INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

Hubness in High-Dimensional Data Clustering

R. S. Ansila[1], S. Sasikala[2]

[1]Research Scholar Computer Science,
Sree Saraswathi Thyagaraja College,
Pollachi-642107, Coimbatore,
Tamilnadu, India.
ansila.suresh@gmail.com
[2]HOD UG Department of Computer Science,
Sree Saraswathi Thyagaraja College,
Pollachi-642107, Coimbatore,
Tamilnadu, India.
sasivenkatesh04@gmail.com

Abstract:*-Learning from high-dimensional data is usually quite a challenging task, as captured by the well known phrase curse of dimensionality. Clustering depends critically on density and distance (similarity), but these concepts become increasingly more difficult to define as dimensionality increases. Most distance based methods become impaired due to the distance concentration of many widely used metrics in high-dimensional spaces. In particular, we use a similarity measure that is based on the number of neighbors that two points share, and define the density of a point as the sum of the similarities of a point's nearest neighbors The impact of hubness on forming shared neighbor distances has not been discussed before and it is what we focus on in this paper. This approach handles many problems that traditionally plague clustering algorithms. Finding clusters in the presence of noise and outliers and finding clusters in data that has clusters of different shapes, sizes, and density.*

Keywords: *Hubness, High-dimensional data, shared neighbor clustering, K-nearest neighbor search.*

1. INTRODUCTION

Machine learning in many dimensions is often very difficult, due to interplay of several prohibitive factors. This is usually referred to as the *curse of dimensionality*. In high-dimensional spaces, all data is sparse, as the requirements for proper density estimates rise exponentially with the number of features. Empty space predominates [11] and data lies approximately on the surface of hyper-spheres around cluster means, i.e.in distribution tails. Relative contrast between distances on sample data is known to decrease with increasing dimensionality, i.e. the distances concentrate [12-13].The expectation of the distance value increases, but the variance remains constant. It is therefore much more difficult to distinguish between close and distant points. This has a profound impact on nearest neighbor methods, where inference is done based on the k instances most similar (relevant) to the point of interest. The very concept of a nearest neighbor was said to be much less meaningful in high-dimensional data [14]. Difficulty in distinguishing between relevant and irrelevant points is, however, not the only aspect of the dimensionality curse which burdens k-nearest neighbor based inference. 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 [15]. 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 [9], 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 high-dimensional 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)[5][7]. We have measured the hubness of the induced shared neighbor spaces and have shown that hubness-aware

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS…..

k-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. This is the main contribution of the paper. Our experiments reveal a consistent and significant improvement when using the newly proposed approach. The paper is structured as follows. In Section 2 we outline the general motivation for using both the shared neighbor distances and the hubness-aware methods when learning from high-dimensional data, by reviewing some of the recent work in both areas. We proceed by discussing how the two approaches might be successfully combined and propose a new way to define shared neighbor similarities in Section 3. In Section 4 we test our hypothesis on several high-dimensional real world and synthetic datasets and discuss our findings.

2. RELATED WORK

The deliberation of distances is an aspect of the curse of dimensionality [14][5], which is a general term for problems of learning in high dimensional spaces. It is the astonishing characteristic of all points in a high dimensional space to be at almost the same distance to all other points in that space .It is usually measured as a ratio between spread and magnitude ,e.g. the ratio between the standard deviation of all distances to an random reference point and the mean of these distances .If the standard deviation stays more or less constant with growing dimensionality while the mean keeps growing, the ratio converges to zero with dimensionality going to infinity .In such a case it is said that the distances concentrate. This is a natural outcome of high dimensionality and has been studied for Euclidean spaces and other l_p norms .For cosine distances it has been shown that the mean stay constant while the standard deviation diminishes with the ratio a gain converging to zero. It is clear that this occurrence has an impact on any algorithm based on measuring distances in high dimensional spaces, e.g. even the meaningfulness of simple nearest neighbor based approaches in high dimensions has been doubted[10]. But it should also be mentioned that incase the data space exhibits a stable cluster configuration (i.e. between-cluster distances dominate within-cluster distances), the distances should not focus at all. To avoid this problem of concentration of distances the use of 'Shared Neighbor Distances' has been proposed by Houle et al. who raised the question whether these secondary distances are able to "defeat the curse of dimensionality".'Shared nearest neighbors' (SNN) was first proposed as a similarity measure by Jarvis and Patrick to improve the clustering of non-globular clusters. As the name suggest, SNN similarity is based on computing the overlap between the k nearest neighbors of two objects. SNN approaches try to symmetries nearest neighbor affairs using only rank and not distance information. Houle et al. argued, that the rank information SNN is based on might still be significant even when distances concentrate in high dimensions. In an extensive study using artificial and three real world image acknowledgment data sets the authors show that SNN is indeed able to reduce the concentration of distances. The secondary SNN distances also result in improved image clustering rates calculated as area under receiver operating curve based on nearest neighbor. But the authors do not make a connection to the' hubness' phenomenon which at the time of their study was not very well-known [2].

