Mutual proximity graphs for improved reachability in music recommendation

Arthur Flexera and Jeff Stevensb

aAustrian Research Institute for Artificial Intelligence (OFAI), Austria; bGeorge Mason University, Virginia, USA

ABSTRACT
This paper is concerned with the impact of hubness, a general problem of machine learning in high-dimensional spaces, on a real-world music recommendation system based on visualisation of a k-nearest neighbour (knn) graph. Due to a problem of measuring distances in high dimensions, hub objects are recommended over and over again while anti-hubs are nonexistent in recommendation lists, resulting in poor reachability of the music catalogue. We present mutual proximity graphs, which are an alternative to knn and mutual knn graphs, and are able to avoid hub vertices having abnormally high connectivity. We show that mutual proximity graphs yield much better graph connectivity resulting in improved reachability compared to knn graphs, mutual knn graphs and mutual knn graphs enhanced with minimum spanning trees, while simultaneously reducing the negative effects of hubness.

ARTICLE HISTORY
Received 14 February 2017
Accepted 3 July 2017

KEYWORDS
Music recommendation; graphs; hubness; curse of dimensionality

1. Introduction
In graph theory, Ozaki, Shimbo, Komachi, and Matsumoto (2011) have observed that knn graphs often produce hubs, i.e. vertices with extremely high numbers of edges to other vertices. The same authors also made the connection to the phenomenon of ‘hubness’, which is a term used to describe a general problem of learning in high-dimensional spaces (Radovanović, Nanopoulos, & Ivanović, 2010). Hubness has been first noted in music information retrieval (MIR) (Aucouturier & Pachet, 2004), but is now acknowledged as a general machine learning problem and a new aspect of the curse of dimensionality (Radovanović et al., 2010; Schnitzer, Flexer, Schedl, & Widmer, 2012). While hubs appear very close to many other vertices, anti-hubs present as distant to all vertices. Both phenomena arise from the concentration of distance measures (Francois, Wertz, & Verleysen, 2007) in high-dimensional spaces.

This hubness phenomenon has been shown to negatively impact a real-world music recommendation system which has been built by our research team. This system uses visualisation of a knn graph to recommend music via a web interface. In previous work (Gasser & Flexer, 2009; Flexer, Gasser, & Schnitzer, 2010) we were able to show that hubness causes some songs to never appear in any nearest neighbour list, because hubs crowd the nearest neighbour’s lists and are therefore repeatedly recommended. As a result, only about two-thirds of the songs are reachable in the recommendation interface, i.e. over a third of the songs are never recommended.

Further analysis of the knn graph shows that only less than a third of the songs are likely to be recommended, since only those are part of one large strongly connected subgraph. We have already applied ‘mutual proximity’ (Schnitzer et al., 2012), a hubness reduction method, to improve this situation, but have not yet explored this topic in graph theoretical terms.

This paper presents a thorough analysis of the properties of this music recommendation system based on knn graphs, as well as three alternative graph construction methods, all in the light of hubness. We present related work in Section 2, the data for the empirical evaluation in Section 3, the graph construction and evaluation methods in Section 4, the results of our analysis in Section 5 and finally discuss and conclude in Sections 6 and 7. This paper is an extended version of a previous workshop contribution (Flexer & Stevens, 2016), with the main novel contributions being introduction of an additional
graph construction method using a minimum spanning tree, a random walk analysis and overall increased depth of treatment.

2. Related work

In our overview of related work, we concentrate on the hubness problem, with an emphasis on its relation to audio-based music retrieval and graph analysis. The music recommendation literature is rich in comprehensive resources such as Oscar Celma’s seminal book (Celma, 2010), surveys on recommendation via collaborative filtering and content-based models (Song, Dixon, & Pearce, 2012), via context-based information (Knees & Schedl, 2013), as well as a treatment of the topic in a recent road map on music information research (Serra et al., 2013). We refer the reader to the above resources for excellent perspectives on music recommendation technology. Note that our work deals with content-based music recommendation computed from audio information directly, where the recommendation approach is based on graphical analysis of interactively built playlists of songs (Section 3).

Before discussing the hubness phenomenon, we would like to make clear that the notion of hub vertices in networks (Barabási & Albert, 1999) predates the notion of hubness in MIR (Aucouturier & Pachet, 2004) and machine learning (Radovanović et al., 2010). Given the focus of this paper, we will start our review of related literature with MIR and machine learning, and then proceed to the graph literature.

In MIR, hubness was first noted as a problem in audio-based music recommendation (Aucouturier & Pachet, 2004), more specifically that certain songs were being recommended conspicuously often in nearest neighbour-based playlists. Hubness has since gained attention in the machine learning community, where it has been described as a new aspect of the curse of dimensionality and a general problem of learning in high-dimensional spaces (Radovanović et al., 2010; Schnitzer et al., 2012). Hubness is related to the phenomenon of concentration of distances, where all pairwise distances are approximately the same for dimensionality approaching infinity (François et al., 2007). Radovanović et al. (2010) presented the argument that for any finite dimensionality, some points are expected to be closer to the centre of all data than other points, and are at the same time closer, on average, to all other points. Such points closer to the centre have a high probability of being hubs, i.e. of appearing in nearest neighbour lists of many other points. Points which are farther away from the centre have a high probability of being anti-hubs, i.e. points that never appear in any nearest neighbour list. It should be noted that the same authors, in their analysis of the influence of hubness on nearest neighbour methods, already made the connection to $k$-nearest neighbour graphs.

