Centering Similarity Measures to Reduce Hubs

Ikumi Suzuki
National Institute of Genetics
Mishima, Shizuoka, Japan
suzuki.ikumi@gmail.com

Kazuo Hara
National Institute of Genetics
Mishima, Shizuoka, Japan
kazuo.hara@gmail.com

Masashi Shimbo
Nara Institute of Science and Technology
Ikoma, Nara, Japan
shimbo@is.naist.jp

Marco Saerens
Université catholique de Louvain
Louvain-la-Neuve, Belgium
marco.saerens@uclouvain.be

Kenji Fukumizu
The Institute of Statistical Mathematics
Tachikawa, Tokyo, Japan
fukumizu@ism.ac.jp

Abstract

The performance of nearest neighbor methods is degraded by the presence of *hubs*, i.e., objects in the dataset that are similar to many other objects. In this paper, we show that the classical method of *centering*, the transformation that shifts the origin of the space to the data centroid, provides an effective way to reduce hubs. We show analytically why hubs emerge and why they are suppressed by centering, under a simple probabilistic model of data. To further reduce hubs, we also move the origin more aggressively towards hubs, through weighted centering. Our experimental results show that (weighted) centering is effective for natural language data; it improves the performance of the *k*-nearest neighbor classifiers considerably in word sense disambiguation and document classification tasks.

1 Introduction

1.1 Background

The *k*-nearest neighbor (*k*NN) algorithm is a simple nonparametric method of classification. It has been applied to various natural language processing (NLP) tasks such as document classification (Masand et al., 1992; Yang and Liu, 1999), part-of-speech tagging (Søgaard, 2011), and word sense disambiguation (Navigli, 2009).

To apply the *k*NN algorithm, data is typically represented as a vector object in a feature space, and (dis)similarity between data is measured by the distance between the vectors, their inner product, or cosine of the angle between them (Jurafsky and Martin, 2008). With such a (dis)similarity measure, the unknown class label of a test object is predicted by a majority vote of the classes of its *k* most similar objects in the labeled training set.

Recent studies (Radovanović et al., 2010a; Radovanović et al., 2010b) have shown that if the feature space is high-dimensional, some objects in the dataset emerge as *hubs*; i.e., these objects frequently appear in the *k* nearest neighbors of other objects.

The emergence of hubs may deteriorate the performance of *k*NN classification and nearest neighbor search in general:

- If hub objects exist in the training set, they have a strong chance to be a *k*NN of many test objects. Because the class of a test object is predicted by a majority vote from its *k* nearest neighbors, prediction is biased toward the labels of the hubs.
- In information retrieval, nearest neighbor search finds objects in the database that are most relevant, or similar, to user-provided queries. If particular objects, such as hubs, are nearly always returned for any query, the retrieved results are probably not very useful.

These drawbacks may hinder application of nearest neighbor methods in NLP, as typical natural language data are extremely high-dimensional (Jurafsky and Martin, 2008) and thus prone to produce hubs.

1.2 Contributions

Centering (Mardia et al., 1979; Fisher and Lenz, 1996; Eriksson et al., 2006) is a standard technique

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 613–623,
Seattle, Washington, USA, 18-21 October 2013. ©2013 Association for Computational Linguistics

for removing observation bias in the data. It is a transformation of feature space in a way that the origin of the space is moved to the data centroid (sample mean). The distance between data objects is not changed by centering, but their inner product and cosine are affected; see Section 3 for detail.

In this paper, we advocate the use of centering as a means of reducing hubs. Specifically, we propose to measure the similarity of objects by the inner product (not distance or cosine) in the centered feature space.

Our approach is motivated by the observation that the objects similar to the data centroid tend to become hubs (Radovanović et al., 2010a). This observation suggests that the number of hubs may be reduced if we can define a similarity measure that makes all objects in a dataset equally similar to the centroid (Suzuki et al., 2012). The inner product in the centered space indeed enjoys this property.

In Section 4, we analyze why hubs emerge under a simple probabilistic model of data, and also give an account of why they are suppressed by centering.

Using both synthetic and real datasets, we show that objects similar to the centroid also emerge as hubs in multi-cluster data (Section 5), so the application of centering is wider than expected. To further reduce hubs, we also propose to move the origin of the space more aggressively towards hubs, through *weighted* centering (Section 6).

In Section 7, we show that centering and weighted centering are effective for natural language data. these methods markedly improve the performance of *k*NN classifiers in word sense disambiguation and document classification tasks.

2 Related work

Centering is a classical technique widely used in many fields of science. For instance, centering forms a preprocessing step in principal component analysis and Fisher linear discriminant analysis.

In NLP, however, centering is seldom used; the use of cosine and inner product similarities is quite common, but they are nearly always used uncentered. Non-centered cosine is used, for instance, in word sense disambiguation (Schütze, 1998; Navigli, 2009), paraphrasing (Erk and Padó, 2008; Thater et al., 2010), and compositional semantics (Mitchell and Lapata, 2008), to name a few.

There have been several approaches to improving *k*NN classification: learning similarity/distance measures from training data (metric learning) (Weinberger and Saul, 2009; Qamar et al., 2008), weighting nearest neighbors for similarity-based classification (Chen et al., 2009), and neighborhood size selection (Wang et al., 2006; Guo and Chakraborty, 2010). However, none of these have addressed the reduction of hubs.

More recently, Schnitzer et al. (2012) proposed the *Mutual Proximity* transformation that rescales distance measures to decrease hubs in a dataset. Suzuki et al. (2012) showed that kernels based on graph Laplacian, such as the commute-time kernels (Saerens et al., 2004) and the regularized Laplacian (Chebotarev and Shamis, 1997; Smola and Kondor, 2003), make all objects equally similar to the data centroid, which in turn reduce hubs.

In Section 7, we evaluate centering, Mutual Proximity, and Laplacian kernels in NLP tasks, and demonstrate that centering is equally or even more effective. Section 4 presents a theoretical justification for using centering to reduce hubs, but this kind of analysis is missing for the Laplacian kernels.

Centering is easier to compute as well. For a dataset of n objects, it takes $O(n^2)$ time to compute, whereas computing a Laplacian-based kernel requires $O(n^3)$ time for matrix inversion. Mutual Proximity also has a time complexity of $O(n^2)$.

