Centering versus Scaling for Hubness Reduction

Roman Feldbauer and Arthur Flexer

Austrian Research Institute for Artificial Intelligence (OFAI),
Freyung 6/6/7, 1010 Vienna, Austria
{roman.feldbauer,arthur.flexer}@ofai.at

Abstract. Hubs and anti-hubs are points that appear very close or very far to many other data points due to a problem of measuring distances in high-dimensional spaces. Hubness is an aspect of the curse of dimensionality affecting many machine learning tasks. We present the first large scale empirical study to compare two competing hubness reduction techniques: scaling and centering. We show that scaling consistently reduces hubness and improves nearest neighbor classification, while centering shows rather mixed results. Support vector classification is mostly unaffected by centering-based hubness reduction.

Keywords: hubness reduction, curse of dimensionality, k-NN, SVM

1 Introduction and Related Work

Hubness is a general problem of learning in high-dimensional spaces and has been recognized as an aspect of the curse of dimensionality in machine learning literature [7, 9]. Hub objects appear very close to many other data objects and anti-hubs very far from most other data objects. The effect has been shown to have a negative impact on classification [7], nearest neighbor based recommendation [2] and retrieval [10], outlier detection [6], clustering [12, 8] and visualization [1].

Hubness is related to the phenomenon of concentration of distances, which is the fact that all points are at almost the same distance to each other for dimensionality approaching infinity [3]. Radovanović et al. [7] presented the argument that for any finite dimensionality, some points are expected to be closer to the center of all data 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.

In order to reduce hubness and its negative effects, we have proposed two unsupervised methods to re-scale high-dimensional distance spaces [9]: Local Scaling (LS) and Mutual Proximity (MP). Both methods aim at repairing asymmetric nearest neighbor relations. The asymmetric relations are a direct consequence of the presence of hubs. 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 one data point

can be the nearest neighbor to a hub. The principle of the scaling algorithms is to re-scale distances to enhance symmetry of nearest neighbors. A small distance between two objects should be returned only if their nearest neighbors concur. Application of LS and MP resulted in a decrease of hubness and an accuracy increase in k-nearest neighbor classification on thirty real world datasets including text, image and music data. The general influence of hubs and anti-hubs on classifiers beyond simple nearest neighbor approaches is so far largely unexplored, with the only result being that removal of certain hubs during support vector machine (SVM) training decreases classification rates [7].

A different approach to reduce hubness is to center the data either locally or globally [11, 4]. Results so far are encouraging and comparable to those achieved with scaling. An advantage of global centering is that it computes centered data vectors whereas scaling and localized centering result in distance and similarity matrices, respectively, which can be a problem for many machine learning tasks. Since comparison of centering and scaling so far has only been conducted on seven datasets, all from the text domain, we present the first comprehensive empirical study of the two competing approaches on 28 diverse datasets. We also conduct a first analysis of the influence of centering on SVM classification.

2 Methods and Data

Before presenting our results in Section 3, we introduce all evaluation measures, methods and datasets used in this work.

2.1 Evaluation measures

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

Hubness (S^n): To characterize the strength of the hubness phenomenon in a dataset we use the hubness measure proposed by Radovanović et al. [7]. 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 S^n is then defined as the skewness of the distribution of n-occurrences O^n. A dataset 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. All our results are based on n=5-occurences.

Nearest neighbor classification accuracy (C^k): We report the k-nearest neighbor classification accuracy C^k using 5-fold 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 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.

[1] Python scripts for hubness analysis are available at: `https://github.com/OFAI`

Support vector classification accuracy: For selected datasets we perform support vector classification using nested cross-validation (CV), tuning the regularization parameter C for the linear kernel in the inner 3-fold CV, and reporting classification accuracy averaged over the outer 5-fold CV. All SVM calculations were performed with the scikit-learn package [13] for Python.

2.2 Reducing hubness

We introduce four hubness reduction methods which either re-compute the whole distance matrix to so-called secondary distances (NICDM, MP, LCENT), or are applied to the data vectors directly (CENT).

