THE ROLE OF HUBNESS IN CLUSTERING HIGH-DIMENSIONAL DATA

N.RAJENDER
Asst.professor,Dept of CSE
Nalla Narasimha Reddy Education Society's Group of Institutions, Korremula X Road, Ghatkesar Mandal, RangaReddy District, Chowdariguda, Telangana 500088
Email:rajender.n@nnrg.edu.in

V.RAJU
Asst.prof, Dept of Cse
Nalla Narasimha Reddy Education Society's Group of Institutions, Korremula X Road, Ghatkesar Mandal, Ranga Reddy District, Chowdariguda, Telangana 500088
Email:raju.v@nnrg.edu.in

Abstract— *High-dimensional data arise naturally in many domains, and have regularly presented a great challenge for traditional data mining techniques, both in terms of effectiveness and efficiency. Clustering becomes difficult due to the increasing sparsity of such data, as well as the increasing difficulty in distinguishing distances between data points. Instead of attempting to avoid the curse of dimensionality by observing a lower dimensional feature subspace, we embrace dimensionality by taking advantage of inherently high-dimensional phenomena. Furthermore SHA-1 algorithm is applied to ensure that there is no duplicate data is inserted into the system. By using this SHA-1 algorithm execution time is also reduced as the number of tuples to be searched in the database is reduced as the redundancy is removed.*

Keywords—component; formatting; style; styling; insert (key words)

I. INTRODUCTION

This project aims at building a system that prevents duplicate data in the database when using the property of hubness in clustering high-dimensional data. This is achieved through the implementation of SHA-1 algorithm in the publisher module.SHA-1 algorithm generates a 20-byte hash value known as message digest. Using this hash value duplicate data is avoided in the database.

II. EXISTING SYSTEM

In existing system, there is no mechanism to prevent the publisher from entering duplicate data into the system. As the current system supports data redundancy it will lead to unnecessary memory is usage, which in turn will cause longer execution type.The role of hubness in clustering high-dimensional dataHigh-dimensional data arise naturally in many domains, and have regularly presented a great challenge for

traditional data mining techniques, both in terms of effectiveness and efficiency. Clustering becomes difficult due to the increasing sparsity of such data, as well as the increasing difficulty in distinguishing distances between data points. In this they proposed a concept called hubness, i.e., the tendency of high-dimensional data to contain points (hubs) that frequently occur in knearest-neighbor lists of other points, can be successfully exploited in clustering.Prepare Your Paper Before Styling

A. *Curse of dimensionality*

Curse of dimensionality is due to two pervasive effects: the empty space phenomenon and concentration of distances. The former refers to the fact that all high-dimensional data sets tend to be sparse, because the number of points required to represent any distribution grows exponentially with the number of dimensions. This leads to bad density estimates for high-dimensional data, causing difficulties for density-based approaches. The latter is a somewhat counterintuitive property of high-dimensional data representations, where all distances between data points tend to become harder to distinguish as dimensionality increases, which can cause problems with distance - based algorithms. The difficulties in dealing with high-dimensional data are omnipresent and abundant. However, not all phenomena that arise are necessarily detrimental to clustering techniques. hubness, which is the tendency of some data points in high-dimensional data sets to occur much more frequently in k-nearestneighbor lists of other points than the rest of the points from the set, can in fact be used for clustering.

B. *The hubness phenomenon*

Hubness is an aspect of the curse of dimensionality pertaining to nearest neighbours which has only recently come to attention, unlike the much discussed distance concentration phenomenon. Let D C IRd be a set of data points and let Nk(x) denote the number of k-occurrences of point x € D, i.e., the number of times x occurs in k-nearest neighbour lists of other points from D. As the dimensionality of data increases, the distribution of k-occurrences becomes considerably skewed. As a consequence, some data points, which we will refer to as hubs, are included in many more k-nearest-neighbour lists than other points. In the rest of the text, we will refer to the number of k-occurrences of point x € D as its hubness score. It has been shown that hubness, as a phenomenon, appears in high-dimensional data as an inherent property of high dimensionality, and is not an artefact of finite samples nor a peculiarity of some

specific data sets. Naturally, the exact degree of hubness may still vary and is not uniquely determined by dimensionality.