3 Centering

Consider a dataset of n objects in an m-dimensional feature space, $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathbb{R}^m$. Throughout this paper, we use the inner product $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ as a measure of similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Let $\mathbf{K}$ be the Gram matrix of the n feature vectors, i.e., the $n \times n$ matrix whose (i, j) element holds $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$. Using $m \times n$ data matrix $\mathbf{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]$, we can write $\mathbf{K}$ as

$$\mathbf{K} = \mathbf{X}^{\mathrm{T}}\mathbf{X},$$

where $\mathbf{X}^{\mathrm{T}}$ represents the matrix transpose of $\mathbf{X}$.

Centering is a transformation in which the origin of the feature space is shifted to the data centroid

$$\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i, \tag{1}$$

and object $\boldsymbol{x}$ is mapped to the centered feature vector

$$\boldsymbol{x}^{\text{cent}} = \boldsymbol{x} - \bar{\boldsymbol{x}}. \tag{2}$$

The similarity between two objects $\boldsymbol{x}$ and $\boldsymbol{x}'$ is now measured by $\langle \boldsymbol{x}^{\text{cent}}, \boldsymbol{x}'^{\text{cent}} \rangle = \langle \boldsymbol{x} - \bar{\boldsymbol{x}}, \boldsymbol{x}' - \bar{\boldsymbol{x}} \rangle$.

After centering, the inner product between any object and the data centroid (which is a zero vector because $\bar{\boldsymbol{x}}^{\text{cent}} = \bar{\boldsymbol{x}} - \bar{\boldsymbol{x}} = \boldsymbol{0}$) is uniformly 0; in other words, all objects in the dataset have an equal similarity to the centroid. According to the observation that the objects similar to the centroid become hubs (Radovanović et al., 2010a), we can expect hubs to be reduced after centering.

Intuitively, centering reduces hubs because it makes the length of the feature vector $\boldsymbol{x}^{\text{cent}}$ short for (hub) objects $\boldsymbol{x}$ that lie close to the data centroid $\bar{\boldsymbol{x}}$; see Eq. (2). And since we measure object similarity by inner product, shorter vectors tend to produce smaller similarity scores. Hence objects close to the data centroid become less similar to other objects after centering, and no longer be hubs. In Section 4, we analyze the effect of centering on hubness in more detail.

3.1 Centered Gram matrix

Let $\mathbf{I}$ be an $n \times n$ identity matrix and $\mathbb{1}$ be an n-dimensional all-ones vector. The symmetric matrix $\mathbf{H} = \mathbf{I} - (1/n)\mathbb{1}\mathbb{1}^{\mathrm{T}}$ is called *centering matrix*, because the centered data matrix $\mathbf{X}^{\text{cent}} = [\boldsymbol{x}_1^{\text{cent}}, \cdots, \boldsymbol{x}_n^{\text{cent}}]$ can be computed by $\mathbf{X}^{\text{cent}} = \mathbf{XH}$ (Mardia et al., 1979).

The Gram matrix $\mathbf{K}^{\text{cent}}$ of the centered feature vectors, whose (i, j) element holds the inner product $\langle \boldsymbol{x}_i^{\text{cent}}, \boldsymbol{x}_j^{\text{cent}} \rangle$, can be calculated from the original Gram matrix $\mathbf{K}$ by

$$\mathbf{K}^{\text{cent}} = \left(\mathbf{X}^{\text{cent}}\right)^{\mathrm{T}} \left(\mathbf{X}^{\text{cent}}\right) = \mathbf{H}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{H} = \mathbf{HKH}. \tag{3}$$

Eq. (3) implies that the original data matrix $\mathbf{X}$ is not needed to compute the centered Gram matrix $\mathbf{K}^{\text{cent}}$, provided that $\mathbf{K}$ is given. It is hence possible to use the so-called *kernel trick*; i.e., centering can be applied even if data matrix $\mathbf{X}$ is not available but the similarity of objects can be measured by a kernel function in an implicit feature space.

4 Theoretical analysis of the effect of centering on hubness

We now analyze why objects most similar to the centroid tend to be hubs in the dataset, and give an explanation as to why centering may suppress the emergence of hubs.

4.1 Before centering

Consider a dataset of m-dimensional feature vectors, with each vector $\boldsymbol{x} \in \mathbb{R}^m$ generated independently from a distribution with a finite mean vector $\boldsymbol{\mu}$. In other words, objects $\boldsymbol{x}$ in this dataset are drawn from a distribution $P(\boldsymbol{x})$, i.e.,

$$\boldsymbol{x} \sim P(\boldsymbol{x}),$$

and

$$\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{x}] = \int \boldsymbol{x} \, dP(\boldsymbol{x}) \tag{4}$$

where $\mathbb{E}[\cdot]$ denotes the expectation of a random variable.

We will use the following elementary lemma on the distributions of inner product subsequently.

Lemma 1. *Let $\boldsymbol{a} \in \mathbb{R}^m$ be a fixed vector, and $\boldsymbol{x} \in \mathbb{R}^m$ be an object sampled according to distribution $P(\boldsymbol{x})$. Then the inner product $\langle \boldsymbol{a}, \boldsymbol{x} \rangle$ follows a distribution with mean $\langle \boldsymbol{a}, \boldsymbol{\mu} \rangle$.*

Proof. From the linearity of the inner product and Eq. (4), we obtain

$$\begin{aligned} \mathbb{E}[\langle \boldsymbol{a}, \boldsymbol{x} \rangle] &= \int \langle \boldsymbol{a}, \boldsymbol{x} \rangle \, dP(\boldsymbol{x}) \\ &= \langle \boldsymbol{a}, \int \boldsymbol{x} \, dP(\boldsymbol{x}) \rangle = \langle \boldsymbol{a}, \boldsymbol{\mu} \rangle. \qquad \square \end{aligned}$$

Now, imagine that we have an object $\boldsymbol{x}$ sampled from $P(\boldsymbol{x})$, and we want to compute its nearest neighbor in a dataset. Let $\boldsymbol{h}$ and $\boldsymbol{\ell}$ be two fixed objects in the dataset, such that the inner product to the true mean $\boldsymbol{\mu}$ is higher for $\boldsymbol{h}$ than for $\boldsymbol{\ell}$, i.e.,

$$\langle \boldsymbol{h}, \boldsymbol{\mu} \rangle - \langle \boldsymbol{\ell}, \boldsymbol{\mu} \rangle > 0. \tag{5}$$

We are interested in which of $\boldsymbol{h}$ and $\boldsymbol{\ell}$ is more similar to $\boldsymbol{x}$ (in terms of inner product), or in other words, the difference of two inner products

$$z = \langle \boldsymbol{h}, \boldsymbol{x} \rangle - \langle \boldsymbol{\ell}, \boldsymbol{x} \rangle = \langle \boldsymbol{h} - \boldsymbol{\ell}, \boldsymbol{x} \rangle. \tag{6}$$

Because $\boldsymbol{x}$ is a random variable, so is z. Let $Q(z)$ be the distribution of z; i.e., $z \sim Q(z)$.

