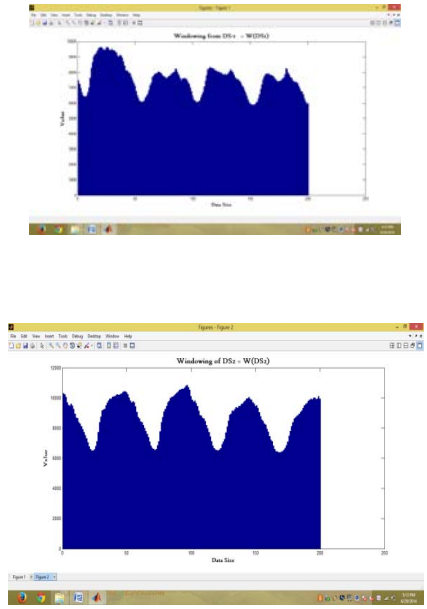

Figure 6. Window datasets W (DS1) and W (DS2).

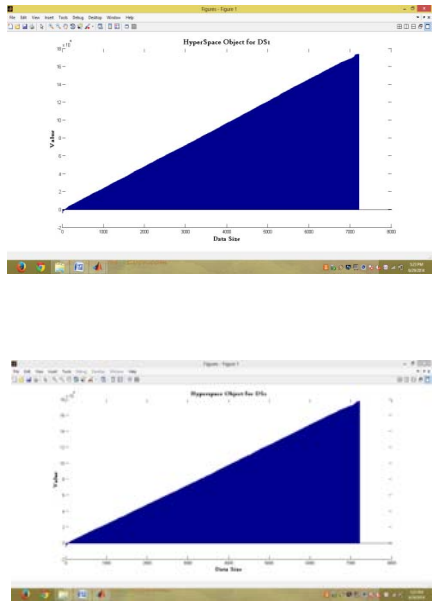

Figure 9. Hyper space objects for DS1 and DS2.

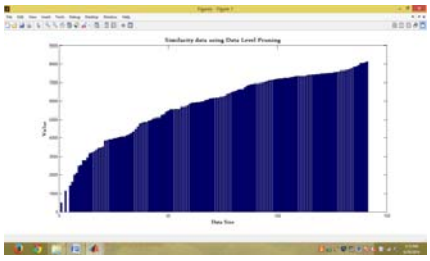

Figure 7. Similarity Data by Data level pruning.

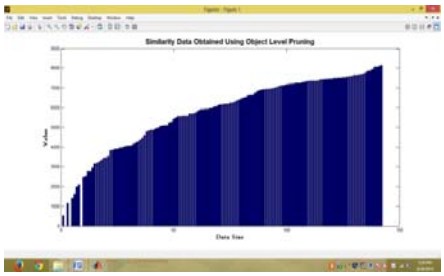

Figure 10. Similarity data obtained using object level pruning.

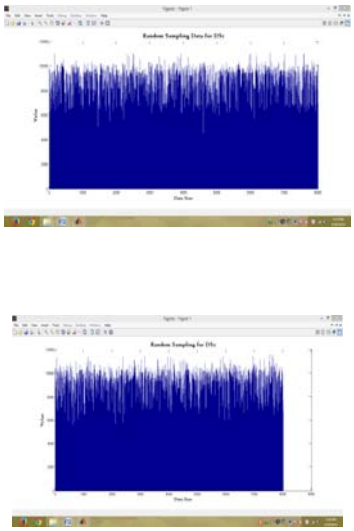

Figure 8. Random sampled data for DS1 and DS2.

The similarity data collected among the data set1 and Dataset2 are shown in Figure 10 is the result of two level pruned similarity data. To evaluate the performance of the stage by stage pruning method, the result of this paper is compared with the existing approach's result in terms of precision and recall.

The Table 1 shows the precision and recall of different approaches by varying the size on NHL and Time Series data sets.

On NHL data set, we can see that, when the data size is relatively small, e.g. 16000 the performance of Brute-Force, TSC and SPRIT is comparable with our SBSP method.

Table 1. Precision/Recall by varying the data size

Method	Data Size	Recall
TSC	3200	1.00
Brute-Force		1.00
SPIRIT		0.41
SBSP		0.98
TSC	6400	1.0
Brute-Force		1.0
SPIRIT		0.3
SBSP		1.0
TSC	9600	1.00
Brute-Force		1.00
SPIRIT		0.38
SBSP		1.00
TSC	12800	0.98
Brute-Force		1.00
SPIRIT		0.37
SBSP		0.88
TSC	16000	0.97
Brute-Force		0.96
SPIRIT		0.36
SBSP		0.82
TSC	1440	0.79
Brute-Force		0.52
SPIRIT		0.59
SBSP		0.77
TSC	10368	0.37
Brute-Force		0.35
SPIRIT		0.29
SBSP		1.00

6. Conclusion

In order to fetch the similarity data as a group it is proposed two stage pruning which has data level and object level pruning. This approach is applied nowadays in real time applications such as object detection, video retrieval, people detection and tracking, earth quake monitoring etc.

