International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2020 IJSRCSEIT | Volume 6 | Issue 1 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT195671

# Survey on Clustering High-Dimensional data using Hubness

**Miss. Archana Chaudahri[1], Mr. Nilesh Vani[2]**
[1]ME Scholar, Computer Engineering, GF's GCOE, Jalgaon, Jalgoan, Maharashtra, India
[2]Assistant Professor, Computer Engineering, GF's GCOE, Jalgon, Jalgoan, Maharashtra, India

## ABSTRACT

Most data of interest today in data-mining applications is complex and is usually represented by many different features. Such high-dimensional data is by its very nature often quite difficult to handle by conventional machine-learning algorithms. This is considered to be an aspect of the well known curse of dimensionality. Consequently, high-dimensional data needs to be processed with care, which is why the design of machine-learning algorithms needs to take these factors into account. Furthermore, it was observed that some of the arising high-dimensional properties could in fact be exploited in improving overall algorithm design. One such phenomenon, related to nearest-neighbor learning methods, is known as hubness and refers to the emergence of very influential nodes (hubs) in k-nearest neighbor graphs. A crisp weighted voting scheme for the k-nearest neighbor classifier has recently been proposed which exploits this notion.

**Keywords :** Hubness, Clustering Methods, Datamining Techniques

## I. INTRODUCTION

In recent years, high dimensional search and retrieval have become very well studied problems because of the increased importance of data mining applications. Over the years, various clustering algorithms have been proposed, which can be roughly divided into four groups: partitional, hierarchical, density based, and subspace algorithms. Algorithms from the fourth group search for clusters in some lower dimensional projection of the original data, and have been generally preferred when dealing with data that are high dimensional [2], [3], [4], [5]. The motivation for this preference lies in the observation that having more dimensions usually leads to the so-called curse of dimensionality, where the performance of many standard machine-learning algorithms becomes impaired.

Typically, most real applications which require the use of such techniques comprise very high dimensional data. For such applications, the curse of high dimensionality tends to be a major obstacle in the development of data mining techniques in several ways. For example, the performance of similarity indexing structures in high dimensions degrades rapidly, so that each query requires the access of almost all the data[6].

Spectral clustering has attracted increasing attention due to its superior performance on some challenging clustering tasks [2]. Because of the capacity of partitioning data with complexed structures, spectral clustering has been widely applied in many research fields, including image segmentation [3], [4], circuit layout [5], video retrieval [6] and bioinformatics [7]. However, when the number of data points becomes large, the applicability of spectral clustering is limited. The general spectral clustering method consists of two

main steps: (1) constructing a similarity matrix; (2) calculating the eigendecomposition of the corresponding Laplacian matrix. For a dataset with n data points, the two steps take computational complexities of $O(n^2)$ and $O(n^3)$ respectively, which is an unbearable burden for large-scale clustering problems. Many accelerated spectral clustering methods have been proposed to overcome the scalability problem by using sampling techniques. Fowlkes et al. [12] apply the Nystrom method to reduce the high complexity in the eigen-decomposition step. By randomly selecting a small subset of samples, a similarity sub-matrix is constructed based on these samples. The calculated eigenvectors based on the similarity sub-matrix are used to estimate an approximation of the eigenvectors of the original similarity matrix. Li et al. [13] further accelerate the Nystr¨om approximation based spectral clustering by using the randomized low-rank matrix approximation algorithms.

Instead of reducing the complexity in the eigendecomposition step of spectral clustering, several methods reduce the data size beforehand to construct the similarity matrix. The K-means based approximate spectral clustering (KASP) method [14] applies K-means with a large cluster number p to find p center points. The general spectral clustering algorithm is then performed on the p cluster centers, with each data point being assigned to the same cluster as its nearest center. A similar method has been proposed by Shinnou and Sasaki [15], which removes the data points close to the p centers, and the general spectral clustering is performed on the remaining data points plus the p centers. The removed data points are finally assigned to the cluster as their nearest centers.

