The Role of Hubness in Clustering High-Dimensional Data

Nenad Tomašev, Miloš Radovanović, Dunja Mladenić, and Mirjana Ivanović

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. In this paper, we take a novel perspective on the problem of clustering high-dimensional data. 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. More specifically, we show that hubness, i.e., the tendency of high-dimensional data to contain points (hubs) that frequently occur in k-nearest-neighbor lists of other points, can be successfully exploited in clustering. We validate our hypothesis by demonstrating that hubness is a good measure of point centrality within a high-dimensional data cluster, and by proposing several hubness-based clustering algorithms, showing that major hubs can be used effectively as cluster prototypes or as guides during the search for centroid-based cluster configurations. Experimental results demonstrate good performance of our algorithms in multiple settings, particularly in the presence of large quantities of noise. The proposed methods are tailored mostly for detecting approximately hyperspherical clusters and need to be extended to properly handle clusters of arbitrary shapes.

Index Terms—Clustering, curse of dimensionality, nearest neighbors, hubs

1 Introduction

Clustering in general is an unsupervised process of grouping elements together, so that elements assigned to the same cluster are more similar to each other than to the remaining data points [1]. This goal is often difficult to achieve in practice. 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. This is mostly 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 [6], [7], [8], [9].

The difficulties in dealing with high-dimensional data are omnipresent and abundant. However, not all phenomena that arise are necessarily detrimental to clustering techniques. We will show in this paper that *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. To our knowledge, this has not been previously attempted. In a limited sense, hubs in graphs have been used to represent typical word meanings in [10], which was not used for data clustering. A similar line of research has identified essential proteins as hubs in the reverse nearest neighbor topology of protein interaction networks [11]. We have focused on exploring the potential value of using hub points in clustering by designing hubness-aware clustering algorithms and testing them in a high-dimensional context. The hubness phenomenon and its relation to clustering will be further addressed in Section 3.

There are two main contributions of this paper. First, in experiments on synthetic data we 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. In addition, we propose three new clustering algorithms and evaluate their performance in various high-dimensional clustering tasks. We compared the algorithms with a baseline state-of-the-art prototype-based method (K-means++ [12]), as well as kernel-based

- *N. Tomašev and D. Mladenić are with the Jožef Stefan Institute, Artificial Intelligence Laboratory and Jožef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia. E-mail: {nenad.tomasev, dunja.mladenic}@ijs.si.*
- *M. Radovanović and M. Ivanović are with the Department of Mathematics and Informatics, University of Novi Sad, Trg D. Obradovića 4, 21000 Novi Sad, Serbia. E-mail: {radacha, mira}@dmi.uns.ac.rs.*

Manuscript received 2 Apr. 2012; revised 16 Nov. 2012; accepted 23 Jan. 2013; published online 31 Jan. 2013.
Recommended for acceptance by C. Böhm.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2012-04-0223.
Digital Object Identifier no. 10.1109/TKDE.2013.25.

1041-4347/14/$31.00 © 2014 IEEE Published by the IEEE Computer Society
Authorized licensed use limited to: Universiteit van Tilburg. Downloaded on November 23,2021 at 15:51:37 UTC from IEEE Xplore. Restrictions apply.

and density-based approaches. The evaluation shows that our algorithms frequently offer improvements in cluster quality and homogeneity. The comparison with kernel K-means [13] reveals that kernel-based extensions of the initial approaches should also be considered in the future. Our current focus was mostly on properly selecting cluster prototypes, with the proposed methods tailored for detecting approximately hyperspherical clusters.

The rest of the paper is structured as follows: In the next section, we present the related work, Section 3 discusses in general the phenomenon of hubness, while Section 4 describes the proposed algorithms that are exploiting hubness for data clustering. Section 5 presents the experiments we performed on both synthetic and real-world data. We expect our observations and approach to open numerous directions for further research, many of which are outlined by our final remarks in Section 6.

2 Related Work

Even though hubness has not been given much attention in data clustering, hubness information is drawn from k-nearest-neighbor lists, which have been used in the past to perform clustering in various ways. These lists may be used for computing density estimates, by observing the volume of space determined by the k-nearest neighbors. Density-based clustering methods often rely on this kind of density estimation [14], [15], [16]. The implicit assumption made by density-based algorithms is that clusters exist as high-density regions separated from each other by low-density regions. In high-dimensional spaces this is often difficult to estimate, due to data being very sparse. There is also the issue of choosing the proper neighborhood size, since both small and large values of k can cause problems for density-based approaches [17].

Enforcing k-nearest-neighbor consistency in algorithms such as K-means was also explored [18]. The most typical usage of k-nearest-neighbor lists, however, is to construct a k-NN graph [19] and reduce the problem to that of graph clustering.

Consequences and applications of hubness have been more thoroughly investigated in other related fields: classification [20], [21], [22], [23], [24], image feature representation [25], data reduction [23], [26], collaborative filtering [27], text retrieval [28], and music retrieval [29], [30], [31]. 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.

Finally, the interplay between clustering and hubness was briefly examined in [23], where it was observed that hubs may not cluster well using conventional prototype-based clustering algorithms, since they not only tend to be close to points belonging to the same cluster (i.e., have low intracluster distance) but also tend to be close to points assigned to other clusters (low intercluster distance). Hubs can, therefore, be viewed as (opposing) analogues of outliers, which have high inter- and intracluster distance, suggesting that hubs should also receive special attention [23]. In this paper, we have adopted the approach of using hubs as cluster prototypes and/or guiding points during prototype search.

3 The Hubness Phenomenon

Hubness is an aspect of the curse of dimensionality pertaining to nearest neighbors which has only recently come to attention, unlike the much discussed distance concentration phenomenon. Let $D \subset \mathbb{R}^d$ be a set of data points and let $N_k(x)$ denote the number of *k-occurrences* of point $x \in D$, i.e., the number of times x occurs in k-nearest-neighbor lists of other points from D. As the dimensionality of data increases, the distribution of k-occurrences becomes considerably skewed [23]. As a consequence, some data points, which we will refer to as *hubs*, are included in many more k-nearest-neighbor lists than other points. In the rest of the text, we will refer to the number of k-occurrences of point $x \in D$ 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 artifact of finite samples nor a peculiarity of some specific data sets [23]. Naturally, the exact degree of hubness may still vary and is not uniquely determined by dimensionality.

3.1 Emergence of Hubs

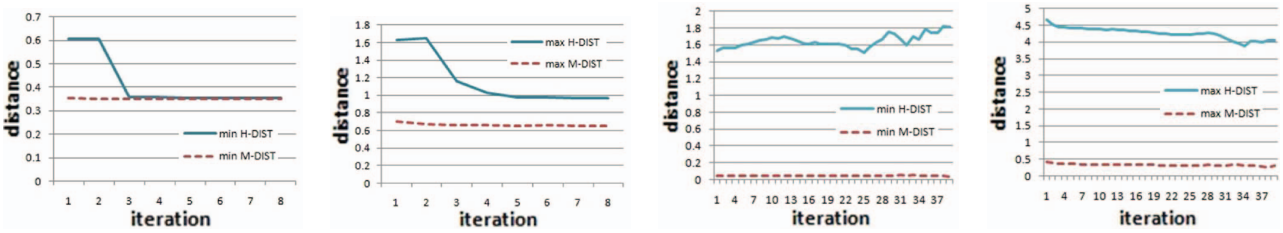
The concentration of distances enables one to view unimodal high-dimensional data as lying approximately on a hypersphere centered at the data distribution mean [23]. However, the variance of distances to the mean remains nonnegligible for any finite number of dimensions [7], [32], which implies that 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 [23]. Such points will have a higher probability of being included in k-nearest-neighbor lists of other points in the data set, which increases their influence, and they emerge as neighbor-hubs.

It was established that hubs also exist in clustered (multimodal) data, tending to be situated in the proximity of cluster centers [23]. 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 pairwise distances in the data [23].

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 (l_2) and Manhattan (l_1) distances, l_p distances with $p > 2$, fractional distances (l_p with rational $p \in (0, 1)$), Bray-Curtis, normalized euclidean, and Canberra distances, cosine similarity, and the dynamic time warping distance for time series [22], [23], [28]. In this vpaper, unless otherwise stated, we will assume the euclidean distance. The methods we propose in Section 4, however, depend mostly on neighborhood relations that are derived from the distance matrix and are, therefore, independent of the particular choice of distance measure.

