Can Shared Nearest Neighbors Reduce Hubness in High-Dimensional Spaces?

Arthur Flexer
Austrian Research Institute for Artificial Intelligence
Freyung 6/6, Vienna, Austria
Email: arthur.flexer@ofai.at

Dominik Schnitzer
Austrian Research Institute for Artificial Intelligence
Freyung 6/6, Vienna, Austria
Email: dominik.schnitzer@ofai.at

***Abstract*—'Hubness' is a recently discovered general problem of machine learning in high dimensional data spaces. Hub objects have a small distance to an exceptionally large number of data points, and anti-hubs are far from all other data points. It is related to the concentration of distances which impairs the contrast of distances in high dimensional spaces. Computation of secondary distances inspired by shared nearest neighbor (SNN) approaches has been shown to reduce hubness and concentration and there already exists some work on direct application of SNN in the context of hubness in image recognition. This study applies SNN to a larger number of high dimensional real world data sets from diverse domains and compares it to two other secondary distance approaches (local scaling and mutual proximity). SNN is shown to reduce hubness but less than other approaches and, contrary to its competitors, it is only able to improve classification accuracy for half of the data sets.**

I. Introduction

In a number of recent publications [1], [2], [3] hubness has been introduced and discussed as a new aspect of the curse of dimensionality [4]. Hub objects are data points which have a small distance to many other data points in high dimensional spaces which is related to the phenomenon of concentration of distances. This behavior has a negative impact on many machine learning tasks including classification [1], nearest neighbor based recommendation [5], [6], outlier detection [1], [7] and clustering [8]. Shared Nearest Neighbors (SNN) [9] is an algorithm that re-scales distance spaces to so-called secondary distances. Without referring to the problem of hubness, it has been discussed as a way to "defeat the curse of dimensionality" [10]. Local scaling and mutual proximity, two approaches inspired by the general idea behind SNN, have already been shown to decisively reduce hubness and the concentration of distances [3]. SNN itself has been applied successfully to high dimensional image recognition data also reducing hubness to a certain degree [11]. The main contributions of this study are (i) an evaluation of the ability of SNN to reduce hubness on a larger set of high dimensional real world data sets from other domains and (ii) a comparison of SNN to local scaling and mutual proximity.

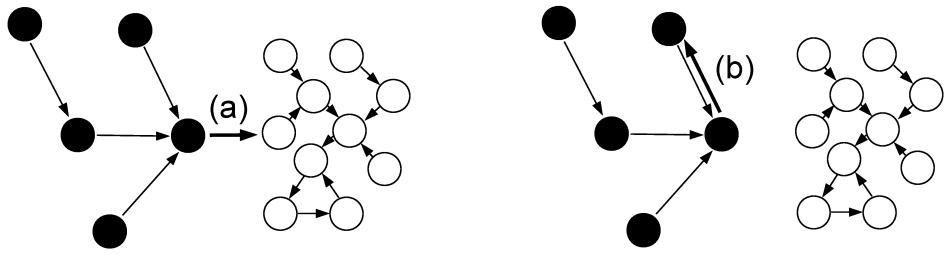
We discuss related work in Section II, present evaluation measures and methods to compute secondary distances in Section III, show results on artificial data in Section IV and real data in Section V and draw conclusions in Section VI.

This research is supported by the Austrian Science Fund (FWF - P24095)

II. Related Work

The concentration of distances is an aspect of the curse of dimensionality [4], which is a general term for problems of learning in high dimensional spaces. It is the surprising characteristic of all points in a high dimensional space to be at almost the same distance to all other points in that space [12]. It is usually measured as a ratio between spread and magnitude, e.g. the ratio between the standard deviation of all distances to an arbitrary 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 consequence of high dimensionality and has been studied for Euclidean spaces and other l^p norms [13], [12]. For cosine distances it has been shown that the mean stays constant while the standard deviation diminishes with the ratio again converging to zero [6]. It is clear that this phenomenon 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 [14]. But it should also be mentioned that in case the data space exhibits a stable cluster configuration (i.e. between-cluster distances dominate within-cluster distances), the distances should not concentrate at all [15].

To avoid this problem of concentration of distances the use of 'Shared Neighbor Distances' has been proposed by Houle et al. [10] 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 [9] to improve the clustering of non-globular clusters. As the name suggests, SNN similarity is based on computing the overlap between the k nearest neighbors of two objects. SNN approaches try to symmetrize nearest neighbor relations using only rank and not distance information. Houle et al. [10] argued, that the rank information SNN is based on might still be meaningful even when distances concentrate in high dimensions. In an extensive study using artificial and three real world image recognition 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 classification rates measured as area under receiver operating curve based on nearest neighbor classification. But the authors do not make a connection to the 'hubness' phenomenon which at the time of their study was not very well-known.