Along the same lines, the relationship between hubness and graph theory has also been established by viewing a recommender system as a directed graph (Celma, 2010; Seyerlehner, Flexer, & Widmer, 2009). Every item in a database is a vertex of the graph and every recommendation of an item $v_j$ from an item $v_i$ is represented as a directed edge $e_{ij}$ leading from $v_i$ to $v_j$ thereby constructing a nearest neighbour graph (Eppstein, Paterson, & Yao, 1997). This point of view connects the hubness problem to the rich literature on graph theory and complex network analysis (see Cohen & Havlin, 2010 for an overview). It also links to results in MIR that demonstrate hubs are distributed along a scale-free distribution (Aucouturier & Pachet, 2008), which is an important property of many real-world graphs. So-called ‘scale-free networks’ contain hub vertices which have many more connections than other vertices and therefore the network as a whole exhibits a power-law distribution in the number of links incident to a node (Barabási & Albert, 1999). In such networks, hubs guarantee a short path length between all vertices and as such are seen beneficial rather than harmful because they promote network resiliency. A graph theory-based analysis of an existing music recommendation service has shown limited accessibility of its audio catalogue due to hub and anti-hub nodes (Flexer et al., 2010), where anti-hub vertices are the logical opposites of hubs, i.e. vertices with in-degrees of zero. Related studies (Celma, 2010; Celma & Cano, 2008) on artist similarity showed that artist similarity graphs based on audio content, collaborative filtering or expert advice all exhibit the hubness characteristic of scale-free networks.

Hubness has also been shown to have a negative impact on many more tasks including classification (Dinu, Lazariadou, & Baroni, 2015; Radovanović et al., 2010; Shigeto, Suzuki, Hara, Shimbo, & Matsumoto, 2015), regression (Buza, Nanopoulos, & Nagy, 2015), nearest neighbour-based recommendation (Flexer, Schnitzer, & Schlüter, 2012) and retrieval (Schnitzer, Flexer, & Tomašev, 2014), clustering (Schnitzer & Flexer, 2015; Tomašev, Radovanović, Mladenić, & Ivanović, 2013), visualisation (Flexer, 2015) and outlier detection (Flexer, 2016; Radovanović, Nanopoulos, & Ivanović, 2015). Hubness also affects data from diverse domains including multimedia (text, music, images, speech), biology and general machine learning (see Feldbauer & Flexer, 2016; Radovanović et al., 2010; Schnitzer et al., 2012 for large-scale empirical studies). It is important to note that the degree of distance 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 degrees of freedom of the submanifold in which the data space can be represented (Francois et al., 2007).

Basically three different approaches have been proposed to reduce hubness and its negative effects: re-scaling (Schnitzer et al., 2012; Tomašev & Mladenić, 2014) of the distance space, centreing of the data (Hara et al., 2015), using $l^p$ norms different than Euclidean $l^2$ norm (Flexer & Schnitzer, 2015). Of special interest for this paper are re-scaling approaches like mutual proximity (MP) and local scaling (LS), which aim at repairing asymmetric non-mutual nearest neighbour relations. The asymmetric relations are a direct consequence of the asymmetric non-mutual nearest neighbour relations. The proximity (MP) and local scaling (LS), which aim at repairing asymmetric non-mutual nearest neighbour relations. The asymmetric relations are a direct consequence of the presence of hubs. A hub $y$ is the nearest neighbour of $x$, but the nearest neighbour of the hub $y$ is another point $a$ ($a \neq x$). This is because hubs are by definition nearest neighbours to very many data points but only one data point can be the nearest neighbour to a hub. The principle of the scaling algorithms is to rescale distances to enhance symmetry of nearest neighbours. A small distance between two objects should be returned only if their nearest neighbours are mutual. Application of MP and LS resulted in a decrease of hubness and an accuracy increase in $k$-nearest neighbour classification on 30 real-world data-sets including text, image and music data (Schnitzer et al., 2012).

Yet another approach towards hubness reduction, and the most relevant work for this paper, is an application of mutual $k$-nearest neighbour graphs to semi-supervised classification of natural language data (Ozaki et al., 2011). The authors observe that knn graphs often produce hubs and that mutual knn graphs can reduce this hub effect and at the same time increase classification accuracy. In what follows, we will essentially apply the mutual knn graphs proposed by the authors in a music recommendation setting and compare it to mutual proximity graphs, which we will introduce in Section 4.1. This connects to our own previous work on using mutual proximity for reducing hubness in music recommendation (Schnitzer et al., 2012) by adding a graph theoretic perspective that has been missing so far.

3. Data

For our analysis, we use data from the real-world music discovery system FM4 Soundpark (http://fm4.orf.at/soundpark), a web platform run by the Austrian public radio station FM4, where artists can upload and present their music free of charge. Website visitors can listen to and download all the music at no cost, with most recent uploads being displayed at the top of the website. Whereas chronological publishing is suitable to promote new releases, older releases tend to quickly disappear from the user’s attention. In the case of the FM4 Soundpark, this has resulted in users mostly listening to music that is advertised on the front-page, and therefore missing the full breadth of the music database. To allow a more intuitive and appealing access to the full database regardless of the song’s publication date, we implemented a recommendation system using a content-based music similarity measure (Gasser & Flexer, 2009). This similarity measure is based on timbre information computed from the audio.

Each time a song is uploaded, 3 min of raw 22,050 Hz mono audio signals from the centre of the submission is used for analysis and similarity ranking. We divide the raw audio data into overlapping frames of short duration and transform them to Mel Frequency Cepstrum Coefficients (MFCC), resulting in a spectrally smoothed and perceptually meaningful representation of the audio signal. MFCCs are a standard technique for computation of spectral similarity in music analysis (see e.g. Logan, 2000). The frame size for computation of MFCCs for our experiments was 46.4 ms (1024 samples). We used the first 20 MFCCs for all our experiments. The MFCCs of each song are modelled via a single Gaussian with full covariance matrix (Mandel & Ellis, 2005). To compute a distance value between two Gaussians, the symmetric Kullback–Leibler (SKL) divergence is used (see Gasser & Flexer, 2009 for more details on both MFCCs and SKL). This results in an $N \times N$ symmetric distance matrix $D$ for the data-set, with $N = 7665$ songs. The data-set of 7665 songs is a snapshot of the constantly growing Soundpark database. As new songs continue to be added, new distance measures are computed and the knn graph is updated (some edges added, and some deleted). As previously mentioned, 20 MFCCs are used to generate a Gaussian song model. Each of these models contains 190 covariances and 20 mean values, resulting in an extrinsic dimension of 210 dimensions. The reader is referred to Mandel and Ellis (2005) for specifics. The intrinsic dimensionality (Levina & Bickel, 2005) of the data computed from the distance matrix $D$ is 14. Whereas this may sound small, previous work (Schnitzer et al., 2012) has shown that data with intrinsic dimensionality as low as 9 can already be negatively impacted by hubness.