Before continuing, we should clearly define what constitutes a hub. Similarly to [23], we will say that hubs are points x having $N_k(x)$ more than two standard deviations higher than the expected value k (in other words, significantly above average). However, in most

Authorized licensed use limited to: Universiteit van Tilburg. Downloaded on November 23,2021 at 15:51:37 UTC from IEEE Xplore. Restrictions apply.



(a) Minimal distances, $d = 30$ (b) Maximal distances, $d = 30$ (c) Minimal distances, $d = 2$ (d) Maximal distances, $d = 2$

Fig. 1. Evolution of minimal and maximal distances from cluster centroids to hubs and medoids on synthetic data for neighborhood size 10, and 10 clusters.

experiments that follow, we will only concern ourselves with one major hub in each cluster, i.e., the point with the highest hubness score.

3.2 Relation of Hubs to Data Clusters

There has been previous work on how well high-hubness elements cluster, as well as the general impact of hubness on clustering algorithms [23]. A correlation between low-hubness elements (i.e., *antihubs* or *orphans*) 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 intracluster distance. 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 lead to a decrease in intercluster distance. This has been observed on real data sets clustered using state-of-the-art prototype-based methods, and was identified as a possible area for performance improvement [23]. We will revisit this point in Section 5.4.

It was already mentioned that points closer to cluster means tend to have higher hubness scores than other 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++ [12] multiple times on several randomly generated 10,000-point Gaussian mixtures for various fixed numbers of dimensions (2, 5, 10, 20, 30, 50, 100), observing the high-dimensional case. We measured in each iteration the distance from current cluster centroid to the medoid and to the strongest 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.

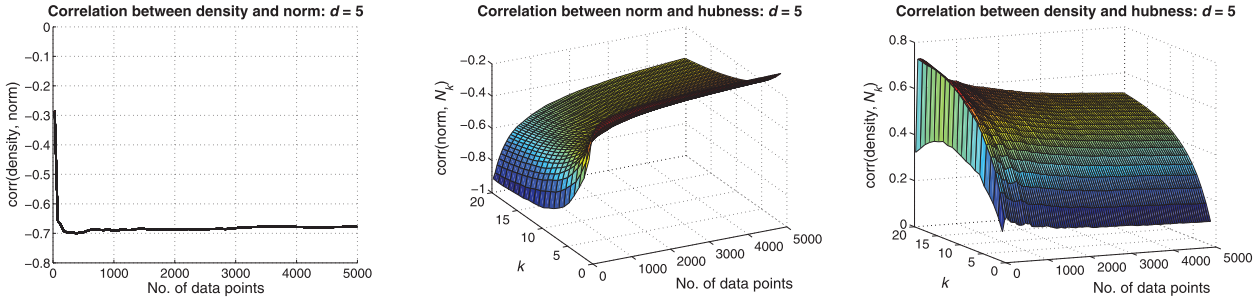
Fig. 1 gives example plots of how these ratios evolve through iterations for the case of 10-cluster data, using neighborhood size 10, with 30 dimensions for the high-dimensional case, and two dimensions to illustrate low-dimensional behavior. The Gaussian mixtures were generated randomly by drawing the centers from a $[l_{bound}, u_{bound}]^d$ uniform distribution (as well as covariance matrices, with somewhat tighter bounds). 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 cluster means and frequent neighbors in the low-dimensional context. This changes with the increase in dimensionality, as 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 drops with iterations, suggesting that as the iterations progress, centroids are becoming closer and closer to data hubs. This already hints at a possibility of developing an iterative approximation procedure.

To complement the above observations and explore the interaction between hubs, medoids, and the classic notion of density, and illustrate the different relationships they exhibit in low- and high-dimensional settings, we performed additional simulations. For a given number of dimensions (5 or 100), we generated a random Gaussian distribution centered around zero and started drawing random points from the distribution one by one, adding them sequentially to a synthetic data set. As the points were being added, hubness, densities, distance contrast, and all the other examined quantities and correlations between them (most of which are shown in Figs. 2 and 3) were calculated on the fly for all the neighborhood sizes within the specified range $\{1, 2, \ldots, 20\}$. The data sets started with 25 points initially and were grown to a size of 5,000. The entire process was repeated 20 times, thus in the end we considered 20 synthetic five-dimensional Gaussian distributions and 20 synthetic 100-dimensional Gaussian distributions. Figs. 2 and 3 display averages taken over all the runs.[1] We report results with euclidean distance, observing similar trends with Manhattan and $l_{0.5}$ distances.

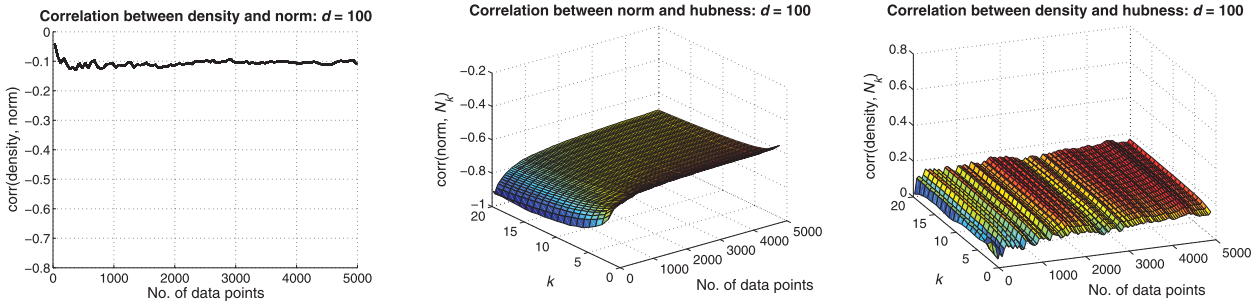
Fig. 2 illustrates the interaction between norm, hubness, and density (as the measurement, not the absolute term) in the simulated setting. From the definition of the setting, the norm of a point can be viewed as an "oracle" that expresses exactly the position of the point with respect to the cluster center.[2] As can be seen in Fig. 2a, strong Pearson correlation between the density measurement and norm indicates that in low dimensions density pinpoints the location of the cluster center with great accuracy. In high dimensions, however,

1. This is the reason why some of the graphs are not smooth.
2. In realistic scenarios, such indicators are not available.

Authorized licensed use limited to: Universiteit van Tilburg. Downloaded on November 23,2021 at 15:51:37 UTC from IEEE Xplore. Restrictions apply.



(a) Correlation between density and norm for $d = 5$ (c) Correlation between norm and hubness for $d = 5$ (e) Correlation between density and hubness for $d = 5$



(b) Correlation between density and norm for $d = 100$ (d) Correlation between norm and hubness for $d = 100$ (f) Correlation between density and hubness for $d = 100$

Fig. 2. Interaction between norm, hubness, and density in the simulated setting, in low- and high-dimensional scenarios.

density loses it connection with centrality (Fig. 2b), and is no longer a good indicator of the main part of the cluster.

Hubness, on the other hand, has some correlation with the norm in low dimensions (Fig. 2c), albeit weak. It is in the high-dimensional setting of Fig. 2d that hubness begins to show its true potential, as the correlation becomes much stronger, meaning that the hubness score of a point represents a very good indicator of its proximity to the cluster center. In both charts, a trend of slight weakening of correlation can be observed as the number of points increases. Meanwhile, strengthening of correlation can be seen for an increasing number of neighbors k, indicating that larger values of k can be used to adjust to larger data set sizes. Quite expectedly, density and hubness are well correlated in low dimensions, but not in the high-dimensional setting (Figs. 2e and 2f).

Fig. 3 shows the interaction between hubs, medoids, and other points in the simulated setting, expressed through distances. Based on the ratio between the average distance to the strongest hub and average distance to the medoid, from Figs. 3a and 3b) it can be seen that in high dimensions the hub is equally informative about the location of the cluster center as the medoid, while in low dimensions the hub and medoid are unrelated. At the same time, generally the hub and the medoid are in neither case the same point, as depicted in Figs. 3c and 3d with the distances from hub to medoid that are always far from 0. This is also indicated in Figs. 3e and 3f that shows the ratio between hub to medoid distance and average pairwise distance. In addition, Fig. 3f suggests that in high dimensions the hub and medoid become relatively closer to each other.