3. SHARED NEIGHBOR CLUSTERING

Regardless of the uncertainty expressed in [14], nearest neighbor queries have been shown to be meaningful in high-dimensional data under some natural postulation, at least when it comes to distinguishing between different clusters of data points. If the clusters are pair wise stable, i.e. inter-cluster distances dictate intra-cluster distances, the neighbors will tend to belong to the same cluster as the original point. A clear issue with this line of reasoning is that cluster assumption abuse is present to different degrees in real world data, so that sometimes the categories do not correspond well to the aforesaid clusters. Nevertheless, this observation motivated the researches to consider using secondary distances based on the ranking induce by the original similarity measure [2]

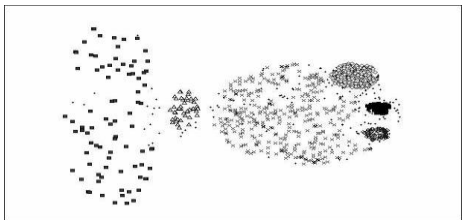


Figure: 1 SNN Clustering

Let D = $(x_1, y_1), (x_2, y_2), ..(x_n, y_n)$ be the data set, where each x_i 2 R_d. The x_i are feature vectors which reside in some high-dimensional Euclidean space, and y_i 2 $c_1, c_2, ..c_C$ are the labels. Denote by $D_k(x_i)$ the k-neighborhood of x_i. A shared neighbor similarity

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

between two points is then usually defines as: $simcos_s(x_i, x_j) = |D_s(x_i) \setminus D_s(x_j)|s(1)$ where we have used s to denote the neighborhood size, since we will use these similarity measures to perform k-nearest neighbor classification, and the neighborhood sizes in these two cases will be different. The $simcos_s$ similarity can easily be transformed into a distance measure. the above given distance measures produce the same ranking, so they are equivalent when being used for k-nearest neighbor inference. In shared neighbor distances, all neighbors are treated as being equally relevant. We argue that this view is inherently flawed and that its deficiencies become more pronounced when the dimensionality of the data is increased. Admittedly, there have been some previous experiments on including weights into the SNN framework for clustering, but these weights were associated with the positions in the neighbor list, not with neighbor objects themselves. [2]In Section 3 we will discuss the role of hubness in SNN measures.