The database of songs is organised in a rather coarse genre taxonomy, where the artists themselves are able to choose which of six genre labels best describe their music. The artists are allowed to choose one or two of the genre labels, hence the following percentages add up to more than 100%: 37.6% of all songs belong to genre Pop, 46.3%
to Rock, 44.0% to Electronica, 14.3% to Hip-Hop, 19.7% to Funk, 5.3% to Reggae (see Flexer et al., 2010 for more detail concerning the database).

The recommendation system has been implemented as a web player which can be accessed from within an artist’s web page on the Soundpark website by clicking on one of the artist’s tracks. In addition to offering the usual player interface (start, stop, skipping forward/backward), the system recommends songs similar to the selected track in a text list and in a graph-based visualisation (see Figure 1). The graph visualisation displays an interactively constructed nearest neighbour graph (number of nearest neighbours = 5), where a song is shown as a large centred circle with edges connecting it to the most similar (nearest neighbor) songs. Circle sizes for similar songs are inversely proportional to edge distance from the selected song, to a maximum two. For every song a user clicks, its nearest neighbours are displayed, thereby interactively constructing a visualisation of the knn graph, with circles representing vertices and lines representing edges. Vertices having an edge distance greater than two from the central starting vertex are trimmed. Songs which are nearest neighbours to more than one other song in the display are only shown the first time they appear as a nearest neighbour, thereby making it possible that less than five neighbours are displayed for some of the songs. The original Soundpark system was characterised by users listening mainly to the latest uploaded songs. However, introduction of this audio-based recommender system resulted in an increase of distinct song accesses by users listening mainly to the latest uploaded songs. The original Soundparksystem was characterised as a nearest neighbour, thereby making it possible that edges of vertices having an edge distance greater than two from the central starting vertex are trimmed. Songs which are nearest neighbours to more than one other song in the display are only shown the first time they appear as a nearest neighbour, thereby making it possible that less than five neighbours are displayed for some of the songs.

**4. Methods**

The user interface of the music recommender has been implemented as a visualisation of a knn graph showing the k = 5 most similar songs to the currently selected track. In what follows we will describe the construction of the knn graph and three alternative construction methods in Section 4.1. Measures to evaluate these graphs are presented in Section 4.2.

**4.1. Graph construction**

A graph G = (V, E) is defined via a finite set of vertices \( V = \{v_1, v_2, \ldots, v_n\} \) and edges \( E = \{e_1, e_2, \ldots, e_m\} \). In our case, the vertices correspond to songs displayed in the music recommender interface and the edges connect similar songs.

**k-nearest neighbour graphs (knn):** knn graphs are a very common graph construction technique, where an edge \( e_{ij} \) is placed between \( v_i \) and \( v_j \) if \( v_j \) is among the \( k \) nearest neighbours of \( v_i \). We use the distance matrix \( D \) defined in Section 3 to compute an adjacency matrix \( A \). If, according to \( D \), a song with the index \( j \) is among the five nearest neighbours of a song with index \( i \), then \( A_{ij} = 1 \), otherwise \( A_{ij} = 0 \). An edge \( e_{ij} \) exists between two vertices \( v_i \) and \( v_j \) if \( A_{ij} = 1 \). Please note that the adjacency matrix \( A \) is not symmetric and the resulting graph is therefore a directed knn graph. For a knn graph, the out-degree \( d^+(v_i) \) of a vertex \( v_i \), i.e. the number of edges having origin \( v_i \), is always equal to \( k \) (\( k = 5 \) in this work). The in-degree \( d^−(v_i) \), i.e. the number of edges with destination \( v_i \), is in the interval \([0, N - 1]\), with \( N \) being the number of vertices.

**Mutual k-nearest neighbour graphs (muknn):** muknn graphs (see e.g. Ozaki et al., 2011) are a simple extension of knn graphs wherein only bidirectional edges are retained, i.e. an edge \( e_{ij} \) exists only if \( A_{ij} = A_{ji} = 1 \), which indicates song \( j \) and song \( i \) are in each other’s nearest neighbour lists. The resulting muknn graph is therefore a subset of the corresponding knn graph and versions of it have already been applied to reduce hubness (Ozaki et al., 2011). Both in- and out-degrees of all vertices are in the interval \([0, k]\), with the additional constraint that \( d^+(v_i) = d^−(v_i) \) for all vertices in the muknn graph.

**Mutual k-nearest neighbour graphs plus minimum spanning tree (muknn+msp):** many edges can be deleted during construction of muknn graphs, sometimes yielding poorly connected vertices. It has already been suggested (Ozaki et al., 2011) to add a minimum spanning tree (Kruskal, 1956) (msp) to the muknn graph to improve connectivity. A msp is a graph of minimum weight connecting all vertices. It is a subset of an edge-weighted complete graph without any self-loops or cycles. We build a msp using the distance matrix \( D \) as input, i.e. the edge weights correspond to the distances between songs as described in Section 3. We use the classical Prim’s algorithm (Prim, 1957) to construct the msp. The resulting msp is an undirected graph, or equivalently a bidirectional graph with \( A_{ij} = A_{ji} = 1 \), i.e. undirected edges are replaced by two corresponding directed edges. The compound muknn+msp graph consists of all edges which are either part of the muknn or msp graph. Both in- and out-degrees of all vertices are in the interval \([1, N - 1]\), with the lower limit of one being due to the fact that now every vertex is part of one fully connected graph. The additional constraint that \( d^+(v_i) = d^−(v_i) \) for all vertices still holds.