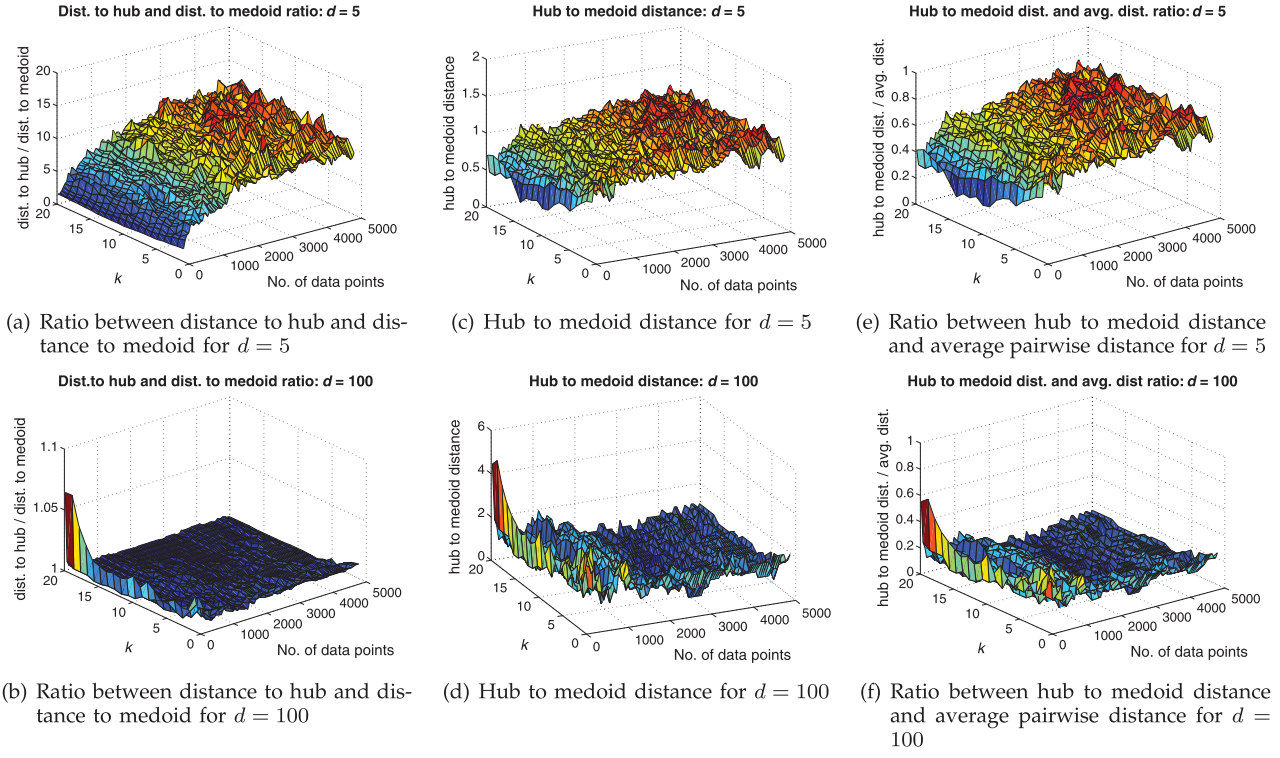
This brings us to the idea that will be explained in detail in the following section: *Why not use hubs as cluster prototypes?* After all, it is expected of points with high hubness scores to be closer to centers of clustered subregions of high-dimensional space than other data points, making them viable candidates for representative cluster elements. We are not limited to observing only points with the highest hubness scores, we can also take advantage of hubness information for any given 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. In addition, hubness of points is straightforward to compute exactly, while the computation cluster centroids and medoids must involve some iterative inexact procedure intrinsically tied to the process of cluster construction. The remaining question of how to assign individual hubs to particular clusters will be addressed in the following section.

4 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 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.

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

Authorized licensed use limited to: Universiteit van Tilburg. Downloaded on November 23,2021 at 15:51:37 UTC from IEEE Xplore. Restrictions apply.



(a) Ratio between distance to hub and distance to medoid for $d = 5$

(c) Hub to medoid distance for $d = 5$

(e) Ratio between hub to medoid distance and average pairwise distance for $d = 5$

(b) Ratio between distance to hub and distance to medoid for $d = 100$

(d) Hub to medoid distance for $d = 100$

(f) Ratio between hub to medoid distance and average pairwise distance for $d = 100$

Fig. 3. Interaction between hubs, medoids, and other points in the simulated setting, expressed through distances, in low- and high-dimensional scenarios.

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 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 lists of elements within the same cluster.[3]

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. There are, however, some important differences. First of all, hubness does not depend on scale. Let D_1 and D_2 be two separate sets of points. If the local distance matrices defined on each of them separately are proportional, we might think of D_1 and D_2 as two copies of the same abstract data model appearing at different scales. Even though the density estimate might be significantly different, depending on the defining volumes which are affected by scale, there will be a perfect match in hubness scores of the corresponding points. However, there is a more subtle difference. Let $D_k(x)$ be the set of points, where x is among the k-nearest neighbors. Hence, the hubness score of x is given by $N_k(x) = |D_k(x)|$. For each $x_i \in D_k(x)$, whether point x is among the k-nearest neighbors of x_i depends on two things: $distance(x, x_i)$, and the density estimate at point x_i, not the density estimate at point x. Consequently, a hub might be a k-neighbor for points where density is high, as well as for points where density is low. Therefore, there is no direct correspondence between the magnitude of hubness and point density. Naturally, since hubs tend to be close to many points, it would be expected that density estimates at hub points are not low, but they may not correspond to the points of highest density in the data. Also, to compute the exact volume of the neighborhood around a given point, one needs to have a suitable data representation. For hubness, one only needs the distance matrix.

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 [33] that it is possible to construct an approximate k-NN graph (from which hubness scores can

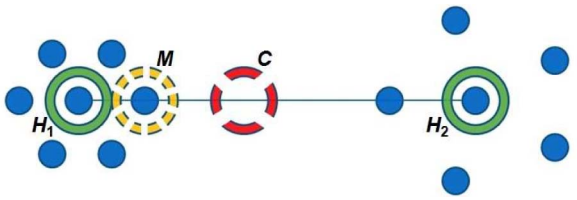


Fig. 4. Illustrative example: The red dashed circle marks the centroid (C), yellow dotted circle the medoid (M), and green circles denote two elements of highest hubness (H_1, H_2), for neighborhood size 3.

3. Henceforth, we will use uppercase K to represent the desired number of clusters and lowercase k for neighborhood size.

Authorized licensed use limited to: Universiteit van Tilburg. Downloaded on November 23,2021 at 15:51:37 UTC from IEEE Xplore. Restrictions apply.

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 [34], as such methods have become quite popular recently. In other words, we expect our algorithms to be applicable in big data scenarios as well.