7. References

1. Gayathri S, Mary Metilda M, Sanjai Babu S. A Shared Nearest Neighbour Density based clustering approach on a proclus method to cluster high dimensional data. Indian Journal of Science and Technology. 2015 Sep; 8(22):1–6. IPL0276.
2. Sasirekha D, Punitha A. A comprehensive analysis on associative classification in medical datasets. Indian Journal of Science and Technology. 2015 Dec; 8(33): 1–9. Doi: 10.17485/ijst/2015/v8i33/80081.
3. Basri R, Jacobs D. Lambertian reflectance and linear subspaces. IEEE Trans Pattern Analysis and Machine Intelligence. 2003 Feb; 25(2):218–33.
4. Ramamoorthi R, Hanrahan P. On the relationship between radiance and irradiance: Determining the illumination from images of convex lambertian object. J Optical Soc of Am. 2001 Oct; 18(10):2448–59.
5. Ullman S, Basri R. Recognition by linear combinations of models. IEEE Trans Pattern Analysis and Machine Intelligence. 1991 Oct; 13(10):992–1006.
6. Tomasi C, Kanade T. Shape and motion from image streams under orthography: A factorization method. Int'l J Computer Vision. 1992 Nov; 9(2):137–54.
7. Simard P, LeCun Y, Denker J, Victorri B. Transformation invariance in pattern recognition - Tangent distance and tangent propagation. Neural Networks: Tricks of the Trade. Springer. 1998; 1524:239–74.
8. Brand ME. Morphable 3D Models from Video. Proc IEEE Conf Computer Vision and Pattern Recognition. 2001; 2:456–63.
9. Torresani L,Yang D, Alexander G, Bregler C. Tracking and modeling non-rigid objects with rank constraints. Proc IEEE Conf Computer Vision and Pattern Recognition. 2001; 1:493–500.
10. Fitzgibbon A, Zisserman A. Joint manifold distance: A new approach to appearance based clustering. Proc IEEE Conf Computer Vision and Pattern Recognition. 2003 Jun; 1:26–33.
11. Zhang H, Berg AC, Maire M, Malik J. SVM-KNN: Discriminative Nearest Neighbor Classification for visual category recognition. Proc IEEE Conf Computer Vision and Pattern Recognition. 2006; 2:2126–36.
12. Atick JJ, Griffin PA, Redlich AN. Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images. Neural Computation. 1996 Aug; 8(6):1321–40.
13. Blanz V, Vetter T. Face recognition based on Fitting a 3D morphable model. IEEE Trans Pattern Analysis and Machine Intelligence. 2003 Sep; 25(9):1063–74.
14. Arya S, Mount D, Netanyahu N, Silverman R, Wu A. An optimal Algorithm for Approximate Nearest Neighbor

searching in fixed dimensions. J ACM. 1998 Nov; 45(6):891–923. Available from: www.cs.umd.edu/mount /ANN /
15. Datar M, Immorlica N, Indyk P, Mirrokni V. Locality-sensitive hashing scheme based on P-stable distributions. Proc 20th Ann Symposium Computational Geometry; 2004. p. 253–62.
16. Indyk P, Motwani R. Approximate Nearest Neighbors: Towards removing the curse of dimensionality. Proc 30th Ann ACM Symp. Theory of Computing; 1998. p. 604–13.
17. Liu T, Moore AW, Gray A,Yang K. An investigation of practical Approximate Nearest Neighbor algorithms. Proc Neural Information Processing Systems; 2004. p. 825–32.
18. Gaede V, Gnther O. Multidimensional access methods. ACM Computing Surveys CSUR). 1998 Jun; 30(2):170–231.
19. Ross KA, Sitzmann I, Stuckey PJ. Cost- based unbalanced R-trees. IEEE International Conference on Scientific and Statistical Database Management (SSDBM); 2001. p. 203–12.
20. Theodoridis Y, Stefanakis E, Sellis TK. Efficient Cost models for spatial queries using R-trees. IEEE TKDE. 2000 Jan/Feb; 12(1):19–32.
21. Burkhard WA, Keller RM. Some approaches to best-match file searching. Communications of the ACM. 1973 Apr; 16(4):230–6.
22. Keogh E, Chakrabarti K, Pazzani, Mehrotra S. Dimensionality reduction for fast similarity search in large time series databases. J Knowledge and Information Systems. 2001 Aug; 3(3):263–86.
23. Kahveci T, Singh AK. Variable length queries for time series data. Proceedings IEEE International Conference on Data Engineering. (ICDE); 2001. p. 273–82.
24. Wu Y, Agrawal, El Abbadi A. A comparison of DFT and DWT based similarity search in time-series databases. Proc Ninth Int'l Conf Information and Knowledge Management (CIKM); 2000. p. 488–95.
25. Keogh E, Chu S, Hart D, Pazzani M. Segmenting time series: A survey and novel approach. Data Mining in Time Series Databases. World Scientific; 2004.
26. Bashir FI, Khokhar AA, Schonfeld D. Real-time motion trajectory-based indexing and retrieval of video sequences. IEEE Trans Multimedia. 2007 Jan; 9(1):58–65.
27. Le TL, Boucher A, Thonnat M. Subtrajectory-based video indexing and retrieval. Proc13th Int'l Multi Media Modeling Conf (MMM). 2007; 4351:418–27.