(a) Original nearest neighbor relations

(b) Desired nearest neighbor relations

Fig. 1. Schematic plot of two classes (black/white filled circles). Each circle has its nearest neighbor marked with an arrow: (a) violates the *pairwise stability* clustering assumption, (b) fulfills the assumption. In many classification and retrieval scenarios, (b) would be the desired nearest neighbor relation for the dataset.

Hubness has been first documented and established by Aucouturier and Pachet in 2004 [16] for computation of music similarity in the field of music information retrieval. Hub songs are, according to the music similarity function, similar to very many other songs and therefore keep appearing unwontedly often in recommendation lists preventing other songs from being recommended at all. In a study analyzing performance of an online audio-based music recommendation service [17], it was demonstrated that only two thirds of the songs can be reached in principle but the majority of recommended songs stems from a subset of only about a third of all songs. Hub songs also exhibit less perceptual similarity (measured via listening tests) to the songs they are close to, according to an audio similarity function, than non-hub songs [5]. The existence of the hub problem has also been reported for music recommendation based on collaborative filtering instead of on audio content analysis [18]. Similar effects have been documented for other types of multimedia retrieval and recommendation such as speech [19], image [20] and text retrieval [1]. The hubness phenomenon has since then been identified as a general problem of machine learning in high dimensional data spaces. The impact of hubness on a range of distance based machine learning algorithms has been documented including classification [1], nearest neighbor based recommendation [5], [6], outlier detection [1], [7] and clustering [8]. Berenzweig [21] was the first to suspect a connection between the hub problem and the high dimensionality of the feature space. The hub problem was seen as a direct result of the curse of dimensionality [4]. Radovanović et al. [1] were able to provide more insight by linking the hub problem to the above explained property of concentration [12]. Proofs concerning concentration of distances and all points being at the same distance to all other points have been formulated for dimensionality approaching infinity. Radovanović et al. [1] presented the argument that in the finite case, some points are expected to be closer to the center than other points and are at the same time closer, on average, to all other points. Such points closer to the center have a high probability of being hubs, i.e. of appearing in nearest neighbor lists of many other points. Points which are further away from the center have a high probability of being 'anti-hubs', i.e. points that never appear in any nearest neighbor list. It is also important to note that the degree of concentration and hubness is linked to the intrinsic rather than extrinsic dimension of the data space. Whereas the extrinsic dimension is the actual number of dimensions of a data space the intrinsic dimension is the, often much smaller, number of dimensions necessary to represent a feature space without loss of information.

Two methods (local scaling (LS) and mutual proximity (MP)) which are closely related to SNN approaches have recently been proposed by Schnitzer et al. [3] as a way to reduce the negative effects of hubness. Both methods aim at repairing asymmetric nearest neighbor relations. The asymmetric relations are a direct consequence of the presence of hubs since a hub y is the nearest neighbor of x, but the nearest neighbor of the hub y is another point a ($a \neq x$). This is because hubs are by definition nearest neighbors to very many data points but only a fixed number of data points can be the k-nearest neighbors to a hub. Fig. 1 illustrates the effect: although a is, in terms of the distance measure, the correct answer to the nearest neighbor query for y, it may be beneficial to use a distance measure that enforces symmetric nearest neighbors. Thus a small distance between two objects should be returned only if their nearest neighbors concur. The positive impact of LS and MP was measured as a decrease of hubness and an accuracy increase in k-nearest neighbor classification experiments on thirty real world data sets.

So-called 'hubness-aware' SNN approaches have been studied for nearest neighbor classification [11] and clustering [22] by Tomašev et al. These hubness-aware approaches are based on the notion of 'bad hubs', i.e. hubs that show a disagreement of class information for the majority of data points they are nearest neighbors to. The definition of bad hubs is motivated by the 'cluster assumption' from semi-supervised learning which states that most data points within a cluster (or an area of high data density) should have the same class label (see e.g. [23]). Bad hubs violate the cluster assumption and are therefore crucial for all kinds of learning tasks. A quantitative index for the 'bad hubness' of a data point can be used for a weighting scheme in k-nearest neighbor classification [1], [11]. These hubness-aware SNN approaches use class label information to compute secondary distances and are therefore less general than the fully unsupervised approaches like classic SNN, LS or MP. Classic SNN has been compared to hubness-aware SNN on a number of artificial data sets and within an image recognition context [11]. Both types of SNN approaches are able to reduce hubness and improve nearest neighbor classification, with hubness-aware SNN being better at classification which seems as expected since it does use class label information.