**Mutual proximity graphs (mp):** for construction of mp graphs we first rescale the distance matrix \( D \) using the hubness reduction method mutual proximity (MP) (Schnitzer et al., 2012). MP rescales the original distance space so that two objects sharing similar nearest neighbours are more closely tied to each other, while two objects with dissimilar neighbourhoods are repelled.
from each other. MP reinterprets the distance of two objects as a mutual proximity (a probability) in terms of their distribution of distances. To compute MP, we assume that the distances \( D_{x,y} \) 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 neighbour of \( x \), given their distance \( D_{x,y} \) and the probability distribution \( P(X) \). MP is then defined as the probability that \( y \) is the nearest neighbour of \( x \) given \( P(Y) \) and \( x \) is the nearest neighbour of \( y \) given \( P(Y) \):

\[
MP(D_{x,y}) = P(X > D_{x,y} \cap Y > D_{y,x}).
\]

To compute MP in our experiments, we use the empirical distribution instead of assuming a functional form. Changing from similarities to distances, computation of \( 1 - MP \) yields aso-called secondary distance matrix \( D^{MP} \), which is then used to construct an adjacency matrix \( A^{MP} \). Using \( A^{MP} \), we construct a knn graph exactly as described above, which we now call a mp graph, since it is based on distances rescaled via MP. Just as the knn graph, the mp graph has an out-degree \( d^+(v_i) = k \) for all vertices, and in-degrees \( d^-(v_i) \) in the interval \([0, N - 1]\).

### 4.2. Graph evaluation measures

**k-occurrence (\#hub, \#anti, maxhub, hubness \( S^k \))**: the number of times the song occurs in the first \( k \) nearest neighbours of all the other songs in the database (so-called \( k \)-occurrence \( O^k \)) is provided as an indication of the hubness of a given song (see e.g. Aucouturier & Pachet, 2004). This \( k \)-occurrence is of course identical to the in-degree \( d^- \) of a vertex. The mean of \( O^k \) across all songs in a database is equal to \( k \). Any \( k \)-occurrence significantly bigger than \( k \) therefore indicates existence of a hub. We select \( k = 5 \) because our music recommender always shows the five most similar songs. As has been done before (Schnitzer et al., 2012), we define that any song with \( O^k > 5k = 25 \) is a hub, any song with \( O^k = 0 \) is an anti-hub, any song with \( O^k > 0 \land O^k \leq 5k = 25 \) is a so-called normal object. We compute the maximum \( k \)-occurrence, maxhub (i.e. the biggest hub), the number of songs of which the \( k \)-occurrence is more than five times \( k \) (\#hub), and for which it is equal to zero indicating an anti-hub (\#anti). To further characterise the strength of the hubness phenomenon, we use a hubness measure (Radovanović et al., 2010), which is defined as the skewness of the distribution of \( k \)-occurrences, \( O^k \), of all objects in a data set:

\[
S^k = \frac{E[(O^k - \mu_{O^k})^3]}{\sigma_{O^k}^3}
\]

where \( \mu \) and \( \sigma \) are the mean and standard deviation of the \( k \)-occurrence (in-degree) histogram, respectively, and \( E \) is the expectation operator. A data-set having...
high hubness produces few hub objects with very high $k$-occurrence and many anti-hubs with $k$-occurrence of zero. This makes the distribution of $k$-occurrences positively skewed, which we use as a measure of hubness.

**Reachability (reach):** This is the percentage of songs from the whole database that are part of at least one of the recommendation lists, i.e. $(1 - (#anti/N)) \times 100$. If a song is not part of any of the recommendation lists of size $k = 5$, it cannot be reached using our recommendation function, unless it is directly chosen as a starting song by a user.

**Number of edges ($\#edges$):** we give the number of edges $e$ in a graph $G$, i.e. $|E|$ in graph notation.

**Self-avoiding random walk ($\bar{\omega}$):** to simulate user behaviour with the recommendation system, we use a so-called self-avoiding random walk approach (Madras & Slade, 1993). A random walk (see e.g. Brandes & Erlebach, 2005) $\omega(1, 2, \ldots, l)$ in a directed graph $G = (V, E)$ is a Markov chain, where the probability to move from one vertex $v_i$ at time point $t$ to vertex $v_j$ at time point $t + 1$ is only conditional on the state of the system at time $t$. A self-avoiding random walk has the additional property not to allow any vertex to be member of the random walk more than once:

$$Pr[\omega(t + 1) = v_j | \omega(t) = v_i] = \begin{cases} \frac{1}{d^+(v_i, t)} & \text{if } (e_{ij} \in E) \land (v_j \notin \omega(1 \ldots t)) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $d^+(v_i, t)$ is the number of edges leaving vertex $v_i$ at time $t$. The temporal dependence of $d^+(v_i)$ arises from the self-avoiding criterion wherein $v_i$’s out-degree count decreases each time a node in it nearest neighbour list is visited during the random walk. This means that at every step, the random walk picks a random edge leaving the current vertex and follows it to the destination vertex of that edge, but it is forbidden to return to any vertex $v_i$ that already is a member of the random walk $\omega$ so far. This last condition of course renders the walk non-Markovian and is introduced to guarantee that in our simulation every song (vertex) is only visited once. After all it seems unlikely that within the course of a rather short interaction, users would return to songs they have already listened to. We simulate 100,000 self-avoiding random walks of length $l = 20$, thereby visiting 20 songs which gives about one hour of music listening, assuming an average duration of 3 min per song. For every walk, a starting vertex is picked at random. This altogether simulates a simple interaction of users with the recommendation system, allowing us to compute average access statistics for all songs in the database. Note that $d^+(v, t) \in [0, d^+(v, 0)]$ due to the self-avoiding construct, so the walk can terminate early if $v$’s nearest neighbours have all been previously visited.

**Strongly connected component (scc, #scc, scc):** For our nearest neighbour graph, a strongly connected component (SCC) is a subgraph where every song is connected to all other songs travelling along the directed nearest neighbour connections. We use Tarjan’s algorithm (Tarjan, 1972) to find all SCC-graphs in our nearest neighbour graph with $k = 5$. We report the size of the largest strongly connected component as a percentage of the whole database (scc), the number of additional strongly connected components (#scc) and their average size ($\bar{scc}$) in number of vertices.