Using Lemma 1 with $\boldsymbol{a} = \boldsymbol{h} - \boldsymbol{\ell}$, together with Eq. (5), we have

$$\mathbb{E}[z] = \langle \boldsymbol{h} - \boldsymbol{\ell}, \boldsymbol{\mu} \rangle = \langle \boldsymbol{h}, \boldsymbol{\mu} \rangle - \langle \boldsymbol{\ell}, \boldsymbol{\mu} \rangle > 0. \qquad (7)$$

Note that the above statement is only concerned about the mean, so it does not in general assure that

$$\langle \boldsymbol{h}, \boldsymbol{x} \rangle > \langle \boldsymbol{\ell}, \boldsymbol{x} \rangle \qquad (8)$$

holds with high probability; there is a chance that a small number of outliers are inflating the mean. To assure that inequality (8) holds with probability greater than $1/2$ for instance, the median rather than the mean of the distribution $Q(z)$ must be greater than 0.

If the distribution $Q(z)$ is symmetric, the median occurs at the same point as the mean, and the above claim holds. Indeed, if the components of $\boldsymbol{x}$ are generated independently from (possibly non-identical) normal distributions, we can show that $Q(z)$ also obeys a normal distribution. Because it is a symmetric distribution, we can safely say that in this case, Eq. (8) holds with probability greater than $1/2$.

For a general non-symmetric distribution with a finite variance, the median is known to be within the standard deviation of the mean (Mallows, 1991), so we could still say that Eq. (8) is likely to hold if $\langle \boldsymbol{h} - \boldsymbol{\ell}, \boldsymbol{\mu} \rangle$ is sufficiently large compared to the standard deviation.

Now, if we let $\boldsymbol{h}$ be the object in a given dataset with the highest similarity (inner product) to the mean $\boldsymbol{\mu}$, and let $\boldsymbol{\ell}$ be any other object in the set, then we see from the above discussion that $\boldsymbol{h}$ is likely to have higher similarity to $\boldsymbol{x}$, a test sample drawn from distribution $P(\boldsymbol{x})$. Because this holds for any $\boldsymbol{\ell}$ in the dataset, the conclusion is that the objects in the dataset most similar to $\boldsymbol{\mu}$ are likely to become hubs.

4.2 After centering

Next let us investigate what happens if the dataset is centered. Let $\bar{\boldsymbol{x}}$ be the sample (empirical) mean given by Eq. (1). After centering, the similarity of $\boldsymbol{x}$ with each of the two fixed objects $\boldsymbol{h}$ and $\boldsymbol{\ell}$ are evaluated by $\langle \boldsymbol{h} - \bar{\boldsymbol{x}}, \boldsymbol{x} - \bar{\boldsymbol{x}} \rangle$ and $\langle \boldsymbol{\ell} - \bar{\boldsymbol{x}}, \boldsymbol{x} - \bar{\boldsymbol{x}} \rangle$, respectively. Their difference z^{cent} is given by

$$\begin{aligned} z^{\text{cent}} &= \langle \boldsymbol{h} - \bar{\boldsymbol{x}}, \boldsymbol{x} - \bar{\boldsymbol{x}} \rangle - \langle \boldsymbol{\ell} - \bar{\boldsymbol{x}}, \boldsymbol{x} - \bar{\boldsymbol{x}} \rangle \\ &= \langle \boldsymbol{h} - \boldsymbol{\ell}, \boldsymbol{x} - \bar{\boldsymbol{x}} \rangle \\ &= \langle \boldsymbol{h} - \boldsymbol{\ell}, \boldsymbol{x} \rangle - \langle \boldsymbol{h} - \boldsymbol{\ell}, \bar{\boldsymbol{x}} \rangle \\ &= z - \langle \boldsymbol{h} - \boldsymbol{\ell}, \bar{\boldsymbol{x}} \rangle. \end{aligned}$$

The last equality follows from Eq. (6). By definition we have $z \sim Q(z)$, and since $\langle \boldsymbol{h} - \boldsymbol{\ell}, \bar{\boldsymbol{x}} \rangle$ is a constant,

$$z^{\text{cent}} = z - \langle \boldsymbol{h} - \boldsymbol{\ell}, \bar{\boldsymbol{x}} \rangle \sim Q(z + \langle \boldsymbol{h} - \boldsymbol{\ell}, \bar{\boldsymbol{x}} \rangle).$$

In other words, the shape of the distribution does not change, but the mean is shifted to

$$\begin{aligned} \mathbb{E}[z^{\text{cent}}] &= \mathbb{E}[z] - \langle \boldsymbol{h} - \boldsymbol{\ell}, \bar{\boldsymbol{x}} \rangle \\ &= \langle \boldsymbol{h} - \boldsymbol{\ell}, \boldsymbol{\mu} \rangle - \langle \boldsymbol{h} - \boldsymbol{\ell}, \bar{\boldsymbol{x}} \rangle \\ &= \langle \boldsymbol{h} - \boldsymbol{\ell}, \boldsymbol{\mu} - \bar{\boldsymbol{x}} \rangle, \end{aligned}$$

where $\mathbb{E}[z]$ is given by Eq. (7). If the sample mean $\bar{\boldsymbol{x}}$ is close enough to the true mean $\boldsymbol{\mu}$, i.e., $\bar{\boldsymbol{x}} \approx \boldsymbol{\mu}$, we have an approximation

$$\mathbb{E}[z^{\text{cent}}] = \langle \boldsymbol{h} - \boldsymbol{\ell}, \boldsymbol{\mu} - \bar{\boldsymbol{x}} \rangle \approx 0. \qquad (9)$$

Thus, if the median and the mean of distribution $Q(z)$ are again not far apart, Eq. (9) suggests that $\boldsymbol{h} - \bar{\boldsymbol{x}}$ and $\boldsymbol{\ell} - \bar{\boldsymbol{x}}$ are about equally likely to be more similar to $\boldsymbol{x} - \bar{\boldsymbol{x}}$; i.e., neither has a greater chance to become a hub.