III. METHODS

Before presenting our results in sections IV and V, we introduce all methods and evaluation measures used in this work. The number of items in a data set will be denoted with N, the extrinsic dimensionality of our data spaces by d.

A. Evaluation measures

The following indices will be used to measure the performance achieved in original and re-scaled data spaces.

Hubness (S^n): To characterize the strength of the hubness phenomenon in a data set we use the hubness measure proposed by Radovanović et al. [1]. To compute hubness[1] we first define $O^n(x)$ as the n-occurrence of point x, that is, the number of times x occurs in the n-nearest neighbor lists of all other objects in the collection. Hubness is then defined as the skewness of the distribution of n-occurrences, O^n:

$$S^n = \frac{\mathrm{E}\left[(O^n - \mu_{O^n})^3\right]}{\sigma_{O^n}^3}. \tag{1}$$

A data set having high hubness produces few hub objects with very high n-occurrence and many anti-hubs with n-occurrence of zero. This makes the distribution of n-occurrences skewed with positive skewness indicating high hubness. Figure 2.a shows a highly skewed O^n histogram ($S^{n=5} = 5.6$).

Nearest neighbor classification accuracy (C^k): We report the k-nearest neighbor (kNN) classification accuracy using leave-one-out cross-validation, where classification is performed via a majority vote among the k nearest neighbors, with the class of the nearest neighbor used for breaking ties. We denote the kNN accuracy as C^k. We use $k = 5$ for all experiments. The classification accuracy measures to what degree the distance space reflects the class information, i.e. the semantic meaning of the data.

B. Reducing hubness

We introduce the three methods we will apply to reduce hubness by using each method on the whole distance matrix and computing secondary distances. As a first illustration of the effect of these methods, Figure 2.b shows an O^n histogram after using one of the hubness reduction methods (MP) presented below.

Shared Nearest Neighbors (SNN): SNN uses the neighborhood information to help enforce pairwise stability. SNN is computed as a set intersection of the k-nearest neighbor lists NN of two objects x, y:

$$SNN(x, y) = |NN(x) \cap NN(y)|/k. \tag{2}$$

This way SNN strictly strengthens symmetric nearest neighbor relations which in turn should also manifest itself in a reduction of hubness. We use SNN with $k = 50$, see Sect. V for more information.

[1]Matlab scripts for hubness analysis are available for download on our web page: http://ofai.at/research/impml/projects/hubology.html

Local Scaling (LS): Local scaling [24] transforms arbitrary distances to so-called *affinities* (that is, similarities) according to:

$$LS(D_{x,y}) = \exp\left(-\frac{{D_{x,y}}^2}{\sigma_x\,\sigma_y}\right), \tag{3}$$

where σ_x denotes the distance between object x and its k'th nearest neighbor. $LS(D_{x,y})$ tends to make neighborhood relations more symmetric by including local distance statistics of both data points x and y in the scaling. We use LS with $k = 10$, as it returned the best and most stable results. Since the focus of this paper is on SNN and for reasons of brevity, we do not present a detailed evaluation of this parameter here. Please note that this variant of LS differs from the one used in [3], where instead of using the distance to the k'th nearest neighbor to rescale the distances, the average distance of the k nearest neighbors is being used.

Mutual Proximity (MP): MP reinterprets the original distance space so that two objects sharing similar nearest neighbors are more closely tied to each other, while two objects with dissimilar neighborhoods are repelled from each other. This is done by transforming the distance of two objects into a mutual proximity in terms of their distribution of distances. It was shown that by using this mutual reinterpretation of distances hubness is decisively reduced, while the intrinsic dimensionality of the data stays the same [3]. To compute MP, we assume that the distances $D_{x,i=1..N}$ from an object x to all other objects in our data set follow a certain probability distribution, thus any distance $D_{x,y}$ can be reinterpreted as the probability of y being the nearest neighbor of x, given their distance $D_{x,y}$ and the probability distribution $P(X)$. In this work we assume that the distances $D_{x,i=1..N}$ follow a Gaussian distribution. MP is defined as the probability that y is the nearest neighbor of x given $P(X)$ and x is the nearest neighbor of y given $P(Y)$:

$$MP(D_{x,y}) = P(X > D_{x,y} \cap Y > D_{y,x}). \tag{4}$$

Computing $1 - SNN$, $1 - LS$ and $1 - MP$ turns the similarities into distances.

IV. EXPERIMENTS WITH ARTIFICIAL DATA

In our experiments we use uniformly distributed data randomly sampled from a d-dimensional unit cube. By using this strategy to generate artificial data for our experiments, the following basic effects should be observable with increasing data dimensionality: (i) the ratio of standard deviation and mean of all distances should concentrate, (ii) with the concentration of distances the hubness phenomenon should also emerge, and (iii) all three methods SNN, LS and MP should be able to reduce the hubness.