4.1 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.
```
initializeClusterCenters();
Cluster[] clusters = formClusters();
repeat
  for all Cluster c ∈ clusters do
    DataPoint h = findClusterHub(c);
    setClusterCenter(c, h);
  end for
  clusters = formClusters();
until noReassignments
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, we resorted 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.

4.2 Probabilistic Approach

Even though points with highest hubness scores are without doubt the prime candidates for cluster centers, there is no need to disregard the information about hubness scores of other points in the data. In the algorithm described below, we implemented a squared hubness-proportional stochastic scheme based on the widely used simulated annealing approach to optimization [35]. The temperature factor was introduced to the algorithm, so that it may start as being entirely probabilistic and eventually end by executing deterministic K-hubs iterations. We will refer to this algorithm, specified by Algorithm 2, as *hubness-proportional clustering* (HPC).

Algorithm 2. HPC.
```
initializeClusterCenters();
Cluster[] clusters = formClusters();
float t = t0; initialize temperature
repeat
  float θ = getProbFromSchedule(t);
  for all Cluster c ∈ clusters do
    if randomFloat(0,1) < θ then
      DataPoint h = findClusterHub(c);
      setClusterCenter(c, h);
    else
      for all DataPoint x ∈ c do
        setChoosingProbability(x, N_k^2(x));
      end for
      normalizeProbabilities();
      DataPoint h = chooseHubProbabilistically(c);
      setClusterCenter(c, h);
    end if
  end for
  clusters = formClusters();
  t = updateTemperature(t);
until noReassignments
return clusters
```

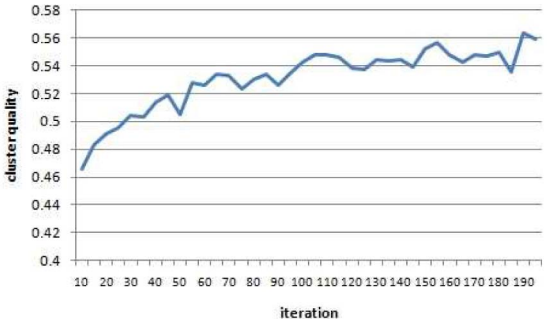


Fig. 5. Estimated quality of clustering for various durations of probabilistic search in HPC.

The reason why hubness-proportional clustering is feasible in the context of high dimensionality lies in the skewness of the distribution of k-occurrences. Namely, there exist many data points having low hubness scores, making them bad candidates for cluster centers. Such points will have a low probability of being selected. To further emphasize this, we use the square of the actual hubness score instead of making the probabilities directly proportional to $N_k(x)$.

We have chosen to use a rather trivial temperature schedule in the getProbFromSchedule(t) function. The number of probabilistic iterations N_{Prob} is passed as an argument to the algorithm and the probability $\theta = \min(1, t/N_{Prob})$. Different probabilistic schemes are possible and might even lead to better results.

The HPC algorithm defines a search through the data space based on hubness as a kind of a local centrality estimate. To justify the use of the proposed stochastic scheme, we executed a series of initial tests on a synthetic mixture of Gaussians, for dimensionality $d = 50, n = 10,000$ instances, and $K = 25$ clusters in the data. Neighborhood size was set to $k = 10$ and for each preset number of probabilistic iterations in the annealing schedule, the clustering was run 50 times, each time reinitializing the seeds. The results are displayed in Fig. 5. The silhouette index [36] was used to estimate the clustering quality. Due to the significant skewness of the squared hubness scores, adding more probabilistic iterations helps in achieving better clustering, up to a certain plateau that is eventually reached. The same shape of the curve also appears in the case of not taking the last, but the error-minimizing configuration.

Authorized licensed use limited to: Universiteit van Tilburg. Downloaded on November 23,2021 at 15:51:37 UTC from IEEE Xplore. Restrictions apply.

4.3 A Hybrid Approach

The algorithms outlined in Sections 4.1 and 4.2 share a property that they do not require knowledge of data/object representation (they work the with distance matrix only), so all that is required is a distance/similarity measure defined for each pair of data objects. However, if the representation is also available such that it is possible to meaningfully calculate centroids, there also exists a third alternative: use point hubness scores to guide the search, but choose a centroid-based cluster configuration in the end. We will refer to this algorithm as *hubness-proportional K-means* (HPKM). It is nearly identical to HPC, the only difference being in the deterministic phase of the iteration, as the configuration cools down during the annealing procedure: instead of reverting to K-hubs, the deterministic phase executes K-means updates.

Algorithm 3. HPKM.