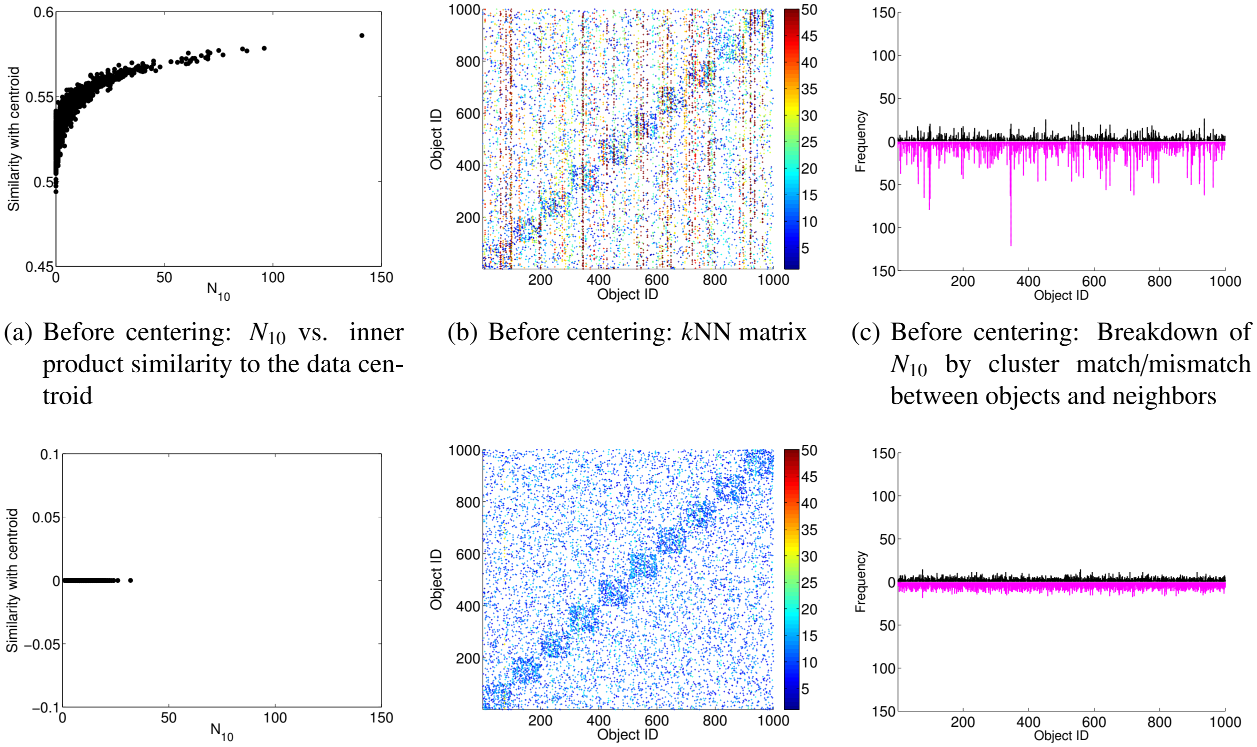
5 Hubs in multi-cluster data

In this section, we discuss emergence of hubs when the data consists of multiple clusters. In fact, the analysis of Section 4 is distribution-free, and thus also applies to the case of multi-modal $P(\boldsymbol{x})$. However, one might still argue that objects similar to the data centroid should hardly occur in that case. Using both synthetic and real datasets, we demonstrate below that even in multi-cluster data, objects that are only slightly more similar to the data mean (centroid) may emerge as hubs.

5.1 Synthetic data

5.1.1 Data generation

We generated a high-dimensional multi-cluster dataset by modeling it as a mixture of ten von Mises-Fisher distributions (Mardia and Jupp, 2000) in



(a) Before centering: N_{10} vs. inner product similarity to the data centroid

(b) Before centering: kNN matrix

(c) Before centering: Breakdown of N_{10} by cluster match/mismatch between objects and neighbors

(d) After centering: N_{10} vs. inner product similarity to the data centroid

(e) After centering: kNN matrix

(f) After centering: Breakdown of N_{10} by cluster match/mismatch between objects and neighbors

Figure 1: 300-dimensional synthetic data. (a), (d): scatter plot of the N_{10} value of objects and their similarity to centroid. (b), (e): kNN matrices. The points are colored according to the N_{10} value of object x; warmer colors indicate higher N_{10} values. (c), (f): the number of times (y-axis) an object (whose ID is on the x-axis) appears in the 10 nearest neighbors of objects of the same cluster (black bars), and those of different clusters (magenta).

$\mathbb{R}^{300}$. The von Mises-Fisher distribution is a distribution of unit vectors (it can roughly be thought of as a normal distribution on a unit hypersphere), so for objects (feature vectors) sampled from this distribution, inner product reduces to cosine similarity.

We sampled[1] 100 objects from each of the ten distributions (clusters), and made a dataset of 1,000 objects in total.

The von Mises-Fisher distribution has two parameters, the mean direction vector $\boldsymbol{\mu}$, and the concentration parameter κ characterizing how strongly the population is concentrated around the direction $\boldsymbol{\mu}$. We set $\kappa = 500$ for all ten distributions, but the mean directions $\boldsymbol{\mu}$ were made distinct; all mean direction vectors had 30 components set to 0.5 while the remaining 270 components were set to 1, but the 30 components with value 0.5 were chosen to be distinct among the ten clusters. This configuration assures that all ten mean directions have the same angle from the all-ones vector $[1, \ldots, 1]^{\mathrm{T}}$, which is the direction of the mean of the entire data distribution.