Figures 3 and 4 show these general effects of high dimensional data for Euclidean (ℓ^2) and $cosine$ distances. We start by generating ten dimensional data ($d = 10$) and gradually increase the data dimensionality to $d = 100$, sampling $N = 3\,000$ data points and averaging our measurements of distance concentration and hubness over 100 repetitions. In Fig. 3 we present results using ℓ^2 norm as distance. In Fig. 3.a and 3.b we can see that with increasing dimensionality, the distances clearly concentrate. At the same time the measured hubness

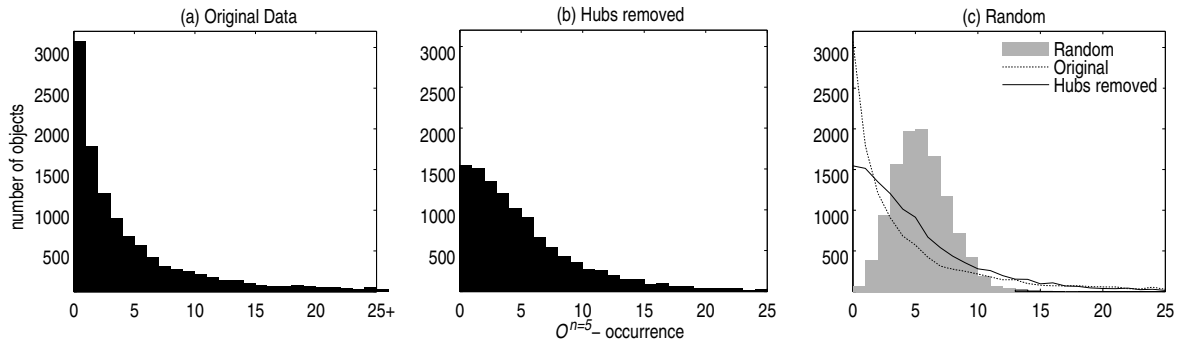


Fig. 2. Illustrative plot of (a) a skewed n-occurrence (O^n) histogram, (b) a corrected distance space with the hubs removed (using mutual proximity), and for comparison (c) the n-occurrence computed from random distances. Please note that the right-most bin in plot (a) reads "25+", but only "25" in plots (b) and (c). The data used is taken from a real world application [3].

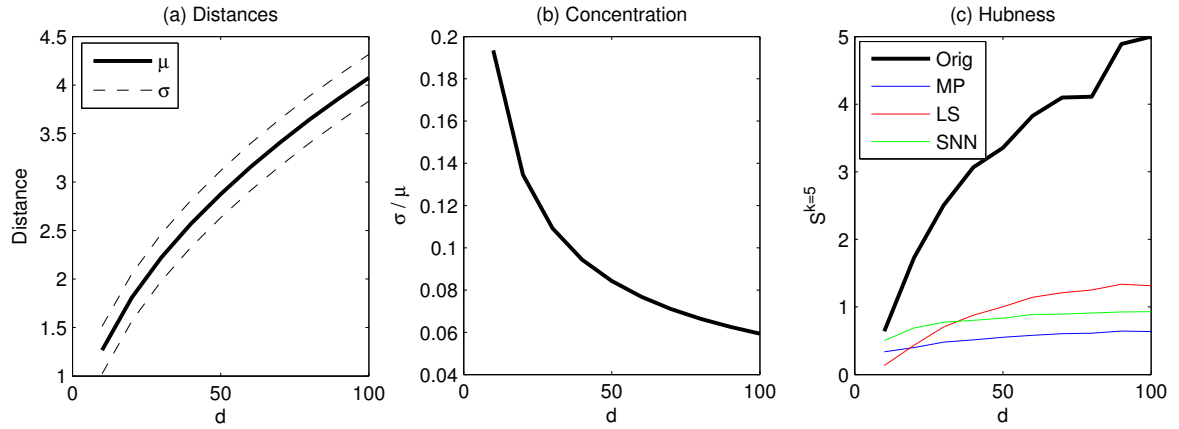


Fig. 3. With increasing dimensionality, the measured ℓ^2 distances concentrate and the hubness effect becomes more pronounced. All three methods, LS (red), MP (blue) and LS (green), reduce hubness.