```
initializeClusterCenters();
Cluster[] clusters = formClusters();
float t = t_0; {initialize temperature}
repeat
  float θ = getProbFromSchedule(t);
  for all Cluster c ∈ clusters do
    if randomFloat(0, 1) < θ then
      DataPoint h = findClusterCentroid(c);
      setClusterCenter(c, h);
    else
      for all DataPoint x ∈ c do
        setChoosingProbability(x, N_k^2(x));
      end for
      normalizeProbabilities();
      DataPoint h = chooseHubProbabilistically(c);
      setClusterCenter(c, h);
    end if
  end for
  clusters = formClusters();
  t = updateTemperature(t);
until noReassignments
return clusters
```

There are, indeed, cases when HPKM might be preferable to the pure hubness-based approach of K-hubs and HPC. Even though our initial experiments (Fig. 3) suggest that the major hubs lie close to local cluster means in high-dimensional data, there is no guarantee that this would hold for every cluster in every possible data set. It is reasonable to expect there to be distributions which lead to such local data structure where the major hub is not among the most central points. Also, an *ideal* cluster configuration (with minimal error) on a given real-world data set is sometimes impossible to achieve by using points as centers, since centers may need to be located in the empty space between the points.

In fact, we opted for the hybrid approach only after observing that, despite the encouraging initial results on synthetic data discussed in Section 5.1, hubness-based algorithms were not consistently better on real-world data sets. This is why we tried to take "the best of both worlds," by combining the centroid-based cluster representation with the hubness-guided search. This way, we are hoping to avoid premature convergence to a local optimum. We must keep in mind, however, that it is not as widely applicable as K-hubs and HPC, since it only makes sense with data where centroids can actually be defined.

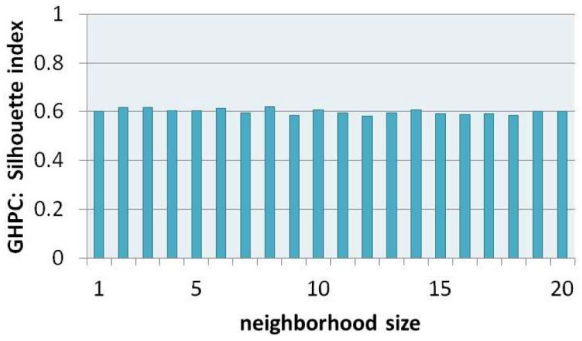


Fig. 6. Sensitivity of the quality of GHPC clustering on neighborhood size (k), measured by silhouette index.

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 [12].[4] The neighborhood size of $k = 10$ 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 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, $k \in \{1, 2, \ldots, 20\}$. 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 [13] and one standard density-based method, GDBScan [37]. 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 nonhyperspherical clusters. Yet, the hubness-based methods could just as easily be "kernelized," pretty much the same way it was done for

4. Hubness could also be used for cluster initialization, an option which we have not fully explored yet.

Authorized licensed use limited to: Universiteit van Tilburg. Downloaded on November 23,2021 at 15:51:37 UTC from IEEE Xplore. Restrictions apply.

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

		LKH	GKH	LHPC	GHPC	KM++
$K = 5$	Silhouette	0.46 ± 0.03	0.51 ± 0.02	$\mathbf{0.61} \pm 0.02$	$\mathbf{0.61} \pm 0.02$	0.56 ± 0.02
	Entropy	0.32 ± 0.04	0.17 ± 0.01	0.09 ± 0.02	$\mathbf{0.06} \pm 0.01$	0.10 ± 0.01
	Perfect	0.32 ± 0.05	0.39 ± 0.05	$\mathbf{0.75} \pm 0.07$	$\mathbf{0.76} \pm 0.06$	0.54 ± 0.04
$K = 10$	Silhouette	0.38 ± 0.02	0.46 ± 0.01	0.52 ± 0.02	$\mathbf{0.57} \pm 0.01$	0.52 ± 0.01
	Entropy	0.52 ± 0.07	0.22 ± 0.01	0.22 ± 0.03	$\mathbf{0.08} \pm 0.01$	0.13 ± 0.01
	Perfect	0.05 ± 0.01	0.06 ± 0.02	0.30 ± 0.05	$\mathbf{0.39} \pm 0.06$	0.11 ± 0.02

K-means. This idea requires further tests and is beyond the scope of this paper.

For evaluation, we used repeated random subsampling, 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 overfitting, 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.

5.1 Synthetic Data: Gaussian Mixtures

In the first batch of experiments, we wanted to compare the value of *global* versus *local* hubness scores. These initial tests were run on synthetic data and do not include HPKM, as the hybrid approach was introduced later for tackling problems on real-world data.

For comparing the resulting clustering quality, we used mainly the silhouette index as an unsupervised measure of configuration validity, and average cluster entropy as a supervised measure of clustering homogeneity. Since most of the generated data sets are "solvable," i.e., consist of nonoverlapping Gaussian distributions, we also report the normalized frequency with which the algorithms were able to find these perfect configurations. We ran two lines 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 hyperspherical (diagonal covariance matrices, independent attributes). Distribution means were drawn randomly from $[l^m_{bound}, u^m_{bound}]^d$, $l^m_{bound} = -20, u^m_{bound} = 20$ and the standard deviations were also uniformly taken from $[l^\sigma_{bound}, u^\sigma_{bound}]^d$, $l^\sigma_{bound} = 2, u^\sigma_{bound} = 5$.

Table 1 shows the final summary of all these runs. (Henceforth, we use boldface to denote measurements that are significantly better than others, in the sense of having no overlap of surrounding one-standard deviation intervals.) Global hubness is definitely to be preferred, especially in the presence of more clusters, which further restrict neighbor sets in the case of local hubness scores. 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 basis, 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.

5.2 Clustering and High Noise Levels

Real-world data often contain noisy or erroneous values due to the nature of the data-collecting process. It can be assumed that hub-based algorithms will be more robust with respect to noise, since hubness-proportional search is driven mostly by the highest-hubness elements, 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 technique may be applied. In setups involving high levels of noise, this may not be the case.

To test this hypothesis, we generated two data sets of 10,000 instances as a mixture of 20 clearly separated Gaussians, farther away from each other than in the previously described experiments. The first data set was 10 dimensional and the second 50 dimensional. In both cases, individual distribution centers were drawn independently from the uniform $[l^m_{bound}, u^m_{bound}]^d$ distribution, $l^m_{bound} = -150, u^m_{bound} = 150$. The covariance matrix was also random generated, independently for each distribution. It was diagonal, the individual feature standard deviations drawn uniformly from $[l^\sigma_{bound}, u^\sigma_{bound}]^d$, $l^\sigma_{bound} = 10, u^\sigma_{bound} = 60$. Cluster sizes were imbalanced. Without noise, both of these data sets represented quite easy clustering problems, all of the algorithms being able to solve them very effectively. This is, regardless, a more challenging task than we had previously addressed [38], by virtue of having a larger number of clusters.

To this data we incrementally added noise, 250 instances at a time, drawn from a uniform distribution on hypercube $[l^n_{bound}, u^n_{bound}]^d$, $l^n_{bound} = -200, u^n_{bound} = 200$, containing all the data points. The hypercube was much larger than the space containing the rest of the points. In other words, clusters were immersed in uniform noise. The highest level of noise for which we tested was the case when there was an equal number of actual data instances in original clusters and

Authorized licensed use limited to: Universiteit van Tilburg. Downloaded on November 23,2021 at 15:51:37 UTC from IEEE Xplore. Restrictions apply.

TABLE 2
Estimated Cluster Quality at Various Noise Levels for Synthetic Data Composed of 20 Different Clusters

(a) d=10, k=20

	GKH		GHPC		KM++		GHPKM	
	Sil.	Ent.	Sil.	Ent.	Sil.	Ent.	Sil.	Ent.
Noise<10%	0.29	0.41	0.37	0.18	0.34	0.22	**0.38**	**0.10**
Noise 10-20%	0.31	0.44	0.38	0.27	0.36	0.28	**0.39**	**0.20**
Noise 20-30%	0.31	0.50	0.35	0.33	0.36	0.38	**0.39**	**0.27**
Noise 30-40%	0.29	0.52	0.36	**0.32**	0.35	0.44	**0.37**	0.36
Noise 40-50%	0.29	0.55	0.35	**0.38**	0.33	0.53	**0.36**	0.45
AVG	0.30	0.50	0.36	**0.31**	0.34	0.41	**0.37**	**0.31**

(b) d=10, k=50

	GKH		GHPC		KM++		GHPKM	
	Sil.	Ent.	Sil.	Ent.	Sil.	Ent.	Sil.	Ent.