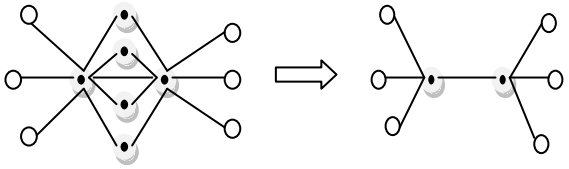


Figure: 2 SNN Graph

4. HUBS: VERY FREQUENT NEAREST NEIGHBOR

High dimensionality gives rise to *hubs*, significant objects which frequently occur as neighbors to other points [3]. Most instances, on the other hand, are very infrequently included ink-neighbor sets, thereby having little or no influence on consequent clustering .What this change in the k-occurrence distribution entails is that potential errors, if present in the hub points, can easily circulate and compromise many k-neighbor sets. Further more, hubness is a statistical property of inherently high-dimensional data, as the points closer to the centers of hyper-spheres where most of the data lies tend to become very close to many points and are hence often included as neighbors [6]. This means that hubness of a particular point has little to do with its semantics. Hubs are often not only neighbors to objects of their own category, but also neighbors to many points from other categories as well. In such cases, they exhibit a highly detrimental manipulate and this is why hubness of the data usually hampers k-nearest neighbor clustering .Hubness-aware algorithms have recently been proposed for clustering , instance selection outlier and anomaly recognition and clustering ,which we will discuss below .Let us introduce some notation. Denote by $R_k(x_i)$ the reverse neighbor set of x_i, so the number of k-occurrences is then $N_k(x_i) = |R_k(x_i)|$. This total number of neighbor occurrences includes both the *good* occurrences, where the labels of a point and its neighbor match and the *bad* occurrences where there is label mismatch. Formally, $Nk(x_i) = GN_k(x_i) + BN_k(x_i)$, the former being referred to as the good hubness and the latter as the bad hubness of x_i. The bad hubness itself can be viewed as a composite quantity, comprising all the class-specific k-occurrences where label mismatch occurs. Let $N_{k,c}(x_i) = |x\ 2\ R_k(x_i) : y = c|$ denote such class-specific hubness. The total occurrence frequency is then simply $N_k(x_i) = P_{c2C} N_{k,c}(x_i)$. [2][1]

4.1 Hubness-Aware Shared-Neighbor Distances

Since hubness affects the distribution of neighbors, it must also have an effect on the distribution of neighbors shared between different points [3]. Each x_i is shared between $N_s(x_i)$ data points and participate in $\square_{Ns(xi)\ 2}$ _ similarity scores. Hub points, by the virtue of being very frequent neighbors, are predictable to arise quite frequently as shared neighbors in pair wise object comparison. What this means, however, is that sharing a hub s-neighbor is quite common and not very informative. This is reliable with observations. Rarely shared neighbors (anti-hubs), on the other hand, carry information more local to the points of interest and should be given inclination when calculating similarities [3]. Figure 3 outlines this observation

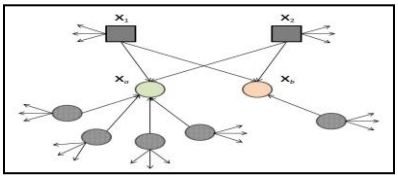


Figure: 3 Shared Neighbor Clustering

.**Figure:3** An illustrative example. x1 and x2 share two neighbors, Ds(x1) [Ds(x2) = xa, xb. The two shared neighbors are not indicative of the same level of similarity, as xb is a neighbor only to x1, x2 and one other point, while xa is a more frequently shared neighbor.

One of the most important properties desired in a metric is to allow for good separation between data clusters. This is achieved by minimizing the intra-class distances while maximizing the inter-class distances. Let us compare several types of hub-points from this perspective. There are some hubs which occur almost always as neighbors to points from a single category. Obviously, increasing their weight in the similarity measure would also increase intra-class pair wise similarity. Other hubs occur as neighbors to

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

many different categories inconsistently. Reducing their weight in the similarity measure would certainly reduce inter-class similarity [2]. This is illustrated in Figure 3. The purity of the reverse neighbor sets can clearly be exploited for improving class separation.

4.2 Hub-Based Clustering

If hubness is viewed as a kind of local centrality measure, it may be probable to use hubness for clustering in various ways. To test this hypothesis, we opted for an move toward that allows observations about the quality of resulting clustering configurations to be related directly to the property of hubness, instead of being a outcome of some other attribute of the clustering algorithm. Since it is expected of hubs to be located near the centers of dense sub clusters in high-dimensional data, a natural way to test the feasibility of using them to approximate these centers is to compare the hub-based advance 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[8][9]. Centroids and medoids in K-means iterations tend to converge to locations close to high-hubness points, which imply 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. 3, which mimics in two dimensions what normally happens in multidimensional data, and suggest 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. 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 listsof elements within the same cluster.[2][6]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 [1]

4.3 Deterministic Approach

A simple way to employ hubs for clustering is to use them as one would normally use centroids. In addition, this allows us to make a direct comparison with the K-means method. The algorithm, referred to as K-hubs, is given in Algorithm 1.