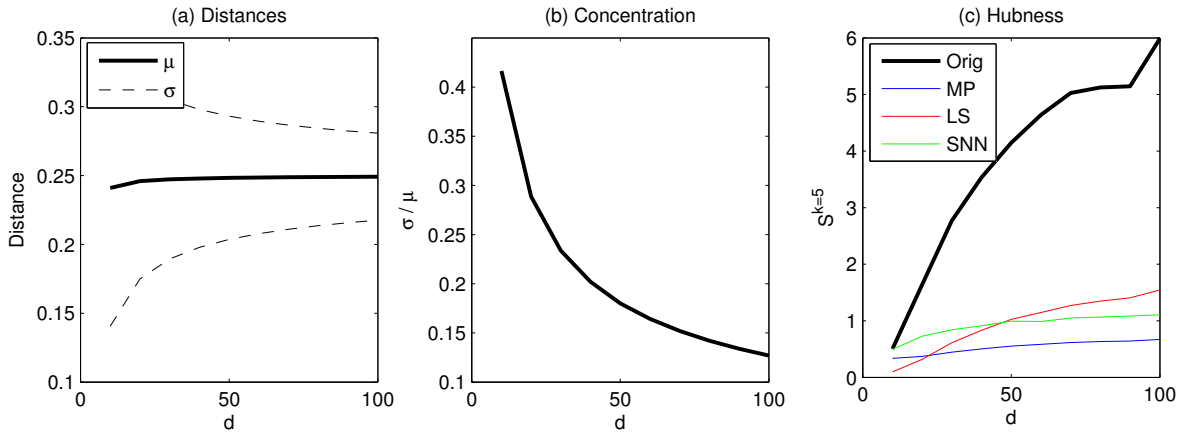


Fig. 4. With increasing dimensionality, the measured **cosine** distances concentrate and the hubness effect becomes more pronounced. All three methods, LS (red), MP (blue) and LS (green), reduce hubness.

(Fig. 3.c) increases steadily up to a value of 5, which already indicates a strong skewness of the O^n–occurrences. The figure also shows the impact of using LS (red line), MP (blue) and SNN (green) on the distances (Fig 3.c). The measured hubness declines to very low values. The same effects can be observed using $cosine$ distances in Fig. 4. The only difference is that for $cosine$ distances the mean remains constant and the standard deviation diminishes with increasing dimensionality as has been observed before [6]. We demonstrated these effects for ℓ^2 and $cosine$ distances since our real world data in the next section uses the same distance norms.

V. Experiments with Real Data

For the experiments with real data we examine six standard machine learning datasets from various domains showing high hubness: (i) *dexter*, *dorothea*, *mini-newsgroups* (UCI, [25]), (ii) *splice*, *dna* (LibSVM, [26]), (iii) *c1ka-twitter* (CP, [27]). Whereas *dexter*, *mini-newsgroups* and *c1ka-twitter* are text-based data sets from different domains, *dorothea*, *splice* and *dna* contain data from biological domains. The selected data sets are characterized in more detail in Tab. I. Each set is described by its number of classes (*Cls*), its size (*N*), its extrinsic (d) and intrinsic (d_{mle}) data dimension and the distance measure used (column *Distance*: cosine, ℓ^2 norm, or secondary distances computed with SNN, LS and MP). To measure the intrinsic data dimension we use the maximum likelihood estimator proposed by [28]. Please note that our dimensionality estimates are based on the original and re-scaled distance spaces where we treat the lines of the distance matrices as vectorial input to the estimator. This is necessary since for SNN, LS and MP only distance spaces, not feature vectors, are available. The remaining two right-most columns show classification accuracy $C^{k=5}$ and hubness $S^{n=5}$. We sorted the table according to the hubness of each dataset with *dexter* showing the smallest but still considerable hubness on top ($S^{n=5} = 4.22$) and *dna* with highest hubness on bottom ($S^{n=5} = 16.52$). For every data set, we also show accuracy and hubness results when using SNN, LS and MP.

Looking at the hubness results ($S^{n=5}$), it can be observed that all three secondary distances are able to decisively reduce hubness. The decrease in hubness is most pronounced for data sets which show very high hubness in the original distance space (*dorothea*, *c1ka-twitter* and *dna*). For all six datasets, MP always performs best, with LS being second and SNN third. Whereas the reduced hubness scores for MP range from 0.48 to 1.79, the scores for SNN show a more considerable remaining hubness ranging from 1.47 to 3.89.

Looking at the classification accuracy results ($C^{k=5}$), LS and MP always show an improvement compared to the original distance space. Sometimes this improvement is small as with dataset *dorothea* where LS achieves 92.9% and MP 93.1% compared to the original 90.2%. But very often the gain is quite impressive as with dataset *c1ka-twitter* where LS achieves 51.7% and MP 50.8% compared to the original 26.6%. Best results are achieved three times using LS and three times using MP. SNN shows more modest improvements only for three datasets: *splice*, *dorothea* and *c1ka-twitter*. For datasets *dexter* and *mini-newsgroups* there is a considerable decrease in accuracy when using SNN, and for dataset *dna* there is a small decrease. For any method computing secondary distances it is not only important to improve the distance space in terms of hubness, but it is also crucial that the semantic meaning of the data is in correspondence with the new distance space. Whereas LS and MP show improved agreement with the class information for all six data sets, SNN is less reliable and achieves this only for three data sets.