**$\phi$-edge ratio ($\phi$):** for a labelled graph $(G, l)$, a $\phi$-edge is any edge $e_{ij}$ for which $l_i \neq l_j$ (Ozaki et al., 2011), i.e. in our case for which the genres $l$ of the songs corresponding to the vertices $v_i$ and $v_j$ do not match. Since our vertices can have one or two labels (genres), each $\phi$-edge is assigned a value according to the percentage of non-overlapping genres between vertices:

$$\phi(i,j) = \left(1 - \frac{|l_i \cap l_j|}{|l_i \cup l_j|}\right) \times 100 \quad (4)$$

with $l_i$ ($l_j$) being a set of all genre labels for the song corresponding to vertex $v_i$ ($v_j$) and $|.|$ counting the number of members in a set. Therefore, $\phi(i,j)$ is defined as one minus the number of shared genre labels divided by the set size of the union of sets $l_i$ and $l_j$, times 100. The latter is done to account for songs with two genre labels as compared to only one genre label. The range of values for $\phi$ is between 0 and 100. The $\phi$-edge ratio is the average of $\phi$-values for all edges in a graph. If the edges in a graph reflect the semantic meaning of its labelled vertices, it will have a low $\phi$-edge ratio.

## 5. Results

Our analysis results using the evaluation indices defined in Section 4.2 are given in Table 1. We will now present the results structured into analysis of hubness, random walks, strongly connected components and semantics in Sections 5.1–5.4.

### 5.1. Hubness analysis

The knn graph, which is the basis of the FM4 Soundpark, shows considerable hubness. The number of hubs (#hub) is 291 and there are 2661 anti-hubs (#anti), which results in a very high skewness of the distribution of $k$-occurrences $S^k = 13.97$. Hubness values of $S^k > 1.4$ can already be seen as problematic (Schnitzer et al., 2012).
As a consequence, only 65.28% of vertices are reachable. About a third of the data are therefore never being recommended. The biggest hub \( \text{maxhub} \) alone appears 419 times in the recommendation lists, i.e., it is recommended for 5.47% of the songs in the database. If the songs were randomly distributed across the recommendation lists, the expected number for \( \text{maxhub} \) would be 5 or only .065%.

Looking at the muknn graph, all hubs are gone but there is a high number of anti-hubs (4566). Hubs vanish as a consequence of the strict requirement of mutual neighbourhoods when building the muknn graph, since now every vertex can only be connected to at most five other vertices. Therefore, the number for \( \text{maxhub} \) is down to 5 and hubness \( S^k \) down to 1.43. This also has the consequence that many edges from the knn graph are being deleted \( n_{\text{edges}} = 5790 \) instead of \( 7665 \times 5 = 38,325 \) for knn), resulting in the very high number of anti-hubs.

To improve this situation and promote better connectivity, we have added a minimum spanning tree to the muknn graph yielding the muknn+msp graph. Since a minimum spanning tree by definition connects all the vertices in a graph, the number of anti-hubs is zero and the reachability is at 100%. Compared to the muknn graph, the msp adds 11,826 edges yielding a total of \( n_{\text{edges}} = 17,616 \) for the muknn+msp graph. These additional edges increase the number of hubs to \( n_{\text{hub}} = 23 \). This is still much better than 291 for the knn graph but also worse than the zero hubs of the muknn graph. The biggest hub \( \text{maxhub} \) is also larger at 145, but still smaller than the original 419 of the knn graph. As a further consequence, hubness \( S^k \) is back up to 10.15.

The mp graph shows a low number of 2 hubs and a moderate number of 641 anti-hubs, with a highly increased reachability of 91.62%. The biggest hub \( \text{maxhub} \) has a moderate size of 30 and hubness \( S^k \) is down to only .91. Since the mp graph is essentially a knn graph built from an adjacency matrix based on rescaled secondary distances, the number of edges is identical to that of the knn graph \( n_{\text{edges}} = 38,325 \).

In Figure 2, we have plotted the distribution of \( k \)-occurrences for the four different graph methods separately. As outlined above, a \( k \)-occurrence of a song is equal to the in-degree \( d^− \) of the corresponding vertex. In the plots in Figure 2, in-degrees ranging from \( d^− = 0 \) to \( d^− > 25 \) are given on the x-axis, with the number of vertices exhibiting these in-degrees on the y-axis. Bars on the very left of each plot therefore show the number of anti-hubs \( n_{\text{anti}} (d^− = 0) \), bars on the far right the number of hubs \( n_{\text{hub}} (d^− > 25) \). As can be seen in the plot for the knn graph (left plot, top row), there is a considerable amount of both anti-hubs and hubs.

For the muknn graph (right plot, top row), there are no in-degrees larger than 5, but at the same time also many more anti-hubs from enforcement of the mutuality criterion. Looking at the plot for the muknn+msp graph (left plot, bottom row), we can see that all anti-hubs are gone (i.e., graph has one connected component), but a few hubs have re-appeared and there exist many vertices with an in-degree equal one. These vertices with only a single edge linking to it are a direct consequence of adding a minimum spanning tree, since a single in-going edge per vertex can be sufficient to allow for full connectivity of the graph. Looking at the plot for the mp graph (right plot, bottom row), one can see a much more balanced distribution of in-degrees around the expected value of five, with much fewer anti-hubs with \( d^− = 0 \) and only few vertices with in-degrees larger than 10 or 15.

5.2. Random walk analysis

As to the results of the random walk analysis, the average lengths of the random walks \( \bar{w} \) are given in Table 1. As outlined in Section 4.2, we simulated 100,000 self-avoiding random walks with random starting vertices. Given that our database is comprised of 7665 songs, on average every vertex corresponding to a song was chosen as a starting vertex more than 10 times. The number of times every vertex was visited during all the walks ranged from 3 to 50,186 times, with both of these extreme values obtained for knn graphs. Each walk was set to a length of 20 songs, but one of the graphs (muknn) has vertices with no outgoing edges which lead to early termination of random walks. The requirement of the random walks to be self-avoiding also contributed to early termination, because it prohibits usage of edges back to a vertex that had already been used during the same walk.