Algorithm 1. K-hubs.
Initialize Cluster Centers ();
Cluster [] clusters ¼ form Clusters ();
Repeat
For all Cluster c 2 clusters do
Data Point h ¼ find Cluster Hub(c);
Set Cluster Center (c, h);
end for
Clusters ¼ form Clusters ();
Until no Reassignments
return clusters

After initial evaluation on synthetic data, it became clear that even though the algorithm manages to find good and even best configurations often, it is quite sensitive to initialization. To increase the probability of finding the global optimum were sorted to the stochastic approach described in the following section. However, even though K-hubs exhibited low stability, it converges to cluster configurations very quickly, in no more than four iterations on all the data sets used for testing, most of which contained around 10,000 data instances. hubness scores[1].

4.4 Probabilistic Approach

Ordinary distance measures have problems– Euclidean distance is less appropriate in high-dimensions– Cosine and Jaccard measure take in to account presences, but do not satisfy the triangle inequality[4][1]

Algorithm 2 SNN clustering
1. Compute the similarity matrix
2. Sparsify the similarity matrix by keeping only the *k* most similar neighbors
3. Construct the shared nearest neighbor graph from the sparsified similarity matrix
4. Find the SNN density of each point
5. Find the core points
6. Form clusters from the core points
7. Discard all noise points
8. Assign all non-noise, non-core points to clusters[4]

5. EXPERIMENTS AND EVALUATION

We tested our approach on various high-dimensional synthetic and real-world data sets. We will use the following abbreviations in the forthcoming discussion: K-Means(KM), kernel K-means (Ker-KM), Global K-Hubs(GKH), Local K-Hubs (LKH), Global Hubness-Proportional Clustering (GHPC) and Local Hubness-Proportional Clustering (LHPC), Hubness-Proportional K-Means(HPKM), local and global referring to the type of hubness score that was used (see Section 4). For all centroid-based algorithms, including KM, we used the D_2 (K-means++) initialization procedure [1]. The neighborhood size of k ¼10 was used by default in our experiments involving synthetic data and we have experimented with different neighborhood size in different real-world tests. There is no known way of

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

selecting the best k for finding neighbor sets, the problem being domain-specific. To check how the choice of k reflects on hubness-based clustering, we ran a series of tests on a fixed 50-dimensional 10-distribution Gaussian mixture for a range of k values, k 2 f1; 2; . . . ; 20g. The results are summarized in Fig. 6. It is clear that, at least in such simple data, the hubness-based GHPC algorithm is not overly sensitive on the choice of k.In the following sections, K-means++ will be used as the main baseline for comparisons, since it is suitable for determining the feasibility of using hubness to estimate local centrality of points. Additionally, we will also compare the proposed algorithms to kernel K-means [1] and one standard density-based method, GDBScan [1].Kernel K-means was used with the nonparametric histogram intersection kernel, as it is believed to be good for image clustering and most of our real-world data tests were done on various sorts of image data. Kernel methods are naturally much more powerful, since they can handle non hyper-spherical clusters. Yet, the hubness-based methods could just as easily be "kernelized," pretty much the same way it was done for K-means. This idea requires further tests and is beyond the scope of this paper. For evaluation, we used repeated random sub sampling, training the models on 70 percent of the data and testing them on the remaining 30 percent. This was done to reduce the potential impact of over fitting, even though it is not a major issue in clustering, as clustering is mostly used for pattern detection and not prediction. On the other hand, we would like to be able to use the clustering methods not only for detecting groups in a given sample, but rather for detecting the underlying structure of the data distribution in general[1].

TABLE 1: Averaged Results of Algorithm Runs on High-Dimensional Mixtures of Gaussians

		KM++	GHPC	SNNC
K=5	Silhouette	0.56±0.02	0.61±0.02	0.71±0.02
	Entropy	0.10±0.01	0.06±0.01	0.09±0.01
	Perfect	0.54±0.04	0.76±0.06	0.79±0.07
K=10	Silhouette	0.52±0.01	0.57±0.01	0.62±0.01
	Entropy	0.13±0.01	0.08±0.01	0.06±0.02
	Perfect	0.11±0.02	0.39±0.06	0.41±0.06