The above SNN results have all been computed with a neighborhood range of $k = 50$. As has already been demonstrated before [10], the performance of SNN might vary with the selected k, probably also depending on the size of the collection and individual class sizes. We did a range search for $k = 5, 10, 15, 20, \ldots, 100$ in all six databases. Results for nearest neighbor classification accuracy ($C^{k=5}$) are given in Fig. 5. As can be seen performance plateaued at $k = 50$ for most tested databases (gray dashed vertical line). Dataset *dexter* performs a little better at values around $k = 15$ and *c1ka-twitter* shows a slight increase even above $k = 50$. Since dataset *dexter* has only 300 data points it is not surprising that smaller neighborhood sizes perform better in this case. At a neighborhood size of $k = 15$, SNN is able to perform on a level with the original data space at about 80.3%, but SNN would still be inferior to LS and MP. For dataset *c1ka-twitter*, a neighborhood size of $k = 100$ improves accuracy to about 45% which is still below LS and MP.

The main difference between SNN and LS and MP is the usage of distance information when computing secondary distances. Whereas SNN relies only on rank information and discards the distance information, both LS and MP use numerical information of the distance spaces. Whereas LS uses local distance information to re-scale the distances, MP does this based on probability distribution models of the full distance space. Since SNN uses only rank information, the re-scaled distance space can only contain $k + 1$ different values, in our case from $SNN(x, y) = 0/50, 1/50, 2/50$ to $50/50$, depending on the size of the nearest neighbor list overlap. As shown in Figure 6, not even all of the 51 values are being used. For every data set, we compute the percentage of times each of the 51 values appears in the re-scaled distance spaces. We sort these percentages in descending order and plot the cumulative percentage on the y-axis versus the number of different distance values needed to achieve the respective cumulative sum on the x-axis. For four data sets (*dna*, *dorothea*, *mini-newsgroups*, *splice*), less then 20 different distance values yield a cumulative sum of 100%, i.e. less than 20 different values appear in the re-scaled distance space. For *dexter* and *c1ka-twitter*, less than 40 values appear in the respective re-scaled distance spaces. This observation of a certain loss of information is confirmed when looking at the estimates for intrinsic dimensionality d_{mle} in Tab. I. With exception of data set *dna*, SNN always results in much lower intrinsic dimensionality compared to the original distance spaces or the ones re-scaled via LS or MP. These differences of utilizing distance information might be at the heart of the better overall performance of LS and MP when compared to SNN.

As to computational costs, all three methods as used in this study require the full distance matrix between all N points in a data set for computation of secondary distances. For MP, there already exists an approximation which has been shown to produce only slightly lower results in terms of hubness and accuracy [3]. The Gaussian distribution parameters are

Name	*Classes*	N	d	d_{mle}	*Distance*	$C^{k=5}$	$S^{n=5}$
dexter	2	300	20 000	29	cos	80.3%	**4.22**
				9	SNN	75.0%	1.61
				32	LS	86.0%	1.42
				15	MP	90.0%	0.58
splice	2	1 000	27	25	ℓ^2	69.4%	**4.55**
				21	SNN	72.1%	1.47
				27	LS	77.9%	1.18
				26	MP	77.2%	0.48
mini-newsgroups	20	2 000	8 811	18	cos	65.6%	**5.14**
				12	SNN	57.5%	1.73
				18	LS	68.9%	0.94
				16	MP	68.4%	0.60
dorothea	2	800	100 000	414	ℓ^2	90.2%	**12.91**
				26	SNN	90.8%	3.57
				409	LS	92.9%	2.23
				379	MP	93.1%	1.66
c1ka-twitter	17	969	49 820	43	cos	26.6%	**14.63**
				10	SNN	40.8%	3.89
				47	LS	51.7%	3.42
				30	MP	50.8%	1.79
dna	3	2 000	180	33	cos	75.6%	**16.52**
				30	SNN	74.5%	1.67
				35	LS	81.3%	1.32
				30	MP	83.4%	0.59

TABLE I. RESULTS FOR EXPERIMENTS WITH REAL DATA, ORDERED BY ASCENDING HUBNESS OF ORIGINAL DISTANCE SPACES (COLUMN $S^{n=5}$, BOLD FACE). PLEASE SEE SECTION V FOR DETAILS.