Noise<10%	0.29	0.44	0.35	0.18	0.33	0.23	**0.39**	**0.10**
Noise 10-20%	0.29	0.50	0.36	0.25	0.36	0.27	**0.39**	**0.15**
Noise 20-30%	0.30	0.53	0.35	0.32	0.36	0.35	**0.38**	**0.24**
Noise 30-40%	0.29	0.59	0.35	0.35	0.35	0.44	**0.38**	**0.32**
Noise 40-50%	0.29	0.60	0.33	**0.39**	0.33	0.50	**0.36**	0.43
AVG	0.29	0.55	0.35	0.33	0.34	0.39	**0.38**	**0.29**

(c) d=50, k=20

	GKH		GHPC		KM++		GHPKM	
	Sil.	Ent.	Sil.	Ent.	Sil.	Ent.	Sil.	Ent.
Noise<10%	0.37	0.45	0.49	0.12	0.48	0.16	**0.55**	**0.03**
Noise 10-20%	0.38	0.54	0.50	0.20	0.46	0.30	**0.55**	**0.02**
Noise 20-30%	0.37	0.54	0.47	0.23	0.42	0.44	**0.55**	**0.04**
Noise 30-40%	0.36	0.58	0.46	0.28	0.40	0.54	**0.53**	**0.09**
Noise 40-50%	0.34	0.64	0.43	0.40	0.38	0.59	**0.51**	**0.17**
AVG	0.36	0.57	0.46	0.27	0.42	0.46	**0.53**	**0.09**

(d) d=50, k=50

	GKH		GHPC		KM++		GHPKM	
	Sil.	Ent.	Sil.	Ent.	Sil.	Ent.	Sil.	Ent.
Noise<10%	0.37	0.36	0.51	0.05	0.48	0.18	**0.55**	**0.02**
Noise 10-20%	0.40	0.39	0.51	0.09	0.46	0.33	**0.56**	**0.02**
Noise 20-30%	0.36	0.44	0.49	0.14	0.43	0.42	**0.55**	**0.03**
Noise 30-40%	0.36	0.45	0.47	0.20	0.42	0.52	**0.54**	**0.10**
Noise 40-50%	0.33	0.46	0.45	0.25	0.41	0.57	**0.52**	**0.17**
AVG	0.36	0.43	0.48	0.17	0.43	0.44	**0.54**	**0.09**

noisy instances. At every noise level, KM++, GKH, GHPC, and Global Hubness-Proportional K-Means (GHPKM) were run 50 times each. We used two different k-values, namely 20 and 50. We have used somewhat larger neighborhoods to try to smooth out the influence of noisy data on hubness scores. The silhouette index and average entropy were computed only on the nonnoisy restriction of the data, i.e., the original Gaussian clusters. This is an important point, as such measures quantify how well each algorithm captures *the underlying structure* of the data. Indeed, if there is noise in the data, we are not overly interested in how well the noisy points cluster. Including them into the cluster quality indices might be misleading.

A brief summary of total averages is given in Table 2, with the best Silhouette index value and the best entropy score in each row given in boldface. The probabilistic hub-based algorithms show substantial improvements with higher noise levels, which is a very useful property. GHPKM is consistently better than KM++ for all noise levels, especially in terms of cluster homogeneity. The difference in average cluster entropy is quite obvious in all cases and is more pronounced in the 50-dimensional case, where there is more hubness in the data.

Fig. 7 shows the rate of change in algorithm performance under various noise levels. We see that the achieved improvement is indeed stable and consistent, especially in the high-dimensional case. The difference increases with increasing noise, which means that HPC and HPKM are not only less affected by the curse of dimensionality, but also more robust to the presence of noise in the data.

5.3 Experiments on Real-World Data

Real-world data are usually much more complex and difficult to cluster, therefore such tests are of a higher practical significance. As not all data exhibit hubness, we tested the algorithms both on intrinsically high-dimensional, high-hubness data and intrinsically low-to-medium dimensional, low-hubness data. There were two different experimental setups. In the first setup, a single data set was clustered for many different K-s (number of clusters), to see if there is any difference when the number of clusters is varied. In the second setup, 20 different data sets were all clustered by the number of classes in the data (the number of different labels).

The clustering quality in these experiments was measured by two quality indices, the silhouette index and the isolation index [39], which measures a percentage of k-neighbor points that are clustered together.

In the first experimental setup, the two-part Miss-America data set (cs.joensuu.fi/sipu/datasets/) was used for evaluation. Each part consists of 6,480 instances having 16 dimensions. Results were compared for various predefined numbers of clusters in algorithm calls. Each algorithm was tested 50 times for each number of clusters. Neighborhood size was 5.

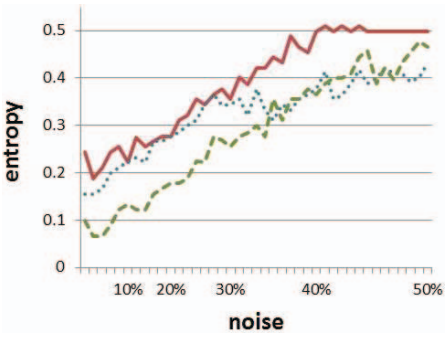
The results for both parts of the data set are given in Table 3. GHPC clearly outperformed KM and other hubness-based methods. This shows that hubs can serve as good cluster center prototypes. On the other hand, hyperspherical methods have their limits and kernel K-means achieved the best overall cluster quality on this data set. Only one quality estimate is given for GDBScan, as it automatically determines the number of clusters on its own.

As mostly low-to-medium hubness data (with the exception of spambase), we have taken several UCI data sets (archive.ics.uci.edu/ml/datasets.html). Values of all the individual features in the data sets were normalized prior to testing. The data sets were mostly simple, composed only of a few clusters. The value of k was set to 20. The results are shown in the first parts of Tables 4a and 4b.[5] In the absence of hubness,[6] purely hubness-based

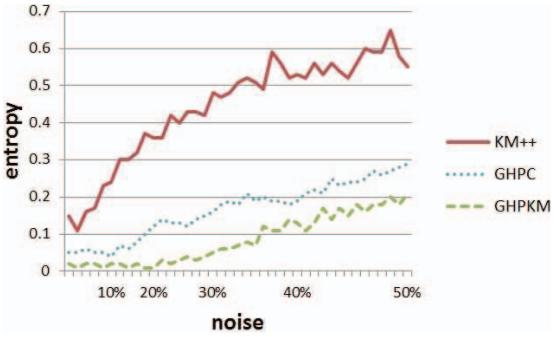
5. Some entries for GDBScan are marked as "−" and in those cases the standard parametrization of the algorithm produced a single connected cluster as a result. Due to space considerations, we show only averages for the isolation index in Table 4b.

6. We quantify hubness using the skewness measure, i.e., the standardized third moment of the distribution of N_k, signified as S_{N_k}. If $S_{N_k} = 0$ there is no skewness, positive (negative) values signify skewness to the right (left).

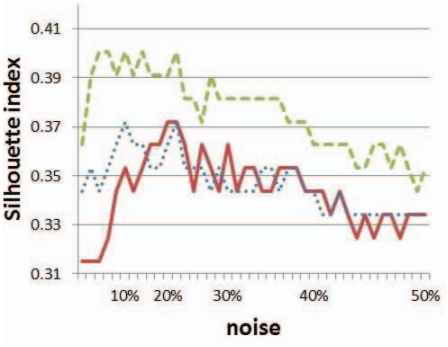
Authorized licensed use limited to: Universiteit van Tilburg. Downloaded on November 23,2021 at 15:51:37 UTC from IEEE Xplore. Restrictions apply.



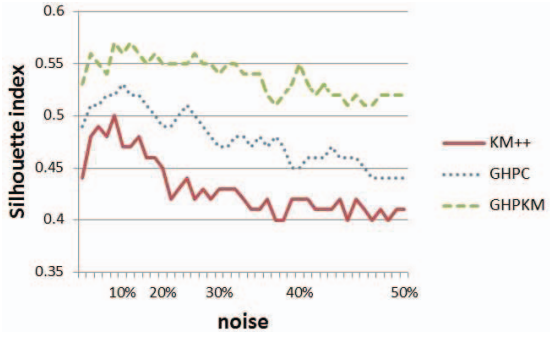
(a) Cluster entropies for $d = 10$, $k = 50$



(b) Cluster entropies for $d = 50$, $k = 50$



(c) Silhouette index values for $d = 10$, $k = 50$



(d) Silhouette index values for $d = 50$, $k = 50$

Fig. 7. Gradual change in cluster quality measures with rising noise levels. The difference between the algorithm performances is much more pronounced in the high-dimensional case.

methods do not perform well. Note, however, that they score comparably to KM++ on several data sets, and that GHPC did as well as KM++ on the Iris data set, which is only four dimensional. On the other hand, hubness-guiding the K-means in HPKM neither helps nor hurts the K-means base in such cases.

TABLE 3
Clustering Quality on the Miss-America Data Set

(a) Silhouette index

	K	2	4	6	8	10	12	14	16
Part I	GKH	0.28	0.14	0.12	0.08	0.07	0.05	0.06	0.05
	GHPC	**0.38**	0.29	0.25	0.21	0.15	0.10	0.10	0.09
	KM++	0.14	0.12	0.09	0.08	0.07	0.07	0.07	0.07
	GHPKM	0.28	0.18	0.17	0.14	0.13	0.11	0.10	0.08
	ker-KM++	0.33	**0.36**	**0.36**	**0.34**	**0.35**	**0.22**	**0.28**	**0.14**
	GDBScan	-0.27							
Part II	GKH	0.33	0.21	0.13	0.08	0.08	0.07	0.06	0.06
	GHPC	0.33	0.27	0.22	0.26	0.18	0.19	0.12	0.11
	KM++	0.18	0.12	0.10	0.08	0.07	0.08	0.07	0.07
	GHPKM	0.33	0.22	0.18	0.14	0.12	0.11	0.10	0.08
	ker-KM++	**0.46**	**0.30**	**0.41**	**0.46**	**0.29**	**0.28**	**0.24**	**0.23**
	GDBScan	-0.25							

(b) Isolation index

	K	2	4	6	8	10	12	14	16
Part I	GKH	0.83	0.58	0.53	0.38	0.27	0.22	0.21	0.15
	GHPC	**0.91**	0.89	0.71	0.53	0.42	0.33	0.30	0.26
	KM++	0.62	0.46	0.34	0.23	0.19	0.16	0.13	0.12
	GHPKM	0.85	0.54	0.45	0.38	0.29	0.26	0.24	0.23
	ker-KM++	0.77	**0.92**	**0.93**	**0.92**	**0.95**	**0.91**	**0.91**	**0.80**
	GDBScan	0.12							
Part II	GKH	0.82	0.56	0.35	0.26	0.21	0.17	0.15	0.14
	GHPC	0.80	0.64	0.45	0.48	0.37	0.35	0.26	0.23
	KM++	0.62	0.35	0.28	0.20	0.16	0.14	0.11	0.09
	GHPKM	0.77	0.50	0.36	0.29	0.26	0.24	0.22	0.19
	ker-KM++	**0.88**	**0.78**	**0.90**	**0.94**	**0.91**	**0.89**	**0.90**	**0.91**
	GDBScan	0.12							

As intrinsically high-dimensional, high-hubness data, we have taken several subsets of the ImageNet public repository (www.image-net.org). These data sets are described in detail in [20], [25]. We examine two separate cases: Haar wavelet representation and SIFT codebook + color histogram representation [40], [41]. This totals to 10 different clustering problems. We set k to 5. The results are given in the second parts of Tables 4a and 4b.