As for the knn graph, the average length \( \bar{w} \) is 19.68 indicating that almost all walks terminated normally, i.e., after the full intended length of 20. As outlined in Section 4.1, the out-degree \( d^+ \) of all vertices in the knn graph is equal five. A premature termination of a walk can therefore only happen if all five vertices linked via outgoing edges have already been visited earlier in a walk, which seems to happen rather rarely. There is a quite different situation for the muknn graph, with the average walk length being only 2.40. As has been explained in Section 4.1, for muknn graphs the in- and out-degree of every vertex is equal and between zero and five. As can be seen in Figure 2 and explained above, there is a large number of anti-hubs with \( d^− = d^+ = 0 \). Such an anti-hub vertex can only be part of a random walk if it is randomly chosen as a starting vertex, which then immediately leads to termination of the walk and a length of only one. For the muknn+msp graph, the average length \( \bar{w} \) is 4.50, which is some improvement
Table 1. Analysis of knn, muknn, muknn+msp and mp graphs.

|               | knn | muknn | muknn+msp | mp  |
|---------------|-----|-------|-----------|-----|
| maxhub        | 419 | 5     | 145       | 30  |
| #hub          | 291 | 0     | 23        | 2   |
| #anti         | 2661| 4566  | 0         | 641 |
| $S^k$         | 13.97| 1.43 | 10.15     | 0.91|
| reach         | 65.28%| 40.43%| 100.00% | 91.62%|
| #edges        | 38,325| 5790 | 17,616    | 38,325|
| $\bar{w}$    | 19.68| 2.40 | 4.50      | 19.50|
| scc           | 29.11%| 11.89%| 100.00% | 85.26%|
| #scc          | 408  | 652  | 0         | 102 |
| $\bar{sCC}$  | 2.87 | 3.36 | –         | 2.87 |
| $\phi$       | 55.78%| 37.79%| 52.10%    | 52.20%|

Figure 2. Distribution of $k$-occurrences for knn, muknn, muknn+msp and mp graphs. Notice the strong skewness in knn, muknn, and muknn+msp, while the mp graph is much more evenly dispersed with few hubs and anti-hubs.

compared to muknn graphs. As can be seen in Figure 2 and outlined above, muknn+msp graphs do not contain any anti-hubs but very many vertices with $d^- = d^+ = 1$. As an additional constraint, $d^-(v_i) = d^+(v_i)$, i.e. for every outgoing edge $e_{ij}$ from vertex $v_i$ there exists an in-going edge $e_{ji}$ returning to the origin vertex $v_i$. For the many vertices with in- and out-degree equal one this means that they become terminating vertices due to the self-avoiding property of the random walks, since a walk is prohibited to return to where it originated from and additional outgoing edges do not exist. To visualise these phenomena, imagine a linear string of nodes from any vertex. The random walk can only proceed towards the end of the string once the first step in that direction has been taken, hence it will eventually terminate. Finally for the mp graph, the average length $\bar{w}$ is 19.50, almost as high as for the knn graph. This is due to the fact the out-degree $d^+$ of all vertices is equal five, just as for the knn graph.

In Figure 3, we have plotted the percentage of vertices visited in all 100,000 walks of length 20 which are normal, hub or anti-hub vertices (as defined at the beginning
of Section 4.2) as black, dark grey and light grey bars. Since some of the walks are shorter than 20 due to premature termination (see paragraph above), we also give the percentage of overall aggregate walk length that is missing as white bars. All four bars together sum up to 100%. In addition, we show the percentage of normal, hub and anti-hub vertices in the database of songs (rightmost group of bars in Figure 3, label ‘expected’). If all walks in a graph would be randomly distributed across all vertices without any preference of vertices, we would observe exactly these percentages shown as ‘expected’: 61.49% normal, 3.80% hubs, 34.72% anti-hubs.

We will now compare the results for the different graphs to these expected values. Looking at the results for the knn graph (left-most group of bars in Figure 3, label ‘knn’), we can see that more than 57% of the visited vertices are hubs, about 39% are normal vertices and less than 2% are anti-hubs. It is therefore clear that a user would spend most of their time listening to hub songs, namely 57% of the listening time would be spent listening to 3.80% of the database (291 songs of 7665 in absolute numbers). Turning to the results for the muknn graph (label ‘muknn’), the most striking observation is that almost 88% of all walks are missing due to early termination. As for the rather small amount of actual simulated random walks, most of the time is spent at normal vertices and there are no hub vertices at all. The results for the muknn+msp graph (label ‘muknn+msp’) are only a little improved with 77% of all walks missing and most of the actual walks spent at normal vertices (14%), with hubs (4%) and anti-hubs (5%) following. The results for the mp graph shows real improvement (label ‘mp’), with less than 3% of all walk lengths missing and about 73% spent for normal vertices, 16% for hubs and 7% for anti-hubs. Comparing this to the expected values, more time should be spent for anti-hubs (34.72%) to ensure all songs can be reached, and still less for hubs (3.80%), which shows that there still is room for further improvement.

5.3. Strongly connected component analysis

Looking at the size of the largest strongly connected component (scc) in Table 1, one can see that it contains only 29.11% of the vertices for knn, which further decreases to 11.89% for muknn. Aside from this largest scc, there exist large numbers of additional sccs (#scc) of very small average size (scc) for both knn (408 with average size of 2.87) and muknn (652 with average size of 3.36). This means that it is very likely that a user of the recommendation system will spend most of their time listening to songs within the largest scc, which comprises only one-third of the data for knn, or just about one-tenth for muknn. Due to the nature of minimum spanning trees, the largest scc of the muknn+msp graph contains all of the vertices and there are no additional smaller sccs. Therefore, every vertex in the graph can be reached from all other vertices. The scc for the mp graph is much larger than those of the knn and muknn graphs, comprised of 85.26% of the vertices, with 102 extra sccs of average size 2.87. It seems that due to hubness reduction, many vertices now connect to others instead of hubs, so the mp graph exhibits a much higher connectivity than the original knn graph.

5.4. Semantic analysis

Looking at the percentage of edges connecting vertices with different labels (\(\phi\) in Table 1), we can see that all three methods, muknn, muknn+msp and mp yield improvements compared to knn. Whereas the \(\phi\)-edge ratio is 55.75% for knn, this improves to 37.79% for muknn and slightly improves to 52.10% for muknn+msp and to 52.20% for mp. This means that the hubness reduction in all three alternative graphs also respects the semantic meaning of the data when deviating from the original knn graph. Only muknn graphs show a very strong improvement in this regard, which is due to muknn graphs having far less, but more semantically correct, edges.