C. *Emergence of hubs*

The concentration of distances enables one to view unimodal high-dimensional data as lying approximately on a hyper sphere centred at the data distribution mean. However, the variance of distances to the mean remains non negligible for any finite number of dimensions, which implies that some of the points still end up being closer to the data mean than other points. It is well known that points closer to the mean tend to be closer (on average) to all other points, for any observed dimensionality. In high-dimensional data, this tendency is amplified. Such points will have a higher probability of being included in k-nearest-neighbour lists of other points in the data set, which increases their influence, and they emerge as neighbour-hubs. It was established that hubs also exist in clustered (multimodal) data, tending to be situated in the proximity of cluster centres. In addition, the degree of hubness does not depend on the embedding dimensionality, but rather on the intrinsic data dimensionality, which is viewed as the minimal number of variables needed to account for all pair wise distances in the data. Generally, the hubness phenomenon is relevant to (intrinsically) high-dimensional data regardless of the distance or similarity measure employed. Its existence was verified for euclidean and Manhattan distances, lp distances with p>2, fractional distances (lp with rational p € (0;1), Bray-Curtis, normalized euclidean, and Canberra distances, cosine similarity, and the dynamic time warping distance for time series. Hubs are points x having Nk(x) more than two standard deviations higher than the expected value k (in other words, significantly above average).

D. *Hub-based clustering*

If hubness is viewed as a kind of local centrality measure, it may be possible to use hubness for clustering in various ways. To test this hypothesis, we opted for an approach that allows observations about the quality of resulting clustering configurations to be related directly to the property of hubness, instead of being a consequence of some other attribute of the clustering algorithm. Since it is expected of hubs to be located near the centres of compact subclusters in high-dimensional data, a natural way to test the feasibility of using them to approximate these centres is to compare the hub-based approach with some centroid-based technique. For this reason, the considered algorithms are made to resemble K-means, by being iterative approaches for defining clusters around

separated high-hubness data elements. Centroids and medoids in K-means iterations tend to converge to locations close to high-hubness points, which implies that using hubs instead of either of these could actually speed up the convergence of the algorithms, leading straight to the promising regions in the data space. To illustrate this point, consider the simple example shown in Fig. 4, which mimics in two dimensions what normally happens in multidimensional data, and suggests that not only might taking hubs as centers in following iterations provide quicker convergence, but that it also might prove helpful in finding the best end configuration. Centroids depend on all current cluster elements, while hubs depend mostly on their neighbouring elements and, therefore, carry localized centrality information. We will consider two types of hubness below, namely global hubness and local hubness. We define local hubness as a restriction of global hubness on any given cluster, considered in the context of the current algorithm iteration. Hence, the local hubness score represents the number of k-occurrences of a point in k-NN lists of elements within the same cluster. The fact that hubs emerge close to centers of dense subregions might suggest some sort of a relationship between hubness and the density estimate at the observed data point.

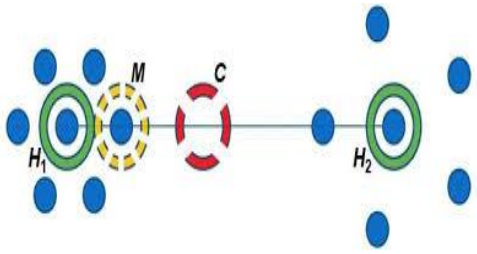

Figure 1: Elements of hubness

III. PROPOSED SYSTEM

This project aims at building a system that prevents duplicate data in the database when using the property of hubness in clustering high-dimensional data. This is achieved through the implementation of SHA-1 algorithm in the publisher module. This new system discourages data redundancy.