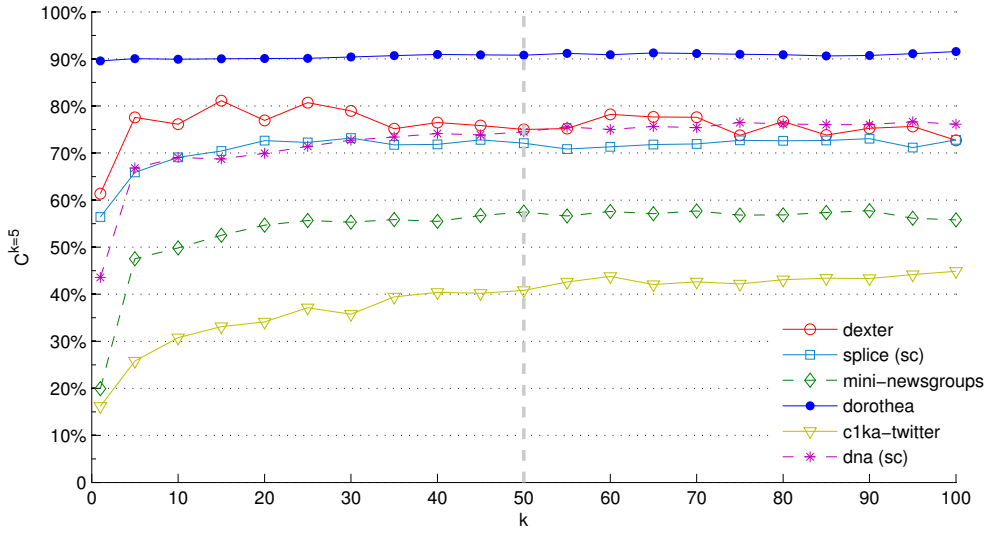


Fig. 5. Neighborhood range k of SNN (x-axis) versus nearest neighbor classification accuracy (y-axis) for all six data sets.

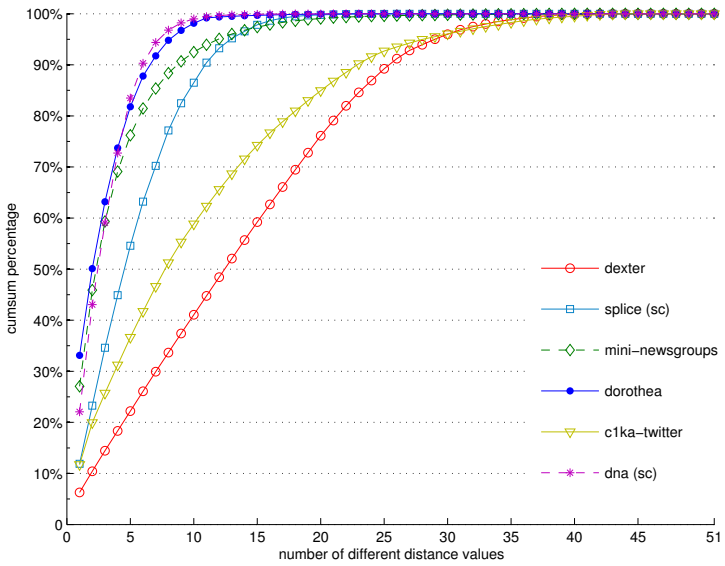


Fig. 6. Number of different distance values in distances spaces (x-axis) vs. cumulative percentage of all distances (y-axis).

estimated by randomly selecting a small fraction (S) of data points to compute the mean and standard deviation of distances for each data point. The difference to the original estimation of the parameters is that only a small fraction of distances ($S \times N$) needs to be computed, which reduces the complexity from quadratic to linear in N. Good results have been reported with S being as low as 30. Contrary to MP, LS and SNN actually need the k nearest neighbors for every data point in a dataset and therefore require information about the full distance matrix. It has been argued [8] that approximate k-nearest neighbor graphs or locality sensitive hashing could be used to lower the computational costs, but this has not yet been demonstrated in the context of hubness research.

VI. Conclusion

In this study we tried to answer the question whether the shared nearest neighbor (SNN) algorithm is able to reduce hubness in high dimensional spaces. To achieve this we evaluated SNN on a larger and more diverse number of high dimensional real world data sets than has been published before. In addition, this is the first study to compare SNN to two other secondary distance methods that have recently been proposed for reduction of hubness (local scaling (LS) and mutual proximity (MP)). In answering the question of the title of this paper, we like to state that SNN is able to decisively reduce hubness in artificial as well as all six real world data sets. But SNN is not able to reach the performance of LS or MP, with MP being the best overall at reducing hubness. Whereas LS and MP are able to also increase nearest neighbor based classification accuracy for all six data sets, SNN achieves this only for three out of six. The fact that application of SNN even decreases classification performance for three out of six datasets shows that its resulting secondary distances are not always in good agreement with the semantics (class label information) of the data spaces. Therefore we would advocate usage of LS and MP rather than SNN to reduce hubness in high dimensional data.