Note that even though all sampled objects reside on the surface of the unit hypersphere, the data centroid lies not on the surface but inside the hypersphere. And after centering, the length of the feature vectors may vary from one another, but we do not normalize these vectors; i.e., object similarity is measured by raw inner product, not by cosine.

[1] We used the random sampling code available at http://people.kyb.tuebingen.mpg.de/suvrit/work/progs/movmf.html (Banerjee et al., 2005).

5.1.2 Correlation between hubness and centroid similarity

The scatter plot in Figure 1(a) shows the correlation between the degree of hubness (N_{10}) of an object and its inner product similarity to the data centroid. The N_{10} value of an object is defined as the number of times the object appears in the 10 nearest neighbors of other objects in the dataset. It was used in (Radovanović et al., 2010a) to measure the degree of hubness of individual objects.

The plot clearly shows that the hub objects (i.e., those with high N_{10}) consist of objects that are similar to the centroid. Figure 1(d) shows the scatter plot after the data is centered, created in the same way as Figure 1(a). The similarity to the centroid is uniformly 0 as a result of centering, and no objects have an N_{10} value greater than 33.

5.1.3 Influence of hubs on objects in different clusters

The kNN matrix of Figure 1(b) depicts the kNN relations with $k = 10$ among objects before centering. In this matrix, both the x- and y- axes represent the ID of the objects. If object x is in the 10 nearest neighbors of object y, a point is plotted at coordinates (x, y). As a result, there are exactly $k = 10$ points in each row. The color of points indicates the degree of hubness of object x; warmer color represents higher N_{10} value of the object.

In this matrix, object IDs are sorted by the cluster the objects belong to. Hence in the ideal case in which the k nearest neighbors of every object consist genuinely of objects from the same cluster, only the diagonal blocks would be colored, and off-diagonal areas would be left blank.

As Figure 1(b) shows, the actual situation is far from ideal, even though ten diagonal blocks are still identifiable. The presence of many warm colored vertical lines suggests that many hub objects appear in the 10 nearest neighbors of other objects that are not in the same cluster as the hubs. Thus these hubs may have a strong influence on the kNN prediction of other objects.

Figure 1(e) shows the kNN matrix after centering. The warm colored lines have disappeared, and the diagonal blocks are now more visible.

The bar graphs of Figures 1(c) and (f) plot the N_{10} value of each object (whose ID is on the x-axis). Recall that N_{10} is the number of times an object appears in the 10 nearest neighbors of other objects. The bar for each object is broken down by whether the object and its neighbors belong to the same cluster (black bar) or in different clusters (magenta bar). In terms of kNN classification, having a large number of nearest neighbors with the same class improves the classification performance, so longer black bars and shorter magenta bars are more desirable.

Before centering (Figure 1(c)), hub objects with large N_{10} values are similar not only to objects belonging to the same cluster (as indicated by black bars), but also to objects belonging to different clusters (magenta bars). After centering (Figure 1(f)), the number of tall magenta bars decreases.

Before centering, 22.7% of the 10 nearest neighbors of an object have the same class label as the object (as indicated by the ratio of the total height of black bars relative to that of all bars in Figure 1(c)). After centering, the percentage increases to 31.6%.

5.2 Real dataset

We did the same analysis as Sections 5.1.2–5.1.3 to a real dataset with multiple-cluster structure: the Reuters Transcribed dataset. This multi-class document classification dataset has ten classes, and each class roughly forms a cluster. We will also use this dataset in an experiment in Section 7.2.

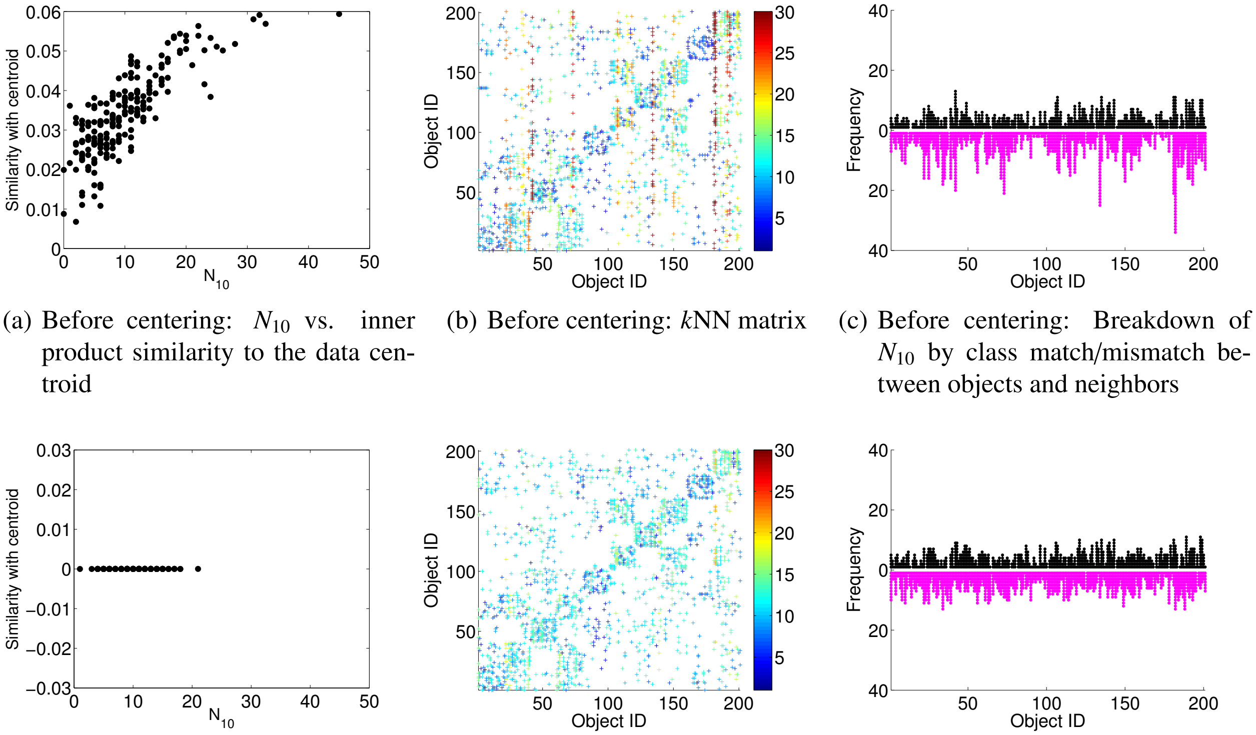
The results are shown in Figure 2. We can observe the same trends as we saw in Figure 1 for the synthetic data: positive correlation between hubness (N_{10}) and inner product with the data centroid before centering; hubs appearing in the nearest neighbors of many objects of different classes; and both are reduced after centering.