A. *Several Links to User*

- Register: For the user to register.
- Sign-in: For the user to sign-in to the system
- New publisher register: For the publisher to register.
- New publisher sign-in: For the publisher to sign-in to the system

B. *Advantages*

- The performance is increased due to well-designed database.
- Data redundancy is prevented.
- Time saving in report generation.

Data flow diagram is a structure analysis tool that is used for graphical representation of Data processes through any organization.

The data flow approach emphasis on the logic underlying the system, by using combination of only 4 symbols. It follows a top down approach. A full description of a system actually consists of set of DFD s, which comprises of various levels. And initial over view model is exploded lower level diagrams that show additional feature of the system.

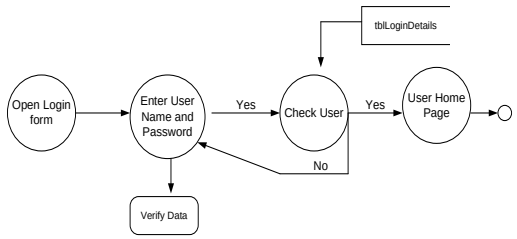

Figure. 2. Data flow of this module

IV. IMPLEMENTATION

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analyzed and specified the software design involves three technical activities design, coding, generation and testing that are required to build and verify the software. The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system. Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

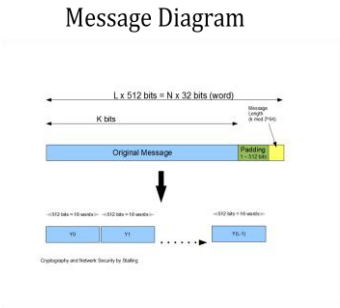

Figure. 3. Message diagram for SHA-1

- Step 1: Append Padding Bits….

  Message is "padded" with a 1 and as many 0's as necessary to bring the message length to 64 bits fewer than an even multiple of 512.

- Step 2: Append Length....

  64 bits are appended to the end of the padded message. These bits hold the binary format of 64 bits indicating

the length of the original message.

- Step 3: Prepare Processing Functions….

  SHA1 requires 80 processing functions defined as:

  f(t;B,C,D) = (B AND C) OR ((NOT B) AND D) (0 <= t <= 19)

  f(t;B,C,D) = B XOR C XOR D (20 <= t <= 39)

  f(t;B,C,D) = (B AND C) OR (B AND D) OR (C AND D) (40 <= t <=59)

  f(t;B,C,D) = B XOR C XOR D (60 <= t <= 79)

- Step 4: Prepare Processing Constants....

  SHA1 requires 80 processing constant words defined as:

  K(t) = 0x5A827999 (0 <= t <= 19)

  K(t) = 0x6ED9EBA1 (20 <= t <= 39)

  K(t) = 0x8F1BBCDC (40 <= t <= 59)

  K(t) = 0xCA62C1D6 (60 <= t <= 79)

- Step 5: Initialize Buffers….

  SHA1 requires 160 bits or 5 buffers of words (32 bits):

  H0 = 0x67452301

  H1 = 0xEFCDAB89

  H2 = 0x98BADCFE

  H3 = 0x10325476

  H4 = 0xC3D2E1F0

- Step 6: Processing Message in 512-bit blocks (L blocks in total message)….

  This is the main task of SHA1 algorithm which loops through the padded and appended message in 512-bit blocks.

  Input and predefined functions:

  M[1, 2, ..., L]: Blocks of the padded and appended message f(0;B,C,D), f(1,B,C,D), ...,

f(79,B,C,D): 80 Processing Functions K(0), K(1), ..., K(79): 80 Processing Constant Words

H0, H1, H2, H3, H4, H5: 5 Word buffers with initial values

- Step 7: Pseudo Code….