We see that the Haar wavelet representation clusters well, while the SIFT + color histogram one does not. This is not a general conclusion, but rather a particular feature of the observed data. GHPKM is clearly the best among the evaluated algorithms in clustering the Haar representation of the images. This is encouraging, as it suggests that hubness-guided clustering may indeed be useful in some real-world high-dimensional scenarios.

The fact that kernel K-means achieves best isolation in most data sets suggests that accurate center localization is not in itself enough for ensuring good clustering quality and the possibilities for extending the basic HPKM and HPC framework to allow for nonhyperspherical and arbitrarily shaped clusters need to be considered. There are many ways to use hubs and hubness in high-dimensional data clustering. We have only considered the simplest approach here and many more remain to be explored.

5.4 Interpreting Improvements in Silhouette Index

This section will discuss the reason why hubness-based clustering can offer better performance when compared to

Authorized licensed use limited to: Universiteit van Tilburg. Downloaded on November 23,2021 at 15:51:37 UTC from IEEE Xplore. Restrictions apply.

TABLE 4
Clustering Quality on Low to Medium-Hubness Data Sets from the UCI Repository and Subsets of High-Hubness ImageNet Data

(a) Silhouette index

data set	size	d	K	S_{N_1}	GKH	GHPC	KM++	GHPKM	ker-KM	GDB-Scan
wpbc	198	33	2	0.64	0.16	0.16	0.16	0.16	0.17	-
spamb.	4601	57	2	21.46	0.29	**0.38**	0.31	**0.50**	0.13	0.01
arcene	100	1000	2	1.08	0.21	0.22	0.20	**0.23**	0.21	-
ovarian	253	15154	2	1.20	0.17	0.17	0.18	0.19	0.13	-
iris	158	4	3	0.46	0.48	0.47	0.49	0.49	0.38	**0.62**
parkins.	195	22	2	0.39	0.25	0.30	0.37	0.37	**0.45**	-
sonar	208	60	2	1.35	0.11	0.11	**0.19**	0.15	0.13	-
wine	178	13	3	0.76	0.27	0.33	0.34	0.35	0.12	-
abalone	4177	8	29	0.92	0.22	0.20	0.26	0.27	0.26	0.05
spectr.	531	100	10	1.20	0.16	0.16	0.23	**0.25**	0.15	0.12
AVG-UCI					0.23	0.25	0.27	**0.30**	0.21	0.20
ds3haar	2731	100	3	2.27	0.62	0.67	**0.70**	**0.70**	0.61	0.63
ds4haar	6054	100	4	2.44	0.53	0.59	0.62	**0.64**	0.52	0.56
ds5haar	6555	100	5	2.43	0.56	0.58	0.65	**0.69**	0.50	0.51
ds6haar	6010	100	6	2.13	0.49	0.55	0.56	**0.58**	0.48	0.50
ds7haar	10544	100	7	4.60	0.33	0.65	0.63	**0.68**	0.50	0.58
AVG-Haar					0.51	0.61	0.63	**0.66**	0.52	0.55
ds3sift	2731	416	3	15.85	0.08	**0.12**	0.05	0.05	0.05	**0.12**
ds4sift	6054	416	4	8.88	0.06	0.06	0.02	0.03	0.02	**0.18**
ds5sift	6555	416	5	26.08	0.05	0.06	0.01	0.02	0.09	**0.11**
ds6sift	6010	416	6	13.19	0.01	0.02	0.01	0.02	**0.11**	0.09
ds7sift	10544	416	7	9.15	0.04	0.05	0.01	0.03	**0.19**	0.16
AVG-Sift					0.05	0.06	0.02	0.03	0.09	**0.13**
AVG-Img					0.28	0.34	0.33	**0.35**	0.31	0.34
AVG-Total					0.26	0.30	0.30	**0.33**	0.26	0.27

(b) Isolation index

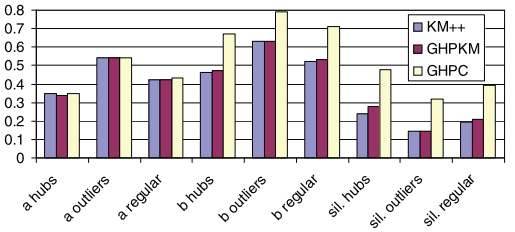
data sets	GKH	GHPC	KM++	GHPKM	ker-KM	DB-Scan
AVG-UCI	0.48	0.47	0.44	0.47	**0.64**	0.55
AVG-Haar	0.64	0.69	0.71	**0.73**	0.70	0.72
AVG-Sift	0.35	0.38	0.37	0.41	**0.79**	0.32
AVG-Img	0.50	0.54	0.54	0.57	**0.76**	0.52
AVG-Total	0.49	0.51	0.49	0.52	**0.70**	0.54

K-means in terms of intra- and intercluster distance expressed by the silhouette index.

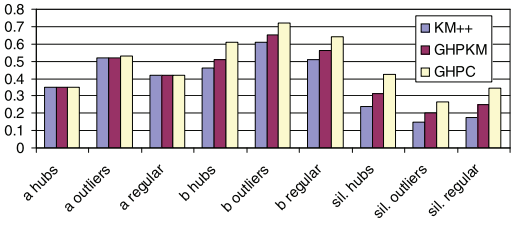
Let us view the a (intra) and b (inter) components of the silhouette index separately, and compute a, b, and the silhouette index on a given data set for hubs, outliers and "regular" points.[7] Let n_h be the number of hubs selected. Next, we select as outliers the n_h points with the lowest k-occurrences. Finally, we select all remaining points as "regular" points.

Fig. 8 illustrates the described break-up of the silhouette index on the Miss-America data set (we have detected similar trends with all other data sets where hubness-based methods offer improvement), for $k = 5$ and $K = 2$. It can be seen that all clustering methods perform approximately

7. For the ith point, a_i is the average distance to all points in its cluster (intra-cluster distance), and b_i the minimum average distance to points from other clusters (inter-cluster distance). The silhouette index of the ith point is then $(b_i - a_i)/\max(a_i, b_i)$, ranging between -1 and 1 (higher values are better). The silhouette index of a set of points is obtained by averaging the silhouette indices of the individual points.



(a) Miss-America, Part I



(b) Miss-America, Part II

Fig. 8. Break-up od the silhouette index into its constituent parts, viewed separately for hubs, outliers, and regular points on the Miss-America data set.

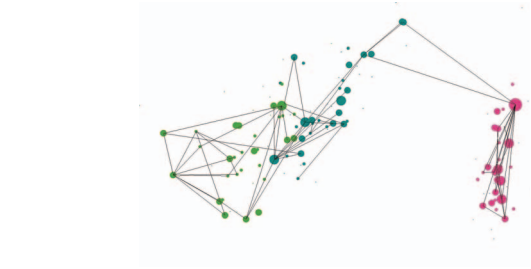
equally with respect to the a (intra) part, but that the hubness-based algorithms increase the b (inter) part, which is the main reason for improving the silhouette index. The increase of b is visible in all three groups of points, but is most prominent for hubs. Earlier research [23] had revealed that hubs often have low b-values, which causes them to cluster badly and have a negative impact on the clustering process. It was suggested that they should be treated almost as outliers. That is why it is encouraging to see that the proposed clustering methods lead to clustering configurations, where hubs have higher b-values than in the case of K-means.

5.5 Visualizing the Hubness-Guided Search

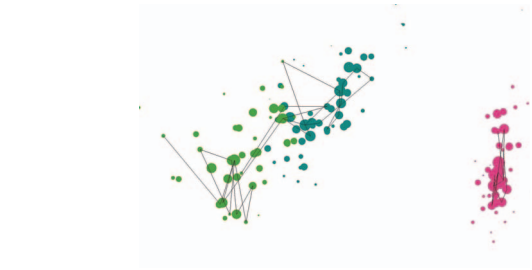
To gain further insight, we have visualized the hubness-guided search on several low-to-medium-dimensional data sets. We performed clustering by the HPC algorithm and recorded the history of all iteration states (visited hub-points). After the clustering was completed, the data were projected onto a plane by a standard multidimensional scaling (MDS) procedure. Each point was drawn as a circle of radius proportional to its relative hubness. Some of the resulting images generated for the well-known Iris data set are shown in Fig. 9.

It can be seen that HPC searches 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.

Authorized licensed use limited to: Universiteit van Tilburg. Downloaded on November 23,2021 at 15:51:37 UTC from IEEE Xplore. Restrictions apply.



(a) k=1



(b) k=10

Fig. 9. Hubness-guided search for the best cluster hub-configuration in HPC on Iris data.

6 Conclusions and Future Work

Using hubness for data clustering has not previously been attempted. We have shown that using hubs to approximate local data centers is not only a feasible option, but also frequently leads to improvement over the centroid-based approach. The proposed GHPKM method had proven to be more robust than the K-Means++ baseline on both synthetic and real-world data, as well as in the presence of high levels of artificially introduced noise. This initial evaluation suggests that using hubs both as cluster prototypes and points guiding the centroid-based search is a promising new idea in clustering high-dimensional and noisy data. Also, global hubness estimates are generally to be preferred with respect to the local ones.

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 intercluster distance, i.e., better cluster separation.

The proposed 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 proposed methods—detecting only hyperspherical 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.

Acknowledgments

This work was supported by the Slovenian Research Agency, the IST Programme of the EC under PASCAL2 (IST-NoE-216886), and the Serbian Ministry of Education, Science and Technological Development project no. OI174023.

References

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

[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. Kröger, 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. Kröger, "Density-Connected Subspace Clustering for High-Dimensional Data," *Proc. Fourth SIAM Int'l Conf. Data Mining (SDM)*, pp. 246-257, 2004.

[5] E. Müller, S. Günnemann, I. Assent, and T. Seidl, "Evaluating Clustering in Subspace Projections of High Dimensional Data," *Proc. VLDB Endowment*, vol. 2, pp. 1270-1281, 2009.