The ratio of the height of black bars to that of all bars in Figure 2(c) is 38.4% before centering, whereas it improves to 41.0% after centering (Figure 2(f)).

6 Hubness weighted centering

Centering shifts the origin of the space to the data centroid, and objects similar to the centroid tend to become hubs. Thus in a sense, centering can be interpreted as an operation that shifts the origin towards hubs.

In this section, we extrapolate this interpretation,



(a) Before centering: N_{10} vs. inner product similarity to the data centroid

(b) Before centering: kNN matrix

(c) Before centering: Breakdown of N_{10} by class match/mismatch between objects and neighbors

(d) After centering: N_{10} vs. inner product similarity to the data centroid

(e) After centering: kNN matrix

(f) After centering: Breakdown of N_{10} by class match/mismatch between objects and neighbors

Figure 2: Reuters Transcribed data.

and move the origin more actively towards hub objects in the dataset, rather than towards the data centroid. To this end, we consider *weighted centering*, a variation of centering in which each object is associated with a weight, and the origin is shifted to the weighted mean of the data. Specifically, we define the weight of an object as the sum of the similarities (inner products) between the object and all objects, regarding this sum as the index of how likely the object can be a hub.

6.1 Weighted centering

In weighted centering, we associate weight w_i to each object i in the dataset, and move the origin to the weighted centroid

$$\bar{\boldsymbol{x}}^{\text{weighted}} = \sum_{i=1}^{n} w_i \boldsymbol{x}_i$$

where $\sum_{i=1}^{n} w_i = 1$ and $0 \le w_i \le 1$ for $i = 1, \dots, n$. Thus, object $\boldsymbol{x}$ is mapped to a new feature vector

$$\boldsymbol{x}^{\text{weighted}} = \boldsymbol{x} - \bar{\boldsymbol{x}}^{\text{weighted}} = \boldsymbol{x} - \sum_{i=1}^{n} w_i \boldsymbol{x}_i.$$

Notice that the original centering formula (2) is recovered by letting $w_i = 1/n$ for all $i = 1, \dots, n$.

Weighted centering can also be kernelized by using the weighted centering matrix $\mathbf{H}(\boldsymbol{w}) = \mathbf{I} - \mathbb{1}\boldsymbol{w}^{\mathrm{T}}$ in place of $\mathbf{H}$ in Eq. (3). The resulting Gram matrix is

$$\mathbf{K}^{\text{weighted}} = \mathbf{H}(\boldsymbol{w})\mathbf{K}\mathbf{H}(\boldsymbol{w})^{\mathrm{T}}. \tag{10}$$

6.2 Similarity-dependent weighting

To move the origin towards hubs more aggressively, we place more weights on objects that are more likely to become hubs. This likelihood is estimated by the similarity of individual objects to all objects in the data set.

Let d_i be the sum of the similarity between object $\boldsymbol{x}_i$ and all objects in the dataset. So,

$$d_i = \sum_{j=1}^{n} \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = n \langle \boldsymbol{x}_i, \frac{1}{n} \sum_{j=1}^{n} \boldsymbol{x}_j \rangle.$$

As seen from the last equation, d_i is proportional to the similarity (inner product) between object $\boldsymbol{x}_i$ and the data centroid.

Now we define $\{w_i\}_{i=1}^n$ from $\{d_i\}_{i=1}^n$ by

$$w_i = \frac{d_i^\gamma}{\sum_{j=1}^n d_j^\gamma},$$

where γ is a parameter controlling how much we emphasize the effect of d_i. Setting $\gamma = 0$ results in $w_i = 1$ for every i, and hence is equivalent to normal centering. When $\gamma > 0$, weighted centering moves the origin closer to the objects with a large d_i than normal centering would.

7 Experiments

We evaluated the effect of centering in two natural language tasks: word sense disambiguation (WSD) and document classification. We are interested in whether hubs are actually reduced after centering, and whether the performance of *k*NN classification is improved.

Throughout this section, $\mathbf{K}$ denotes cosine similarity matrix; i.e., inner product of feature vectors normalized to unit length; $\mathbf{K}^{\text{cent}}$ denotes the centered similarity matrix computed by Eq. (3) from $\mathbf{K}$; $\mathbf{K}^{\text{weighted}}$ denotes its hubness weighted variant given by Eq. (10). Depending on context, these symbols are also used to denote *k*NN classifiers using respective similarity measures.

For comparison, we also tested two recently proposed approaches to hub reduction: transformation of the base similarity measure (in our case, $\mathbf{K}$) by Mutual Proximity (Schnitzer et al., 2012)[2], and the one (Suzuki et al., 2012) based on graph Laplacian kernels. Since the Laplacian kernels are defined for graph nodes, we computed them by taking the cosine similarity matrix $\mathbf{K}$ as the weighted adjacency (affinity) matrix of a graph. For Laplacian kernels, we computed both the regularized Laplacian kernel (Chebotarev and Shamis, 1997; Smola and Kondor, 2003) with several parameter values, as well as the commute-time kernel (Saerens et al., 2004), but present only the best results among these kernels.

7.1 Word sense disambiguation

7.1.1 Task and dataset

In the WSD experiment, we used the dataset for the Senseval-3 English Lexical Sample (ELS) task (Mihalcea et al., 2004). It is a collection of sentences containing 57 polysemous words, and each of these sentences is annotated with a gold standard sense of the target word. The goal of the ELS task is to build a classifier for each target word, which, given a context around the word, predicts a sense from the known set of senses.

We used a basic bag-of-words representation for the context surrounding a target word (Mihalcea, 2004; Navigli, 2009). A context is thus represented as a high-dimensional feature vector holding the tf-idf weighted frequency of words[3] in context.