For loop on k = 1 to L

(W(0),W(1),...,W(15)) = M[k] */* Divide M[k] into 16 words */*

For t = 16 to 79 do:

W(t) = (W(t-3) XOR W(t-8) XOR W(t-14) XOR W(t-16)) <<< 1

A = H0, B = H1, C = H2, D = H3, E = H4

For t = 0 to 79 do:

TEMP = A<<<5 + f(t;B,C,D) + E + W(t) + K(t)

E = D, D = C,

C = B<<<30,

B = A, A = TEMP

End of for loop

H0 = H0 + A, H1 = H1 + B, H2 = H2 + C, H3 = H3 + D, H4 = H4 + E

End of for loop

OUTPUT:

H0, H1, H2, H3, H4, H5: Word buffers with final message digest

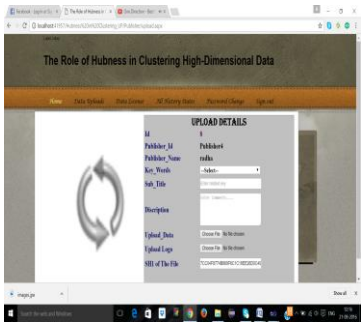

Figure. 4 SHA-1 Generation

V. CONCLUSION

We have proposed application of a redundancy removal technique to the implementation of role hubness in clustering high-dimensional data. For this purpose SHA-1 algorithm is incorporated in the system. Using this algorithm it is made sure, that no duplicate data is entered in to the database. Which will reduce the space utilized in the database. It will also reduce the time taken to generate the required results.

REFERENCES

[1] G. Eason, B. Noble, and I.N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc.

London, vol. A247, pp. 529-551, April 1955. (*references*)

[2] 1] J. Han and M. Kamber, Data Mining: Concepts and Techniques, second ed. Morgan Kaufmann, 2006.

[3] [2] C.C. Aggarwal and P.S. Yu, "Finding Generalized Projected Clusters in High Dimensional Spaces," Proc. 26th ACM SIGMOD Int'l Conf. Management of Data, pp. 70-81, 2000.

[4] K. Kailing, H.-P. Kriegel, P. Kro ¨ger, and S. Wanka, "Ranking Interesting Subspaces for Clustering High Dimensional Data," Proc. Seventh European Conf. Principles and Practice of Knowledge Discovery in Databases (PKDD), pp. 241-252, 2003.

[5] K. Kailing, H.-P. Kriegel, and P. Kro ¨ger, "Density-Connected Subspace Clustering for High-Dimensional Data," Proc. Fourth SIAM Int'l Conf. Data Mining (SDM), pp. 246-257, 2004.

[6] E. Mu ¨ller, S. Gu ¨nnemann, I. Assent, and T. Seidl, "Evaluating Clustering in Subspace Projections of High Dimensional Data," Proc. VLDB Endowment, vol. 2, pp. 1270-1281, 2009.

[7] C.C. Aggarwal, A. Hinneburg, and D.A. Keim, "On the Surprising Behavior of Distance Metrics in High Dimensional Spaces," Proc. Eighth Int'l Conf. Database Theory (ICDT), pp. 420-434, 2001.

[8] D. Franc¸ois, V. Wertz, and M. Verleysen, "The Concentration of Fractional Distances," IEEE Trans. Knowledge and Data Eng., vol. 19, no. 7, pp. 873-886, July 2007.

[9] R.J. Durrant and A. Kaba ´n, "When Is 'Nearest Neighbour' Meaningful: A Converse Theorem and Implications," J. Complexity, vol. 25, no. 4, pp. 385-397, 2009.

[10] A. Kaba ´n, "Non-Parametric Detection of Meaningless Distances in High Dimensional Data," Statistics and Computing, vol. 22, no. 2, pp. 375-385, 2012.

[11] E. Agirre, D. Martı ´nez, O.L. de Lacalle, and A. Soroa, "Two Graph-Based Algorithms for State-of-the-Art WSD," Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), pp. 585-593, 2006.

N.Rajender, Received M.Tech in Software Engineering from JNTU Hyderabad and Had 10 years of experience in teaching.

V.Raju, Received M.Tech in Software Engineering from JNTU-Hyderabad and Had 6 years of experience in teaching