NICDM: The non-iterative contextual dissimilarity measure [5] is a local scaling variant that transforms arbitrary distances according to:

$$\mathrm{NICDM}(D_{x,y}) = D_{x,y} \frac{{\mu_{geom}}^2}{\sqrt{\mu_x \, \mu_y}}, \tag{1}$$

where μ_x (μ_y) denotes the average distance between object x (y) and its k nearest neighbors, and μ_{geom} denotes the geometric mean of all such average distances in the data. NICDM tends to make neighborhood relations more symmetric by including local distance statistics of both data points x and y in the scaling. We use NICDM with k=10, as it returned the best and most stable results in previous studies [8].

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 [9]. This is done by transforming the distance of two objects into a mutual proximity in terms of their distribution of distances. We assume that distances $D_{x,i=1..N}$ from an object x to all other objects in a dataset 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 $P(X)$. 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 $P(Y)$:

$$\mathrm{MP}(D_{x,y}) = P(X > D_{x,y} \cap Y > D_{y,x}). \tag{2}$$

In this work we assume that distances $D_{x,i=1..N}$ follow a Gaussian distribution and that $P(X)$ and $P(Y)$ are statistically independent. Computing $1 - \mathrm{MP}$ turns the respective similarities into distances.

Centering (CENT): Centering is a common preprocessing step in data analysis transforming vector data by subtracting the dataset centroid, thus shifting the origin to the latter. Suzuki et al. [11] use the method for hubness reduction in inner product similarity spaces, where the similarity of each sample to the centroid is equal to zero after centering. Since hubs are then no longer closer to the centroid than other samples, hubness might be reduced by centering. Since this holds only for inner product spaces, we replace Euclidean ℓ^2 norms with

cosine distances to gain the same effect in case ℓ^2 is the original distance for the given dataset. This is also done in case of localized centering.

Localized centering (LCENT): Localized centering tries to reduce hubness by considering *local affinity*, i.e. the average similarity of a sample x to its k nearest neighbors [4]. Similarities are calculated as

$$\mathrm{Sim}(x,y)^{\mathrm{LCENT}} = \mathrm{Sim}(x,y) - \mathrm{Sim}(x, c_\kappa(x))^\gamma \tag{3}$$

where Sim denotes a similarity measure ($1-D$, where D is a cosine distance) and $c(x)$ denotes the local centroid of x. We tune the parameters κ (neighborhood size) and γ (controls penalty) in nested cross-validation with a 3-fold inner loop in order to optimally reduce hubness.

2.3 Datasets

The four previously introduced hubness reduction methods are evaluated using 28 real-world datasets (Table 1), comprising data from biology, multimedia retrieval and general machine learning fields. All datasets have previously been used to evaluate NICDM and MP in [9] and are described further therein. Please note that we excluded two datasets from this previous study which use the symmetrized Kullback-Leibler divergence, since this is not a full metric and therefore not easy to combine with centering methods.

3 Results

We evaluate all centering and scaling methods on 28 datasets using the evaluation measures introduced in Section 2.1. Figure 1 shows the results of the evaluation, ordered by ascending hubness. Results are given as absolute decreases or increases in hubness and accuracy relative to the values obtained in original similarity spaces given in Table 1. Results are given in light blue (NICDM), blue (MP), light green (CENT) and green (LCENT) bars. For data sets based on Euclidean ℓ^2 norm, there is an additional black bar (COS) showing the decrease/increase due to switching to cosine distances alone. If cosine is already the original distance, this is marked with a small 'cos' instead of a black bar.

In accordance with the results from [9], we find that both scaling methods NICDM and MP consistently reduce hubness. For datasets of high hubness (defined as $S^{k=5} > 1.4$, from dataset 'corel1000' onwards) the reduction is more pronounced and leads to significant increases in classification accuracy (McNemar's test, marked with asterisks on right-hand side of Figure 1). There are few significant changes among the low hubness datasets.

CENT reduces hubness and improves classification for all datasets, which are originally based on cosine distances ('c224a-web', 'reuters-transcribed', 'movie-reviews', 'dexter', 'mini-newsgroups', 'c1ka-twitter'). Significant changes in other datasets appear to be either fully ('gisette', 'dorothea') or at least partially ('ionosphere','splice') caused by switching to cosine distances rather than centering, since results for COS are almost as high as those for CENT. In case

Table 1. 28 real-world datasets are reported in terms of their name, number of classes (Cls.) and instances (N), dimensionality (d), original distance measure (Dist.) and classification accuracy ($C^{k=5}$). Datasets are ordered by ascending hubness ($S^{n=5}$).