The recently described phenomenon of hubness has been marked as potentially highly detrimental. The term was coined after hubs, very frequent neighbor points which dominate among all the occurrences in the k-neighbor sets of inherently high-dimensional data. Most other points either never appear as neighbors or do so very rarely. They are referred to as anti-hubs. This property is usually of a geometric nature and does not reflect the semantics of the data, as discussed in the context of music retrieval. The researchers have noticed that some songs are very frequently being retrieved, but were unable to attribute these occurrences to any similarity observable by people. There is no easy way out, as demonstrated in [17], since dimensionality reduction techniques fail to eliminate the neighbor occurrence distribution skewness for any reasonable dimensionality of the projection space. The skewness decreases only when mapping to very low-dimensional spaces, where too much potentially relevant information is irretrievably lost. Therefore, hubness remains a phenomenon which needs to be taken into account when using nearest neighbor methods on high-dimensional data. Shared neighbor distances are sometimes used as secondary distance measures when dealing with highdimensional data, usually in clustering applications Similarity between points is defined as the number of shared neighbors in their k- neighbor sets, and distances between points are then usually defined in one of the several essentially equivalent ways, Shared neighbor distances have been mentioned as a potential cure for the curse of dimensionality. We have chosen to focus on using the shared neighbor distances in supervised learning, k-nearest neighbor (k-NN) clustering particular (where the neighbors are determined based on the secondary distances)[16][17]. Researcher have measured the hubness of the induced shared neighbor spaces and have shown that hubness-awarek-nearest neighbor classification leads to significant improvements over the basic k-NN even when using these secondary distances instead of the original underlying metrics. In other words, shared neighbor distances do not eliminate hubness, so they do not entirely overcome the curse of dimensionality. Hubness has an impact on the forming of the shared neighbor similarity scores, so we propose a new

hubness-aware method for calculating shared neighbor similarities/distances.

## II. LITERATURE SURVEY

Here are the survey of some of the clustering methods usually used in data mining.

1) **Distribution based methods :** It is a clustering model in which we will fit the data on the probability that how it may belong to the same distribution. The grouping done may be *normal or gaussian* . Gaussian distribution is more prominent where we have fixed number of distributions and all the upcoming data is fitted into it such that the distribution of data may get maximized. This result in grouping which is shown in figure 1

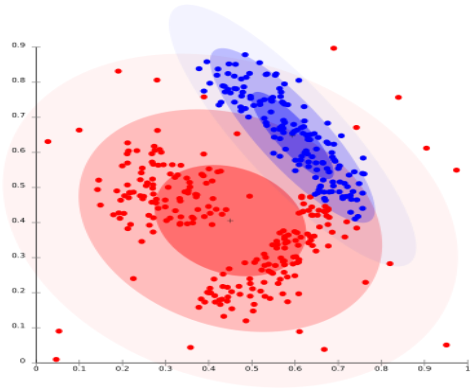

Fig. 1 Distribution Based Method

2) **Centroid based methods :**This is basically one of iterative clustering algorithm in which the clusters are formed by the closeness of data points to the centroid of clusters. Here, the cluster center i.e. centroid is formed such that the distance of data points is minimum with the center. This problem is basically one of NP- Hard problem and thus solutions are commonly approximated over a number of trials.
3) **Connectivity based methods:** The core idea of connectivity based model is similar to Centroid based model which is basically defining clusters on the basis of closeness of data points .Here we work on a notion that the data points which are closer have similar behavior as compared to data points that are farther .It is not a single partitioning of the data set , instead it provides an extensive hierarchy of clusters that merge with each other at certain distances. Here the choice of distance function is subjective. These models are very easy to interpret but it lacks scalability.
4) **Density Models** Density based clustering methods allow the identification of arbitrary, not necessarily convex regions of data points that are densely populated. The number of clusters does not need to be specified beforehand; a cluster is defined to be a connected region that exceeds a given density threshold. Local scaling in density based clustering, which determines the density threshold based on the local statistics of the data. The local maxima of density are discovered using a k-nearest-neighbor density estimation and used as centers of potential clusters. Each cluster is grown until the density falls below a pre-specified ratio of the center point's density. The resulting clustering technique is able to identify clusters of arbitrary shape on noisy backgrounds that contain significant density gradients.
5) **Subspace clustering :** Subspace clustering is an unsupervised learning problem that aims at grouping data points into multiple clusters so that data point at single cluster lie approximately on a low-dimensional linear subspace. Subspace clustering is an extension of feature selection just as with feature selection subspace clustering requires a search method and evaluation criteria but in addition subspace clustering limit the scope of evaluation criteria. Subspace clustering algorithm localize the search for relevant dimension and allow to them to find cluster that exist in multiple overlapping subspaces. Subspace clustering was originally purpose to solved very specific computer vision problem having a union of subspace structure in the data but it gains increasing

attention in the statistic and machine learning community. People use this tool in social network, movie recommendation, and biological dataset. Subspace clustering raise the concern of data privacy as many such application involve dealing with sensitive information. Data points are assumed to be incoherentas it only protects the differential privacy of any feature of a user rather than the entire profile user of the database. There are two branches of subspace clustering based on their search strategy.

1. Top-down algorithms find an initial clustering in the full set of dimension and evaluate the subspace of each cluster.
2. Bottom-up approach finds dense region in low dimensional space then combine to form clusters.

## III. Hub-Based Clustering

Hubness, which is the tendency of some data points in high-dimensional data sets to occur much more frequently in k-nearest neighbor lists of other points than the rest of the points from the set, can in fact be used for clustering. In experiments on synthetic data author show that hubness is a good measure of point centrality within a high-dimensional data cluster and that major hubs can be used effectively as cluster prototypes [1].