This work focused on fully unsupervised methods to compute secondary distances not using any class label information. Since both LS and MP seem to be superior to SNN, it would be interesting to also use these two methods as part of the supervised hub-aware approaches to classification [11] and clustering [22] that so far rely on SNN information.

References

[1] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Hubs in space: Popular nearest neighbors in high-dimensional data," *Journal of Machine Learning Research*, vol. 11, pp. 2487–2531, December 2010.

[2] I. Suzuki, K. Hara, M. Shimbo, Y. Matsumoto, and M. Saerens, "Investigating the effectiveness of laplacian-based kernels in hub reduction," in *Proceedings of the Twenty-Sixth Conference on Artificial Intelligence (AAAI)*, 2012, pp. 1112–1118.

[3] D. Schnitzer, A. Flexer, M. Schedl, and G. Widmer, "Local and global scaling reduce hubs in space," *Journal of Machine Learning Research*, vol. 13, pp. 2871–2902, October 2012.

[4] R. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[5] A. Flexer, D. Schnitzer, and J. Schlüter, "A mirex meta-analysis of hubness in audio music similarity." in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, 2012, pp. 175–180.

[6] A. Nanopoulos, M. Radovanović, and M. Ivanović, "How does high dimensionality affect collaborative filtering?" in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 293–296.

[7] A. Flexer and D. Schnitzer, "Using mutual proximity for novelty detection in audio music similarity," in *Procceedings of the 6th International Workshop on Machine Learning and Music*, Prague, Czech Republic, 2013.

[8] N. Tomašev, M. Radovanović, D. Mladenić, and M. Ivanović, "The role of hubness in clustering high-dimensional data," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2011, pp. 183–195.

[9] R. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *IEEE Transactions on Computers*, vol. 22, pp. 1025–1034, 1973.

[10] M. E. Houle, H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Can Shared-Neighbor Distances Defeat the Curse of Dimensionality?" in *Scientific and Statistical Database Management*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6187, ch. 34, pp. 482–500.

[11] N. Tomašev and D. Mladenić, "Hubness-aware shared neighbor distances for high-dimensional k-nearest neighbor classification," *Knowledge and Information Systems*, 2013.

[12] D. François, V. Wertz, and M. Verleysen, "The concentration of fractional distances," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 873–886, 2007.

[13] C. Aggarwal, A. Hinneburg, and D. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *Database Theory - ICDT 2001*, ser. Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2001, pp. 420–434.

[14] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?" in *In Int. Conf. on Database Theory*, 1999, pp. 217–235.

[15] K. P. Bennett, U. M. Fayyad, and D. Geiger, "Density-based indexing for approximate nearest-neighbor queries," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 233–243.

[16] J.-J. Aucouturier and F. Pachet, "Improving timbre similarity: How high is the sky?" *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, no. 1, 2004.

[17] A. Flexer, M. Gasser, and D. Schnitzer, "Limitations of interactive music recommendation based on audio content," in *Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound*. New York, NY, USA: ACM, 2010.

[18] Ò. Celma, "Music recommendation and discovery in the long tail," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, 2008.

[19] G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D. Reynolds, "Sheep, goats, lambs and wolves a statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation," in *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP-98)*, 1998.

[20] A. Hicklin, C. Watson, and B. Ulery, *The myth of goats: How many people have fingerprints that are hard to match?* US Dept. of Commerce, National Institute of Standards and Technology (NIST), 2005.

[21] A. Berenzweig, "Anchors and hubs in audio-based music similarity," Ph.D. dissertation, Columbia University, 2007.

[22] N. Tomašev, M. Radovanović, D. Mladenić, and M. Ivanović, "The role of hubness in clustering high-dimensional data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 99, no. PrePrints, p. 1, 2013.

[23] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006.

[24] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2005, vol. 17, pp. 1601–1608.

[25] A. Frank and A. Asuncion, "UCI machine learning repository," 2010, repository located at: http://archive.ics.uci.edu/ml.

[26] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[27] M. Schedl, "On the Use of Microblogging Posts for Similarity Estimation and Artist Labeling," in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, the Netherlands, August 2010.

[28] E. Levina and P. J. Bickel, "Maximum likelihood estimation of intrinsic dimension," in *Advances in Neural Information Processing Systems 17*. Cambridge, MA: MIT Press, 2005, pp. 777–784.