Name	Cls.	N	d	Dist.	$C^{k=5}$	$S^{n=5}$
LibSVM fourclass (sc)	2	862	2	ℓ^2	1.0	0.15
UCI arcene	2	100	10000	ℓ^2	0.729	0.25
UCI liver-disorders (sc)	2	345	6	ℓ^2	0.594	0.38
LibSVM australian	2	690	14	ℓ^2	0.677	0.44
UCI diabetes (sc)	2	768	8	ℓ^2	0.733	0.49
LibSVM heart	2	270	13	ℓ^2	0.815	0.50
KR ovarian-61902	2	253	15154	ℓ^2	0.917	0.66
LibSVM breast-cancer (sc)	2	683	10	ℓ^2	0.972	0.70
UCI mfeat-factors	10	2000	216	ℓ^2	0.946	0.79
LibSVM ger.num (sc)	2	1000	24	ℓ^2	0.711	0.81
LibSVM colon-cancer	2	62	2000	ℓ^2	0.740	0.81
KR amlall	2	72	7129	ℓ^2	0.830	0.82
UCI mfeat-karhunen	10	2000	64	ℓ^2	0.972	0.84
KR lungcancer	2	181	12533	ℓ^2	0.994	1.07
CP c224a-web	14	224	1244	cos	0.898	1.09
UCI mfeat-pixels	10	2000	240	ℓ^2	0.975	1.28
UCI duke (train)	2	38	7129	ℓ^2	0.582	1.37
Corel corel1000	10	1000	192	ℓ^2	0.671	1.45
UCI sonar (sc)	2	208	60	ℓ^2	0.513	1.54
UCI ionosphere (sc)	2	351	34	ℓ^2	0.875	1.56
UCI reuters-transcribed	10	201	2730	cos	0.478	1.61
PaBo movie-reviews	2	2000	10382	cos	0.696	4.07
UCI dexter	2	300	20000	cos	0.770	4.22
UCI gisette	2	6000	5000	ℓ^2	0.957	4.48
LibSVM splice (sc)	2	1000	60	ℓ^2	0.706	4.55
UCI mini-newsgroups	20	2000	8811	cos	0.672	5.14
UCI dorothea	2	800	100000	ℓ^2	0.891	12.93
CP c1ka-twitter	17	969	49820	cos	0.273	14.63

of 'fourclass' both CENT and COS lead to considerable accuracy decreases. In two other cases CENT is beneficial for classification ('diabetes') or detrimental ('mfeat-factors'), with COS alone showing no effect.

We obtain very mixed results using LCENT. While the method performs equally in terms of hubness reduction and accuracy to scaling techniques for some datasets in high hubness regimes ('ionosphere', 'splice', 'dorothea', 'c1ka-twitter'), it increases hubness for some datasets and effectively decreases classification accuracy for 'fourclass', 'ovarian', 'mfeat-factors', 'mfeat-karhunen', 'mfeat-pixels' and 'mini-newsgroups'. Neither positive nor negative changes are strictly coupled to the original distance metric, and there is only a moderate correlation between changes in hubness and accuracy (Pearson's $r = -0.56$).

To sum up our results, whereas both scaling methods NICDM and MP consistently help against the negative effects of hubness, LCENT reaches the same

performance only for some datasets, while at the same time having a higher computational cost due to tuning of two parameters. CENT on the other hand is computationally very efficient and effective for all cosine-based datasets.

Centering and support vector classification: Additionally, we investigated the effect of hubness on support vector machines. Among the introduced hubness reduction methods, only CENT returns vector data instead of distance matrices. We therefore restrict SVM analysis (i) to this technique and (ii) to datasets, which exhibit reduced hubness after centering (i.e. all cosine-based and three other datasets). In this section we refer to objects with an n-occurence greater than $5 \times n$ as 'hubs', to those with n-occurence of zero as 'anti-hubs' and to the remaining objects as 'normal' ($n = 5$). We perform support vector classification (linear kernel), tracking accuracies of hubs, normal points and anti-hubs before and after centering. Across all datasets (except 'gisette') and before and after centering, hub points show higher accuracy than normal and anti-hub points, which both seem to perform at a comparable level (Table 2). Using the same statistical testing procedure as above, we find no significant changes between centered and uncentered data except for the 'movie-reviews' dataset, for which we observe a minor decrease in accuracy. Centering appears to have no major impact on linear SVMs.