Consequences and applications of hubness have been more thoroughly investigated in other related fields: classification, image feature representation, data reduction, collaborative filtering, text retrieval, and music retrieval. In many of these studies it was shown that hubs can offer valuable information that can be used to improve existing methods and devise new algorithms for the given task[8-11].

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, researcher 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 centers of compact subclusters in high-dimensional data, a natural way to test the feasibility of using them to approximate these centers 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 [1].

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.

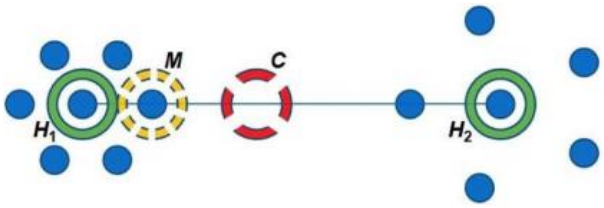


Fig. 2 Hubness based K-Means clustering

Consider the simple example shown in Fig. 2 where The red dashed circle marks the centroid (C), yellow dotted circle the medoid (M), and green circles denote two elements of highest hubness (H1; H2), for neighborhood size 3. This 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 neighboring elements and, therefore, carry localized centrality information.

Computational complexity of hubness-based algorithms is mostly determined by the cost of computing hubness scores. Several fast approximate approaches are available. It was demonstrated that it is possible to construct an approximate k-NN graph from which hubness scores can be read in $\theta(ndt)$ time, where the user-defined value t > 1 expresses the desired quality of graph construction. It was reported that good graph quality may be achieved with small values of t, which we were able to confirm in our initial experiments. Alternatively, locality-sensitive hashing could also be used, as such methods have become quite popular recently.

## IV. The Emergence of Hubs

Hubness is closely related to the aforementioned concentration of distances in highdimensional spaces. If distances do concentrate for a given data set, then its points are lying approximately on a hypersphere centered at the data mean. Naturally, if data is drawn from several distributions, as is usually the case in clustering problems, this could be rephrased by saying that data are lying approximately on several hyperspheres centered at the corresponding distribution means. However, it has been shown that the variance of distances to the mean is still non-negligible, regardless of the concentration phenomenon – for any finite number of dimensions [18]. This implies that some of the points will still end up being closer to the data (or cluster) mean than other points. It is well known that points closer to the mean tend to, on average, be closer to all other points, for any observed dimensionality. However, in high-dimensional data, this tendency is amplified [19]. On average, points which are closer to all other points will naturally have a higher probability of being included in k-nearest-neighbor lists of other points in the data set, which gives rise to an increase in their hubness scores.

## V. Relation of Hubs to Data Clusters

There has been some previous work on how well high-hubness elements cluster, as well as the general impact of hubness on clustering algorithms [119]. A correlation between low-hubness elements and outliers was also observed. A low hubness score indicates that a point is on average far from the rest of the points and hence probably an outlier. In high-dimensional spaces, however, low-hubness elements are expected to occur by the very nature of these spaces and data distributions. These data points will lead to an average increase in intra-cluster dissimilarity. It was also shown for several clustering algorithms that hubs do not cluster well compared to the rest of the points. This is due to the fact that some hubs are actually close to points in different clusters. Hence, they also lead to a decrease in inter-cluster dissimilarity. However, this does not necessarily hold for an arbitrary cluster configuration. It was already mentioned that points closer to cluster means tend to have higher hubness scores than the rest of the points. A natural question which arises is: Are hubs medoids? When observing the problem from the perspective of partitioning clustering approaches, of which K-means is the most commonly used representative, a similar question might also be posed: Are hubs the closest points to data centroids in clustering iterations? To answer this question, we ran K-means++ [20] multiple times on several randomly generated Gaussian mixtures for various fixed numbers of dimensions, observing the high-dimensional case. Researcher measured in each iteration the distance from current cluster centroid to the medoid and to the hub, and scaled by the average intracluster distance. This was measured for every cluster in all the iterations, and for each iteration the minimal and maximal distance from any of the centroids to the corresponding hub and medoid were computed.

It can be noticed that, in the low-dimensional case, hubs in the clusters are far away from the centroids,

even farther than average points. There is no correlation between data means and high-hubness instances in the low-dimensional scenario. On the other hand, for the high-dimensional case, we observe that the minimal distance from centroid to hub converges to minimal distance from centroid to medoid. This implies that some medoids are in fact cluster hubs. Maximal distances to hubs and medoids, however, do not match. There exist hubs which are not medoids, and vice versa. Also, we observe that maximal distance to hubs also drops with iterations, hinting that as the iterations progress, centroids are becoming closer and closer to data hubs. This brings us to the idea that will be explained in detail in the following section: Why not use hubs to approximate data centers? After all, we expect points with high hubness scores to be closer to centers of relatively dense regions in high-dimensional spaces than the rest of the data points, making them viable candidates for representative cluster elements. We are not limited to observing only the points with the highest hubness scores, we can also take advantage of hubness information for any given data point. More generally, in case of irregularly shaped clusters, hubs are expected to be found near the centers of compact sub-clusters, which is also beneficial.