6. Discussion

We would now like to summarise and discuss our main findings which were described in Section 5.

Our first result is that \(k\)-nearest neighbour graphs, as used in our music recommendation system, are negatively impacted by hubness. In corroborating earlier results (Schnitzer et al., 2012), we have found that due...
to hubness, only two-thirds of the music catalogue is reachable at all, while only about a third of the songs are likely to be listened to according to the size of the largest strongly connected component. Based on our random walk analysis, we found that the situation is even more severe, since almost 60% of simulated listening time is spent listening to less than 4% of the whole catalogue. All this is due to a very skewed distribution of in-degrees in the knn graph, with hub vertices having very high numbers of in-degrees.

Our second result is that mutual nearest neighbour graphs are able to reduce the negative effects of hubness, but at the cost of very poor overall connectivity. Muknn graphs do not contain any hub vertices, but very many anti-hub vertices with an in-degree equal zero. As a consequence, the reachability is down to 40% and just about 12% are likely to being listened to, which is both even lower than results for knn graphs. The random walk analysis showed that the vast majority of walks terminate prematurely. On average a simulated listening session lasts less than 3 songs instead of the intended 20. This is all due to the very strict requirement of mutuality of edges between vertices, which eliminates all hub vertices but at the same time deletes 85% of all edges from the original knn graph creating many new anti-hubs and overall poor connectivity.

Our third result is that adding a minimum spanning tree to mutual nearest neighbour graphs is not sufficient to fix the connectivity issues of muknn graphs. The resulting muknn+msp graph by its very definition guarantees full reachability of 100%, but the random walk analysis makes it clear that the enhanced connectivity still is rather weak. The large majority of walks still terminate prematurely, which on average, reduces the simulated listening session to 5 songs instead of the intended 20. The compound muknn+msp graph does contain more edges than the muknn graph, but very often what used to be anti-hub vertices are now vertices with in- and out-degree equal only one, i.e. single edges leading to and from vertices.

Our fourth result is that mutual proximity graphs, as proposed in this paper, are able to decisively reduce the negative impact of hubness on knn graphs. The mp graph shows only two small hub vertices instead of 291 (a 99% reduction), as well as only few anti-hubs resulting in a reachability of more than 90%. About 80% of songs are now likely to be listened to according to the size of the largest strongly connected component. The random walk analysis shows that only 16% of the listening time is spent on what used to be hub songs, which is a great improvement compared to the 60% as observed for knn graphs. It is important to note that mp graphs keep the exact same overall number of edges as knn graphs; therefore, the average length of simulated listening sessions is almost 20 songs, just as for knn graphs.

Our final result is that all three alternative graphs respect the semantic meaning of the data, as measured as the proportion of edges between vertices with matching genre labels. Whereas muknn+msp and mp graphs show modest enhancements in this respect, muknn graphs allow for more improvement, but at the cost of deleting so many edges that the overall connectivity is quite poor.

As to future work, the first step will be to actually replace the knn graph in the real life Soundpark music recommendation system with a mutual proximity graph. Parallel to this update of the system, we will be able to measure differences in user behaviour like an expected increase of distinct song accesses, showing the improved reachability which we already demonstrated in our offline analysis in Section 5. This will also make it possible to measure any change of the users’ satisfaction with the updated system based on comparing questionnaires administered before and after the system update. Since mutual knn graphs can also be used for semi-supervised classification (Ozaki et al., 2011), we are also planning to employ mutual proximity graphs for this task. In the case of music recommendation, this could be useful to propagate genre labels to vertices which represent songs without genre information. It will also be interesting to explore usefulness of mutual proximity graphs in other domains where hub vertices can be found, going beyond mere MIR.

7. Conclusion

We have presented mutual proximity graphs, which are a new type of nearest neighbour graph able to decisively reduce hubs, vertices with extremely high numbers of edges. Whereas the related mutual $k$-nearest neighbour graphs are able to completely prevent formation of hub vertices, they result in graphs with poor overall connectivity. Adding a minimum spanning tree to the mutual knn graph to help, in this respect, only yields very little improvement. Whereas mutual knn graphs have the strict requirement of connecting only vertices which belong to each other’s nearest neighbours, mutual proximity graphs mitigate nearest neighbour asymmetries in a more flexible probabilistic way. We showed that mutual proximity graphs can improve a real-world music recommendation system, but future work should also explore usefulness of this new approach in other scenarios where hub vertices can be found.

Funding

This work was supported by the Austrian Science Fund (FWF) [grant number P27082].
References

Aucouturier, J.-J., & Pachet, F. (2004). Improving timbre similarity: How high is the sky. Journal of Negative Results in Speech and Audio Sciences, 1(1), 1–13.

Aucouturier, J.-J., & Pachet, F. (2008). A scale-free distribution of false positives for a large class of audio similarity measures. Pattern Recognition, 41, 272–284.

Barabási, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. Science, 286, 509–512.

Brandes, U., & Erlebach, T. (Eds.). (2005). Network analysis: Methodological foundations (Vol. 3418). Berlin Heidelberg: Springer Science & Business Media.

Buza, K., Nanopoulos, A., & Nagy, G. (2015). Nearest neighbor regression in the presence of bad hubs. Knowledge-Based Systems, 86, 250–260.

Celma, O. (2010). Music recommendation and discovery: The long tail, long fail, and long play in the digital music space. Heidelberg: Springer Science & Business Media.

Celma, O., & Cano, P. (2008). From hits to niches? or how popular artists can bias music recommendation and discovery. In 2nd Workshop on Large-scale Recommender Systems and the Netflix Prize Competition (ACM KDD). New York, NY: ACM.

Cohen, R., & Havlin, S. (2010). Complex networks: Structure, robustness and function. Cambridge: Cambridge University Press.

Dinu, G., Lazaridou, A., & Baroni, M. (2015). Improving zero-shot learning by mitigating the hubness problem. In Proceedings of International Conference on Learning Representations, Workshop Track, San Diego, CA.

Eppstein, D., Paterson, M. S., & Yao, F. F. (1997). On nearest-neighbor graphs. Discrete & Computational Geometry, 17, 263–282.