Table 2. Classification accuracy for linear SVM. Results are given for the complete dataset (all). Additionally, they are partitioned into hubs (H), normal (N) and anti-hubs (A). Superscript C indicates centering. Significant changes between non-centered and centered data are marked with an asterisk ($\alpha = .05$). Three datasets do not contain hubs according to the $5 \times n$ criterion (N/A).

Name	all	H	N	A	all^{C}	H^{C}	N^{C}	A^{C}
c224a-web	0.929	N/A	0.934	0.885	0.906	N/A	0.914	0.846
sonar	0.620	N/A	0.632	0.467	0.620	N/A	0.627	0.533
reuters-transcribed	0.582	N/A	0.579	0.75	0.597	N/A	0.594	0.75
movie-reviews	0.845	0.904	0.844	0.840	0.841^{*}	0.885	0.842	0.826
dexter	0.937	1.0	0.942	0.913	0.93	1.0	0.947	0.875
gisette	0.972	0.948	0.975	0.950	0.973	0.948	0.976	0.950
splice	0.793	0.931	0.805	0.751	0.794	0.931	0.805	0.754
mini-newsgroups	0.954	1.0	0.949	0.970	0.956	0.976	0.952	0.974
c1ka-twitter	0.590	0.733	0.676	0.520	0.614	0.867	0.715	0.530

4 Conclusion

We have presented the first large-scale empirical study to compare scaling and centering techniques for hubness reduction. Scaling methods outperform centering methods in terms of reduced hubness and improved nearest neighbor classification in most datasets. They are effective for datasets from various domains,