## VI. Conclusion and Future Work

Hub-based algorithms are designed specifically for high dimensional data. This is an unusual property, since the performance of most standard clustering algorithms deteriorates with an increase of dimensionality. The existing algorithms represent only one possible approach to using hubness for improving high-dimensional data clustering. It is also intended to explore other closely related research directions, including kernel mappings and shared neighbor clustering. This would allow us to overcome the major drawback of the existing methods detecting only hyper spherical clusters, just as K-Means. Additionally, one can explore methods for using hubs to automatically determine the number of clusters in the data.

## VII. REFERENCES


[1]. Nenad T., Milos R., Dunja M., and Mirjana I., "The Role of Hubness in Clustering High-Dimensional Data" IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 3, March 2014

[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.

[3]. 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.

[4]. 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.

[5]. 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

[6]. Weber R., Schek H.-J., Blott S.: A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. VLDB Conference Proceedings, 1998.

[7]. Ergun Bic¸ici and Deniz Yure, "Locally Scaled Density Based Clustering", Proc. Eighth Int'l Conf. Adaptive and Natural Computing Algorithms (ICANNGA), Part I, pp. 739-748, 2007

[8]. N. Tomasev, M. Radovanovic, D. Mladenic, and M. Ivanovic, "Hubness-Based Fuzzy Measures for High-Dimensional kNearest Neighbor Classification," Proc. Seventh Int'l Conf. Machine

Learning and Data Mining (MLDM), pp. 16-30, 2011.

[9]. N. Tomasev, M. Radovanovic, D. Mladenic, and M. Ivanovic, "A Probabilistic Approach to Nearest-Neighbor Classification: Naive Hubness Bayesian kNN," Proc. 20th ACM Int'l Conf. Information and Knowledge Management (CIKM), pp. 2173-2176, 2011.

[10]. M. Radovanovic, A. Nanopoulos, and M. Ivanovic, "Hubs in Space: Popular Nearest Neighbors in High-Dimensional Data," J. Machine Learning Research, vol. 11, pp. 2487-2531, 2010.

[11]. N. Tomasev, R. Brehar, D. Mladenic, and S. Nedevschi, "The Influence of Hubness on Nearest-Neighbor Methods in Object Recognition," Proc. IEEE Seventh Int'l Conf. Intelligent Computer Comm. and Processing (ICCP), pp. 367-374, 2011.

[12]. C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystr¨om method," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, pp. 214-225, 2004.

[13]. M. Li, J. T. Kwok, and B. L. Lu, "Making large-scale nystrm approximation possible," in Proceeding of 27th International Conference on Machine Learning, pp. 631-638, 2010.

[14]. D. Yan, L. Huang, and M. I. Jordan, "Fast approximate spectral clustering," in Proceeding of 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 907-916, 2009.

[15]. H. Shinnou and M. Sasaki, "Spectral clustering for a large data set by reducing the similarity matrix size," in Proceeding of International Conference on Language Resources and Evaluation, pp. 201-204, 2008.

[16]. Nenad Tomašev, Miloš Radovanovi´c, Dunja Mladeni´c, A Probabilistic Approach to Nearest-Neighbor Classification: Naive Hubness Bayesian Knn", CIKM'11 , Glasgow, Scotland, UK, 24-28, October 2011.

[17]. Thomas Low1, Christian Borgelt, Sebastian Stober, and Andreas N¨urnberger, "The Hubness Phenomenon: Fact or Artifact?" , Studies in Fuzziness and Soft Computing, 267-278, January 2013

[18]. Franc¸ois, D., Wertz, V., Verleysen, M.: The concentration of fractional distances. IEEE Transactions on Knowledge and Data Engineering 19(7) 873-886, 2007

[19]. Radovanovi´c, M., Nanopoulos, A., Ivanovi´c, M.: Hubs in space: Popular nearest neighbors in high-dimensional data. Journal of Machine Learning Research 11, 2487-2531,2010

[20]. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 1027-1035, 2007

**Cite this article as :**

Miss. Archana Chaudahri, Mr. Nilesh Vani, "Survey on Clustering High-Dimensional data using Hubness", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 6 Issue 1, pp. 01-07, January-February 2020. Available at doi : https://doi.org/10.32628/CSEIT195671
Journal URL : http://ijsrcseit.com/CSEIT195671