Feldbauer, R., & Flexer, A. (2016). Centering versus scaling for hubness reduction. In Proceedings of the 25th International Conference on Artificial Neural Networks (ICANN’16), Part I (pp. 175–183). New York, NY: Springer International Publishing.

Flexer, A. (2015). Improving visualization of high-dimensional music similarity spaces. In Proceedings of the 16th International Society for Music Information Retrieval Conference, Malaga, Spain.

Flexer, A. (2016). An empirical analysis of hubness in unsupervised distance-based outlier detection. In Proceedings of 4th International Workshop on High Dimensional Data Mining (HDM), in conjunction with the IEEE International Conference on Data Mining (IEEE ICDM 2016), Barcelona, Spain.

Flexer, A., Gasser, M., & Schnitzer, D. (2010). Limitations of interactive music recommendation based on audio content. In Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound (pp. 96–102). New York, NY: ACM.

Flexer, A., & Schnitzer, D. (2015). Choosing $l^p$ norms in high-dimensional spaces based on hub analysis. Neurocomputing, 169, 281–287.

Flexer, A., Schnitzer, D., & Schlüter, J. (2012). A MIREX meta-analysis of hubness in audio music similarity. In Proceedings of the 13th International Society for Music Information Retrieval Conference, FEUP Edicoes, Porto, Portugal.

Flexer, A., & Stevens, J. (2016). Mutual proximity graphs for music recommendation. In Proceedings of the 9th International Workshop on Machine Learning and Music, Riva del Garda, Italy.

Francois, D., Wertz, V., & Verleysen, M. (2007). The concentration of fractional distances. IEEE Transactions on Knowledge and Data Engineering, 19, 873–886.

Gasser, M., & Flexer, A. (2009). Fm4 soundpark: Audio-based music recommendation in everyday use. In Proceedings of the 6th Sound and Music Computing Conference (pp. 161–166), Porto, Portugal.

Hara, K., Suzuki, I., Shimbo, M., Kobayashi, K., Fukumizu, K., & Radovanovic, M. (2015). Localized centering: Reducing hubness in large-sample data hubness in high-dimensional data. In Proceedings of the 29th AAAI Conference on Artificial Intelligence (pp. 2645–2651), Palo Alto, CA.

Knees, P., & Schedl, M. (2013). A survey of music similarity and recommendation from music context data. ACM Transactions on Multimedia Computing, Communications and Applications, 10(1), 2:1–2:21.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical society, 7, 48–50.

Levina, E., & Bickel, P. J. (2005). Maximum likelihood estimation of intrinsic dimension. In Advances in Neural Information Processing Systems (Vol. 17, pp. 777–784). Cambridge, MA: MIT Press.

Logan, B. (2000). Mel frequency cepstral coefficients for music modeling. In Proceedings of the 1st International Conference on Music Information Retrieval, Plymouth, MA.

Madras, N., & Slade, G. (1993). The self-avoiding walk. Boston: Birkhäuser.

Mandel, M. I., & Ellis, D. (2005). Song-level features and support vector machines for music classification. In Proceedings of the 6th International Conference on Music Information Retrieval (pp. 594–599), London.

Ozaki, K., Shimbo, M., Komachi, M., & Matsumoto, Y. (2011). Using the mutual k-nearest neighbor graphs for semi-supervised classification of natural language data. In Proceedings of the 15th Conference on Computational Natural Language Learning (pp. 154–162). Madison, WI: Omnipress.

Prim, R. C. (1957). Shortest connection networks and some generalizations. Bell System Technical Journal, 36, 1389–1401.

Radovanovic, M., Nanopoulos, A., & Ivanovic, M. (2010). Hubs in space: Popular nearest neighbors in high-dimensional data. Journal of Machine Learning Research, 11, 2487–2531.

Radovanovic, M., Nanopoulos, A., & Ivanovic, M. (2015). Reverse nearest neighbors in unsupervised distance-based outlier detection. IEEE Transactions on Knowledge and Data Engineering, 27, 1369–1382.

Schnitzer, D., & Flexer, A. (2015). The unbalancing effect of hubs on k-medoids clustering in high-dimensional spaces. In Proceedings of the International Joint Conference on Neural Networks (IJCNN). Killarney, Ireland: IEEE.

Schnitzer, D., Flexer, A., Schedl, M., & Widmer, G. (2012). Local and global scaling reduce hubs in space. Journal of Machine Learning Research, 13, 2871–2902.

Schnitzer, D., Flexer, A., & Tomašev, N. (2014). A case for hubness removal in high-dimensional multimedia retrieval. In Proceedings of the 36th European Conference on Information Retrieval (ECIR).
Serra, X., Magas, M., Benetos, E., Chudy, M., Dixon, S., Flexer, A., ... Widmer, G. (2013). Roadmap for Music Information ReSearch. In G. Peeters (Ed.). ISBN: 978-2-9540351-1-6.
Seyerlehner, K., Flexer, A., & Widmer, G. (2009). On the limitations of browsing top-N recommender systems. In Proceedings of the 3rd ACM Conference on Recommender Systems. New York, NY: ACM.
Shigeto, Y., Suzuki, I., Hara, K., Shimbo, M., & Matsumoto, Y. (2015). Ridge regression, hubness, and zero-shot learning. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 135–151). Berlin Heidelberg: Springer.
Song, Y., Dixon, S., & Pearce, M. (2012). A survey of music recommendation systems and future perspectives. In Proceedings of the 9th International Symposium on Computer Music Modeling and Retrieval. Berlin Heidelberg: Springer.
Tarjan, R. (1972). Depth first search and linear graph algorithms. SIAM Journal on Computing, 1, 146–160.
Tomašev, N., & Mladenić, D. (2014). Hubness-aware shared neighbor distances for high-dimensional k-nearest neighbor classification. In Proceedings of the 9th International Conference on Hybrid artificial intelligent systems (pp. 116–127). Switzerland: Springer.
Tomašev, N., Radovanović, M., Mladenić, D., & Ivanović, M. (2013). The role of hubness in clustering high-dimensional data. IEEE Transactions on Knowledge and Data Engineering, 26(3), 739–751.