5.1 Synthetic Data: Gaussian Mixtures

In the first group of experiments, we wanted to compare the value of global vs. local hub-ness scores. These initial tests were run on synthetic data and do not include HPKM, as the hybrid approach was introduced later for tackle problems on real-world data. For comparing the resulting clustering quality, we used mainly the silhouette index as an unsupervised measure of pattern validity, and average cluster entropy as a supervised measure of clustering homogeneity. Since most of the generated data sets are "solvable," i.e., consisting of non overlapping Gaussian distributions, we also report the normalized frequency with which the algorithms were able to find these perfect configurations. We ran two Ines of experiments, one using five Gaussian generators, the other using 10. For each of these, we generated data of 10 different high dimensionalities: 10, 20, . . . , 100. In each case, 10 different Gaussian mixtures were generated, resulting in 200 different generic sets, 100 of them containing five data clusters, the others containing 10. On each of the data sets, KM++ and all of the hub-based algorithms were executed 30 times and the averages of performance measures were computed. The generated Gaussian distributions were hyper spherical (diagonal covariance matrices, independent attribute). Distribution means were drawn randomly from ½lmbound; um bound_d, lmbound ¼ _20; um bound ¼ 20 and the standard deviations were also uniformly taken from ½l_bound; u_bound_d, l_bound ¼ 2; u_ bound ¼ 5. Probabilistic approaches significantly outperform the deterministic ones, even though GKH and LKH also sometimes converge to the best configurations, but much less frequently. More importantly, the best overall algorithm in these tests was GHPC, which outperformed KM++ on all bases, having lower average entropy, a higher silhouette index, and a much higher frequency of finding the perfect configuration. This suggests that GHPC is a good option for clustering high-dimensional Gaussian mixtures. Regarding the number of dimensions when the actual improvements begin to show, in our lower dimensional test runs, GHPC was better already on 6-dimensional mixtures. Since we concluded that using global hubness leads to better results, we only consider GKH and GHPC in the rest of the experiments. [1][4]

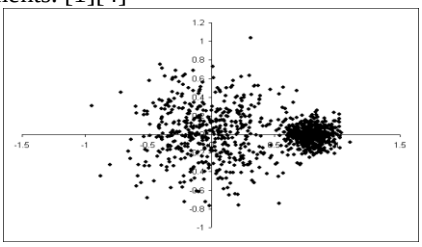


Figure:4 Gassian Datset

5.2 Clustering and High Noise Levels

Real-world data often contain noisy or untrue values due to the nature of the data-collecting process. It can be assumed that hub-based algorithms will be more vigorous with respect to noise, since hubness-proportional search is driven mostly by the highest-hubness essentials, not the outliers. In the case of KM++, all instances from the current cluster directly determine the location of the centroid in the next iteration. When the noise level is low, some sort of outlier removal

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

technique may be applied. In setups involving high levels of noise, this may not be the case [1][4]

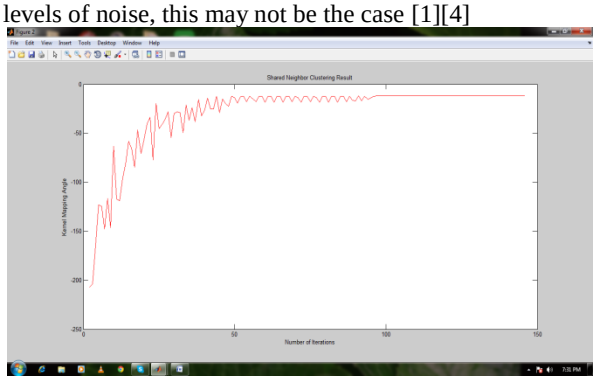
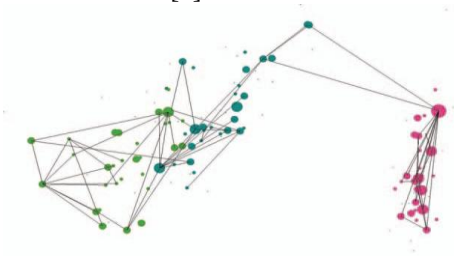
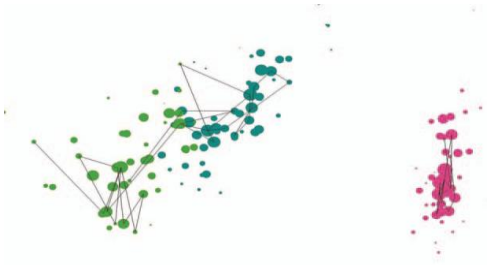


Figure: 5 Gradual changes in cluster quality measures with rising noise levels. The difference between the algorithm performances is much more pronounced in the high-dimensional case.