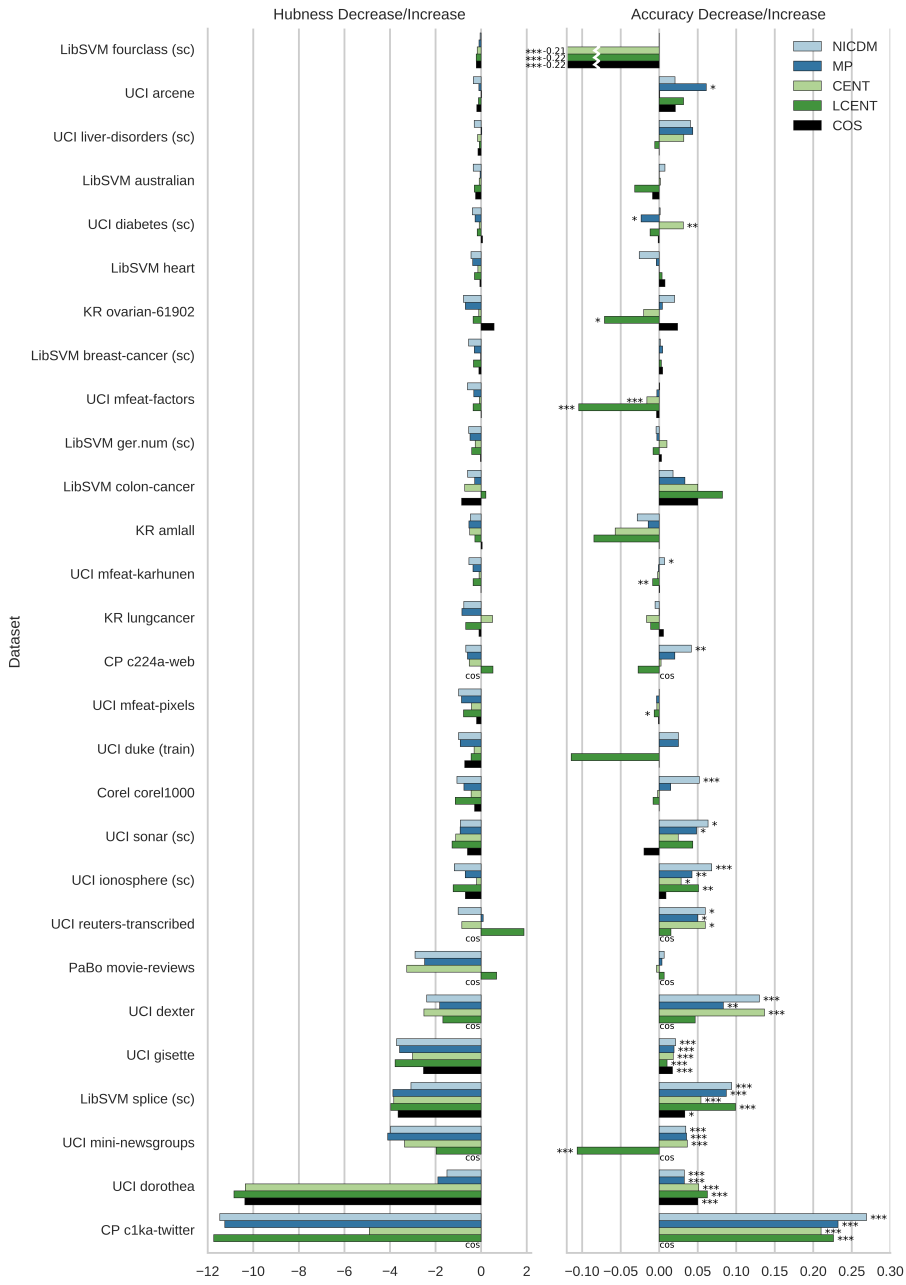

Fig. 1. Absolute decrease/increase in hubness (lower is better) and accuracy (higher is better) evaluated with $k = 5$. Significant changes are marked with asterisks: $^*\alpha = .05$, $^{**}\alpha = .01$, $^{***}\alpha = .001$. See Section 3 for more information.

and for various distance measures. Centering performs equally well for cosine distances and has the advantage of being applicable to vector data. This is especially relevant for large datasets, for which operations on similarity matrices might be computationally intractable. Localized centering is effective only for few datasets. We find no evidence for improved support vector classification due to hubness reduction via centering.

Acknowledgments This research is supported by the Austrian Science Fund (FWF): P27082, P27703.

References

1. Flexer A.: Improving visualization of high-dimensional music similarity spaces, in: *16th ISMIR Conference*, 2015.
2. Flexer A., Schnitzer D., Schlüter J.: A MIREX meta-analysis of hubness in audio music similarity, in: *13th ISMIR Conference*, 2012.
3. Francois D., Wertz V., Verleysen M.: The concentration of fractional distances, *IEEE Trans. on Knowledge and Data Engineering*, 19:873886, 2007.
4. Hara, K., Suzuki, I., Shimbo, M., Kobayashi, K., Fukumizu, K., Radovanović, M.: Localized centering: Reducing hubness in large-sample data hubness in high-dimensional data, in: *29th AAAI Conference on Artificial Intelligence*, pp. 2645–2651, 2015.
5. Jegou, H., Harzallah, H., Schmid, C.: A contextual dissimilarity measure for accurate and efficient image search, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.
6. Radovanović M., Nanopoulos A., Ivanović M.: Reverse nearest neighbors in unsupervised distance-based outlier detection, *IEEE Trans. on Knowledge and Data Engineering*, 27(5), 1369-1382, 2015.
7. Radovanović M., Nanopoulos A., Ivanović M.: Hubs in space: Popular nearest neighbors in high-dimensional data, *Journal of Machine Learning Research*, 11:2487–2531, 2010.
8. Schnitzer D., Flexer A.: The Unbalancing Effect of Hubs on K-medoids Clustering in High-Dimensional Spaces, in: *International Joint Conference on Neural Networks*, 2015.
9. Schnitzer D., Flexer A., Schedl M., Widmer G.: Local and Global Scaling Reduce Hubs in Space, *Journal of Machine Learning Research*, 13:2871–2902, 2012.
10. Schnitzer D., Flexer A., Tomasev N.: A Case for Hubness Removal in High-Dimensional Multimedia Retrieval, in: *36th ECIR*, 2014.
11. Suzuki, I., Hara, K., Shimbo, M., Saerens, M., Fukumizu, K.: Centering similarity measures to reduce hubs, in: *Conference on Empirical Methods in Natural Language Processing (EMNLP 13)*, pp. 613–623, 2013.
12. Tomašev N., Radovanović M., Mladenić D., Ivanović M.: The Role of Hubness in Clustering High-dimensional Data, *IEEE Transactions on Knowledge and Data Engineering*, Volume 26, Issue 3, 2014.
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, Volume 12, pp. 2825–2830, 2011.