7.1.2 Compared methods

We applied *k*NN classification using cosine similarity $\mathbf{K}$, and its four transformed similarity measures: centered similarity $\mathbf{K}^{\text{cent}}$, its weighted variant $\mathbf{K}^{\text{weighted}}$, Mutual Proximity and graph Laplacian kernels. The sense of a test object was predicted by voting from the k training objects most similar to the test object, as measured by the respective similarity measures.

We used leave-one-out cross validation within the training data to tune neighborhood size k for the *k*NN classification and the voting scheme, i.e., either (unweighted) majority vote, or weighted vote in which votes from individual objects are weighted by their similarity score to the test objects. We also selected parameter γ in $\mathbf{K}^{\text{weighted}}$ and the best graph Laplacian kernel among the regularized Laplacian and commute time kernels using the training data.

7.1.3 Evaluation

We computed two indices for each similarity measure: (i) skewness of the N_{10} distribution to evaluate

[2] We used the Matlab script downloaded from http://www.ofai.at/~dominik.schnitzer/mp/.

[3] We removed stop words listed in the on-line appendix of (Lewis et al., 2004).

Method	F1 score	Skewness
$\mathbf{K}$	60.3	4.55
$\mathbf{K}^{\text{cent}}$	64.0	1.19
$\mathbf{K}^{\text{weighted}}$	**64.8**	1.02
Mutual Proximity	63.0	1.00
Graph Laplacian	61.2	4.51
GAMBL (Decadt et al., 2004)	64.5	—

Table 1: WSD results: Macro-averaged F1 score (points) of the compared methods (larger is better) and empirical skewness of the N_{10} distribution for each similarity measure (smaller is better).

the emergence of hubs, and (ii) macro-averaged F1 score to evaluate the classification performance.

Skewness To evaluate the degree of hub emergence for each similarity measure, we followed (Radovanović et al., 2010a) and counted $N_k(\boldsymbol{x})$, the number of times object $\boldsymbol{x}$ occurs in the kNN lists of other objects in the dataset (we fix $k = 10$ below). The emergence of hubs in a dataset can then be quantified with *skewness*, defined as follows:

$$S_{N_k} = \frac{\mathbb{E}\left[(N_k - \mu_{N_k})^3\right]}{\sigma_{N_k}^3}.$$

In this equation, $\mathbb{E}[\,\cdot\,]$ denotes expectation, and μ_{N_k} and σ_{N_k} are the mean and the standard deviation of the N_k distribution, respectively.

When hubs exist in a dataset, the distribution of N_k is expected to skew to the right, and yields a large S_{N_k} (Radovanović et al., 2010a). In other words, similarity measures that yield smaller S_{N_k} are more desirable in terms of hub reduction.

Skewness can only be computed for each dataset, and in the WSD task, each target word has its own dataset. Hence we computed the skewness $S_{N_{10}}$ for each word and then took average.

Macro-averaged F1 score Classification performance was measured by the F1 score macro-averaged over all the 57 target words in the Senseval-3 ELS dataset. The standard Senseval-3 ELS scoring method is based on micro average, but we used macro average to make the evaluation consistent with skewness computation, which, as mentioned above, can only be computed for each dataset (i.e., word).

Dataset	#classes	#objects	#features
Reuters Transcribed	10	201	2730
Mini Newsgroups	20	2000	8811

Table 2: Document classification datasets: Number of classes, data size, and number of features.

7.1.4 Result

Table 1 shows the F1 scores and the skewness of the N_{10} distributions, macro averaged over the 57 target words. The table also includes the macro-averaged F1 score[4] of the GAMBL system, the best memory-based system participated in the Senseval-3 ELS task. Note however that GAMBL uses more elaborate features (e.g., part-of-speech of words) than just a plain bag-of-words used by other methods in this comparison. GAMBL also employs complex post-processing of the kNN outputs.

After centering ($\mathbf{K}^{\text{cent}}$ and $\mathbf{K}^{\text{weighted}}$) skewness became markedly smaller than that of the non-centered cosine $\mathbf{K}$. F1 score also improved with the decrease in skewness. In particular, weighted centering ($\mathbf{K}^{\text{weighted}}$) slightly outperformed GAMBL, though the difference was small. Recall however that $\mathbf{K}^{\text{cent}}$ and $\mathbf{K}^{\text{weighted}}$ only use naive bag-of-words features, unlike GAMBL.

7.2 Document classification

7.2.1 Task and dataset

Two multiclass document classification datasets were used: Reuters Transcribed and Mini Newsgroups, distributed at http://archive.ics.uci.edu/ml/. The properties of the datasets are summarized in Table 2.

7.2.2 Evaluation

The performance was evaluated by the F1 score (equivalent to accuracy in this task) of prediction using leave-one-out cross validation, due to the limited number of documents.

7.2.3 Compared methods

We used the cosine similarity as the base similarity matrix ($\mathbf{K}$). The centered similarity matrix ($\mathbf{K}^{\text{cent}}$) and its weighted variant ($\mathbf{K}^{\text{weighted}}$), Mutual

[4] The macro-averaged F1 of GAMBL was calculated from the per-word F1 scores listed in Table 1 of (Decadt et al., 2004).

Method	F1 score	Skewness
$\mathbf{K}$	56.7	1.61
$\mathbf{K}^{\text{cent}}$	**61.2**	0.11
$\mathbf{K}^{\text{weighted}}$	60.2	0.04
Mutual Proximity	60.2	−0.10
Graph Laplacian	57.2	0.37

(a) Reuters Transcribed

Method	F1 score	Skewness
$\mathbf{K}$	76.5	4.37
$\mathbf{K}^{\text{cent}}$	79.0	1.56
$\mathbf{K}^{\text{weighted}}$	**79.4**	1.68
Mutual Proximity	79.0	0.49
Graph Laplacian	77.6	2.13

(b) Mini Newsgroups

Table 3: Document classification results: F1 score (%) (larger is better) and skewness of the N_{10} distribution for each similarity measure (smaller is better).

Proximity, and graph Laplacian based kernels were computed from $\mathbf{K}$.