It can be seen that HPC search through many different hub-configurations before settling on the final one. Also, what seems to be the case, at least in the majority of generated images, is that the search is somewhat wider for lower k-values. This observation is reasonable due to the fact that with an increase in neighborhood size, more points have hubness greater than a certain threshold and it is easier to distinguish between genuine outliers and slightly less central regular points. Currently, we do not have a universal robust solution to the problem of choosing a k value. This is, on the other hand, an issue with nearly all kNN-based methods, with no simple, efficient, and general work-around [1].



(a) K=5



(b) K=10

Figure: 6 Hubness-guided search for the best cluster hub-configuration in HPC on Iris data

6. CONCLUSION

In this paper we proposed a new shared nearest neighbor similarity measure. This is especially important in high-dimensional data, where hubness plays an important role as a nearest neighbor related aspect of the more general curse of dimensionality. As shared neighbor distances have in general been recommended specifically for high-dimensional data, enriching them with hubness information is even more significant. 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. Hubness, on the other hand, is a property of intrinsically high-dimensional data, and this is precisely where GHPKM and GHPC excel, and are expected to offer improvement by providing higher inter cluster distance, i.e., better cluster separation. The existing algorithms represent only one possible approach to using hubness for improving high-dimensional data clustering. We also intend 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, we would like to explore methods for using hubs to automatically determine the number of clusters in the data.

REFERENCE

[1] The Role of Hubness in Clustering High-Dimensional Data, Nenad Toma_sev, Milo_s Radovanovi_c, Dunja Mladeni_c, and Mirjana Ivanovi_IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 3, MARCH 2014

[2] Hubness-aware Shared Neighbor Distances for High-dimensional k-Nearest Neighbor Classification Nenad Tomasev and Dunja Mladeni

[3] Can Shared Nearest Neighbors Reduce Hubness in High-Dimensional Spaces? Arthur Flexer, Dominik Schnitze

[4] A New Shared Nearest Neighbor Clustering Algorithm and its Applications, Levent Ertöz, Michael Steinbach, Vipin Kumar

[5] A Probabilistic Approach to Nearest-Neighbor Classification: Naive Hubness Bayesian *K*nn Nenad Tomašev, Miloš Radovanovi´c, Dunja Mladeni´c

[6] A Study on Clustering High Dimensional Data Using Hubness Phenomenon, V.Suganthi1, S.Tamilarasi

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

[7] The Hubness Phenomenon: Fact or Artifact? Thomas Low1, Christian Borgelt2, Sebastian Stober1, and Andreas N¨urnberger1

[8] A Survey on Various Clustering Techniques with K-means Clustering Algorithm in Detail, Supreet Kaur1, Usvir Kaur2

[9] Efficient High Dimensional Data Clustering Using Hubness Phenomenon

[10] The influence of weighting the k-occurrences on hubness-aware classification methods, *Nenad Tomašev, Dunja Mladenić*

[11] Scott, D., Thompson, J.: Probability density estimation in higher dimensions. In: Proceedings of the Fifteenth Symposium on the Interface. (1983) 173–179

[12] Aggarwal, C. C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional spaces. In: Proc. 8th Int. Conf. on Database Theory (ICDT). (2001) 420–434

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

[14] Durrant, R. J., Kaban, A.: When is 'nearest neighbour' meaningful: A converse theorem and implications. Journal of Complexity 25(4) (2009) 385–397

[15] Radovanovic, M., Nanopoulos, A., Ivanovi´c, M.: Nearest neighbors in high-dimensional data: The emergence and influence of hubs. In: Proc. 26th Int. Conf. on Machine Learning (ICML). (2009) 865–872.