[6] 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.

[7] D. François, 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.

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

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

[10] E. Agirre, D. Martínez, 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.

[11] K. Ning, H. Ng, S. Srihari, H. Leong, and A. Nesvizhskii, "Examination of the Relationship between Essential Genes in PPI Network and Hub Proteins in Reverse Nearest Neighbor Topology," *BMC Bioinformatics*, vol. 11, pp. 1-14, 2010.

[12] D. Arthur and S. Vassilvitskii, "K-Means++: The Advantages of Careful Seeding," *Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA)*, pp. 1027-1035, 2007.

[13] I.S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-Means: Spectral Clustering and Normalized Cuts," *Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pp. 551-556, 2004.

[14] T.N. Tran, R. Wehrens, and L.M.C. Buydens, "Knn Density-Based Clustering for High Dimensional Multispectral Images," *Proc. Second GRSS/ISPRS Joint Workshop Remote Sensing and Data Fusion over Urban Areas*, pp. 147-151, 2003.

[15] E. Biçici and D. Yuret, "Locally Scaled Density Based Clustering," *Proc. Eighth Int'l Conf. Adaptive and Natural Computing Algorithms (ICANNGA), Part I*, pp. 739-748, 2007.

[16] C. Zhang, X. Zhang, M.Q. Zhang, and Y. Li, "Neighbor Number, Valley Seeking and Clustering," *Pattern Recognition Letters*, vol. 28, no. 2, pp. 173-180, 2007.

[17] S. Hader and F.A. Hamprecht, "Efficient Density Clustering Using Basin Spanning Trees," *Proc. 26th Ann. Conf. Gesellschaft für Klassifikation*, pp. 39-48, 2003.

[18] C. Ding and X. He, "K-Nearest-Neighbor Consistency in Data Clustering: Incorporating Local Information into Global Optimization," *Proc. ACM Symp. Applied Computing (SAC)*, pp. 584-589, 2004.

[19] C.-T. Chang, J.Z.C. Lai, and M.D. Jeng, "Fast Agglomerative Clustering Using Information of k-Nearest Neighbors," *Pattern Recognition*, vol. 43, no. 12, pp. 3958-3968, 2010.

[20] N. Tomašev, M. Radovanović, D. Mladenić, and M. Ivanović, "Hubness-Based Fuzzy Measures for High-Dimensional k-Nearest Neighbor Classification," *Proc. Seventh Int'l Conf. Machine Learning and Data Mining (MLDM)*, pp. 16-30, 2011.

[21] N. Tomašev, M. Radovanović, D. Mladenić, and M. Ivanović, "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.

[22] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Time-Series Classification in Many Intrinsic Dimensions," *Proc. 10th SIAM Int'l Conf. Data Mining (SDM)*, pp. 677-688, 2010.

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

Authorized licensed use limited to: Universiteit van Tilburg. Downloaded on November 23,2021 at 15:51:37 UTC from IEEE Xplore. Restrictions apply.

[24] N. Tomašev and D. Mladenić, "Nearest Neighbor Voting in High Dimensional Data: Learning from Past Occurrences," *Computer Science and Information Systems,* vol. 9, no. 2, pp. 691-712, 2012.
[25] N. Tomašev, R. Brehar, D. Mladenić, 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.
[26] K. Buza, A. Nanopoulos, and L. Schmidt-Thieme, "INSIGHT: Efficient and Effective Instance Selection for Time-Series Classification," *Proc. 15th Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD), Part II,* pp. 149-160, 2011.
[27] A. Nanopoulos, M. Radovanović, and M. Ivanović, "How Does High Dimensionality Affect Collaborative Filtering?" *Proc. Third ACM Conf. Recommender Systems (RecSys),* pp. 293-296, 2009.
[28] M. Radovanović, A. Nanopoulos, and M. Ivanović, "On the Existence of Obstinate Results in Vector Space Models," *Proc. 33rd Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 186-193, 2010.
[29] J.J. Aucouturier and F. Pachet, "Improving Timbre Similarity: How High Is the Sky?" *J. Negative Results in Speech and Audio Sciences,* vol. 1, 2004.
[30] J.J. Aucouturier, "Ten Experiments on the Modelling of Polyphonic Timbre," PhD dissertation, Univ. of Paris 6, 2006.
[31] D. Schnitzer, A. Flexer, M. Schedl, and G. Widmer, "Local and Global Scaling Reduce Hubs in Space," *J. Machine Learning Research,* vol. 13, pp. 2871-2902, 2012.
[32] S. France and D. Carroll, "Is the Distance Compression Effect Overstated? Some Theory and Experimentation," *Proc. Sixth Int'l Conf. Machine Learning and Data Mining in Pattern Recognition (MLDM),* pp. 280-294, 2009.
[33] J. Chen, H. Fang, and Y. Saad, "Fast Approximate *k*NN Graph Construction for High Dimensional Data via Recursive Lanczos Bisection," *J. Machine Learning Research,* vol. 10, pp. 1989-2012, 2009.
[34] V. Satuluri and S. Parthasarathy, "Bayesian Locality Sensitive Hashing for Fast Similarity Search," *Proc. VLDB Endowment,* vol. 5, no. 5, pp. 430-441, 2012.
[35] D. Corne, M. Dorigo, and F. Glover, *New Ideas in Optimization.* McGraw-Hill, 1999.
[36] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining.* Addison Wesley, 2005.
[37] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-Based Clustering in Spatial Databases: The Algorithm Gdbscan and Its Applications," *Data Mining and Knowledge Discovery,* vol. 2, no. 2, pp. 169-194, 1998.
[38] N. Tomašev, M. Radovanović, D. Mladenić, and M. Ivanović, "The Role of Hubness in Clustering High-Dimensional Data," *Proc. 15th Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD), Part I,* pp. 183-195, 2011.
[39] G. Frederix and E.J. Pauwels, "Shape-Invariant Cluster Validity Indices," *Proc. Fourth Industrial Conf. Data Mining (ICDM),* pp. 96-105, 2004.
[40] D. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proc. IEEE Seventh Int'l Conf. Computer Vision (ICCV),* vol. 2, pp. 1150-1157, 1999.
[41] Z. Zhang and R. Zhang, *Multimedia Data Mining.* Chapman and Hall, 2009.

Nenad Tomašev graduated in 2008 from the Department of Mathematics and Informatics at the University of Novi Sad. He is working toward the PhD degree at the Artificial Intelligence Laboratory, Jožef Stefan Institute in Ljubljana. His research focus is in the area of machine learning and data mining, as well as stochastic optimization and artificial life. He has actively participated as a teaching assistant in the Petnica Science Center and Višnjan Summer School.

Miloš Radovanović received the BSc, MSc, and PhD degrees from the Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Serbia, and is also an assistant professor there. He was/is a member of several international projects supported by DAAD, TEMPUS, and bilateral programs. From 2009, he is the managing editor of the *Computer Science and Information Systems Journal.* He has (co)authored one programming textbook, a research monograph, and more than 40 papers in data mining and related fields.

Dunja Mladenić received the BSc, MSc, and PhD degrees all in computer science from the University of Ljubljana. She is an expert on the study and development of machine learning, data and text mining, semantic technology techniques, and their application on real-world problems. She has been associated with the J. Stefan Institute since 1987 and is currently leading the Artificial Intelligence Laboratory at the Institute. She was a visiting researcher at the School of Computer Science, Carnegie Mellon University, from 1996-1997 and 2000-2001. She has published papers in refereed conferences and journals, served on the program committees of international conferences, organized international events, and coedited several books.

Mirjana Ivanović is a full professor at the Faculty of Sciences, University of Novi Sad. She is the Head of the Chair of Computer Science and a member of the University Council for Informatics. She is an author or coauthor of 13 textbooks and of more than 230 research papers on multiagent systems, e-learning, intelligent techniques (CBR, data, and web mining), most of which are published in international journals and conferences. She is/was a member of program committees of more than 100 international conferences and is the editor-in-chief of the *Computer Science and Information Systems Journal.*

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

Authorized licensed use limited to: Universiteit van Tilburg. Downloaded on November 23,2021 at 15:51:37 UTC from IEEE Xplore. Restrictions apply.