*k*NN classification was done in a standard way: The class of object $\boldsymbol{x}$ is predicted by the majority vote from $k = 10$ objects most similar to $\boldsymbol{x}$, measured by a specified similarity measure. The parameter k for the *k*NN classification, the voting scheme (i.e., either unweighted or weighted majority vote), γ in $\mathbf{K}^{\text{weighted}}$, and the best graph Laplacian kernel were selected by leave-one-out cross validation.

7.2.4 Result

Table 3 shows the F1 score and the skewness of the N_{10} distribution of the respective methods in document classification. Centered cosine ($\mathbf{K}^{\text{cent}}$) outperformed uncentered cosine similarity $\mathbf{K}$, and achieved an F1 score comparable to Mutual Proximity. Weighted centering ($\mathbf{K}^{\text{weighted}}$) further improved F1 on the Mini Newsgroups data.

8 Conclusion

We have shown that centering similarity matrices reduces the emergence of hubs in the data, and consequently improves the accuracy of nearest neighbor classification. We have theoretically analyzed why objects most similar to the mean tend to make hubs, and also proved that centering cancels the bias in the distribution of inner products, and thus is expected to reduce hubs.

In WSD and document classification tasks, *k*NN classifiers showed much better performance with centered similarity measures than non-centered ones. Weighted centering shifts the origin towards hubs more aggressively, and further improved the classification performance in some cases.

In future work, we plan to exploit the class distribution in the dataset to make more effective similarity measures; notice that the hubness weighted centering of Section 6 is an unsupervised method, in the sense that class information was not used for determining weights. We will investigate if more effective weighting can be done using this information.

Acknowledgments

We thank anonymous reviewers for helpful comments.

References

Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. 2005. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382.

P. Yu. Chebotarev and E. V. Shamis. 1997. The matrix-forest theorem and measuring relations in small social groups. *Automation and Remote Control*, 58(9):1505–1514.

Yihua Chen, Eric K. Garcia, Maya R. Gupta, Ali Rahimi, and Luca Cazzanti. 2009. Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research*, 10:747–776.

Bart Decadt, Véronique Hoste, Walter Daelemans, and Antal Van den Bosch. 2004. GAMBL, genetic algorithm optimization of memory-based WSD. In Rada Mihalcea and Phil Edmonds, editors, *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3)*, pages 108–112.

L. Eriksson, E. Johansson, N. Kettaneh-Wold, J. Trygg, C. Wikström, and S. Wold. 2006. *Multi- and Megavariate Data Analysis, Part 1, Basic Principles and Applications*. Umetrics, Inc.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*, pages 897–906, Honolulu, Hawaii, USA.

Douglas H. Fisher and Hans-Joachim Lenz, editors. 1996. *Learning from Data: Artificial Intelligence and*

Statistics V: Workshop on Artificial Intelligence and Statistics. Lecture Notes in Statistics 112. Springer.

Ruixin Guo and Sounak Chakraborty. 2010. Bayesian adaptive nearest neighbor. *Statistical Analysis and Data Mining*, 3(2):92–105.

Daniel Jurafsky and James H. Martin. 2008. *Speech and Language Processing*. Prentice Hall, 2nd edition.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: a new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Colin Mallows. 1991. Another comment on O'Cinneide. *The American Statistician*, 45(3):257.

K. V. Mardia and P. Jupp. 2000. *Directional Statistics*. John Wiley and Sons, 2nd edition.

K. V. Mardia, J. T. Kent, and J. M. Bibby. 1979. *Multivariate Analysis*. Academic Press.

Brij M. Masand, Gordon Linoff, and David L. Waltz. 1992. Classifying news stories using memory based reasoning. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '92)*, pages 59–65.

Rada Mihalcea, Timothy Chklovski, and Adam Kilgarriff. 2004. The Senseval-3 English lexical sample task. In Rada Mihalcea and Phil Edmonds, editors, *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3)*, pages 25–28, Barcelona, Spain.

Rada Mihalcea. 2004. Co-training and self-training for word sense disambiguation. In Hwee Tou Ng and Ellen Riloff, editors, *Proceedings of the 8th Conference on Computational Natural Language Learning (CoNLL '04)*, pages 33–40, Boston, Massachusetts, USA.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics: Human Language Technologies (ACL '08)*, pages 236–244, Columbus, Ohio, USA.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41:10:1–10:69.

Ali Mustafa Qamar, Éric Gaussier, Jean-Pierre Chevallet, and Joo-Hwee Lim. 2008. Similarity learning for nearest neighbor classification. In *Proceedings of the 8th International Conference on Data Mining (ICDM '08)*, pages 983–988, Pisa, Italy.

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010a. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11:2487–2531.

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. 2010b. On the existence of obstinate results in vector space models. In *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10)*, pages 186–193, Geneva, Switzerland.

Marco Saerens, François Fouss, Luh Yen, and Pierr Dupont. 2004. The principal components analysis of graph, and its relationships to spectral clustering. In *Proceedings of the 15th European Conference on Machine Learning (ECML '04)*, Lecture Notes in Artificial Intelligence 3201, pages 371–383, Pisa, Italy. Springer.

Dominik Schnitzer, Arthur Flexer, Markus Schedl, and Gerhard Widmer. 2012. Local and global scaling reduce hubs in space. *Journal of Machine Learning Research*, 13:2871–2902.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24:97–123.

Alexander J. Smola and Risi Kondor. 2003. Kernels and regularization on graphs. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, Proceedings*, Lecture Notes in Artificial Intelligence 2777, pages 144–158. Springer.

Anders Søgaard. 2011. Semisupervised condensed nearest neighbor for part-of-speech tagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL '11)*, pages 48–52, Portland, Oregon, USA.

Ikumi Suzuki, Kazuo Hara, Masashi Shimbo, Yuji Matsumoto, and Marco Saerens. 2012. Investigating the effectiveness of Laplacian-based kernels in hub reduction. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI-12)*, pages 1112–1118, Toronto, Ontario, Canada.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2010. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*, pages 948–957, Uppsala, Sweden.

Jigang Wang, Predrag Neskovic, and Leon N. Cooper. 2006. Neighborhood size selection in the *k*-nearest-neighbor rule using statistical confidence. *Pattern Recognition*, 39(3):417–423.

Kilian Q. Weinberger and Lawrence K. Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, pages 42–49, Berkeley, California, USA.