Hubness as a case of technical algorithmic bias in music recommendation

Arthur Flexer*, Monika Dörfler[†], Jan Schlüter*, Thomas Grill*

* Austrian Research Institute for Artificial Intelligence (OFAI), Vienna, Austria

[†] Numerical Harmonic Analysis Group, Faculty of Mathematics, University of Vienna, Vienna, Austria

arthur.flexer@ofai.at, monika.doerfler@univie.ac.at, jan.schlueter@ofai.at, thomas.grill@ofai.at

Abstract—This paper tries to bring the problem of technical algorithmic bias to the attention of the high-dimensional data mining community. A system suffering from algorithmic bias results in systematic unfair treatment of certain users or data, with technical algorithmic bias arising specifically from technical constraints. We illustrate this problem, which so far has been neglected in high-dimensional data mining, for a real world music recommendation system. Due to a problem of measuring distances in high dimensional spaces, songs closer to the center of all data are recommended over and over again, while songs far from the center are not recommended at all. We show that these so-called hub songs do not carry a specific semantic meaning and that deleting them from the data base promotes other songs to hub songs being recommended disturbingly often as a consequence. We argue that it is the ethical responsibility of data mining researchers to care about the fairness of their algorithms in high-dimensional spaces.

Index Terms—hubness, technical algorithmic bias, algorithmic fairness, ethical responsibility

I. INTRODUCTION

Learning and data mining in high-dimensional spaces is challenging due to a number of phenomena that are commonly referred to as curse of dimensionality [5]. In this paper we want to show that high-dimensional data mining can cause problems of algorithmic fairness, or more specifically a problem of *technical algorithmic bias*. We will do this by analyzing and discussing a specific aspect of the curse of dimensionality, so-called hubness, in the context of automatic music recommendation where it causes unfair treatment of certain songs.

Hubness was first noted as a problem in music information retrieval (MIR), which is the interdisciplinary science of retrieving information from music. An analysis [3] of a music recommendation system showed that hub songs were recommended conspicuously often in nearest neighbor-based playlists, while other songs acting as anti-hubs were never recommended. Hubness is now understood as a general problem of learning and data-mining in high-dimensional spaces [36], [43]. Hubness is related to the phenomenon of concentration of distances, where all pairwise distances are approximately the same for dimensionality approaching infinity [20], casting doubts whether distances in high dimensions are meaningful at all [6], [28].

The term algorithmic bias [21] is used to describe systematic and repeatable errors that create unfair outcomes in computer experiments, i.e. generating one result for certain users or certain data and a different result for others. With machine learning now being at the heart of many advanced technologies pervading the lives of millions of people, such unfair treatment by algorithms could have a negative impact on a society-wide level. In this paper we want to raise awareness for another problem of algorithm fairness, that of *technical* algorithmic bias arising specifically from technical constraints [21] in high dimensions. Let us give an example by imagining a hypothetical music recommendation system, which for every query song recommends five random songs belonging to the same genre, with the genre label being given as metainformation. Imagine the algorithm choosing the five random songs using a flawed random number generator favoring data objects at the end of the alphabetically ordered song data base. The result would be recommendation of pre-dominantly artists with names starting with letters at the end of the alphabet. resulting in unfair treatment due to technical algorithmic bias.

In this paper we will analyze technical algorithmic bias in high dimensions in a real-life content-based music recommendation system. Due to a problem of learning in high dimensional spaces, in this system some data objects act as 'hubs', which means that they are abnormally close to many other data objects and thus some songs are recommended over and over while others are never recommended. We first present related work in Section II, data from the music recommendation system in Section III, methods of signal processing and machine learning in Section IV, results in Section V, before we discuss and conclude in Sections VI and VII.

II. RELATED WORK

In our section on related work, we first review results concerning hubness including those from the field of music information retrieval (MIR), where hubness has first been discovered, before we review work on algorithmic bias. MIR is the interdisciplinary science of retrieving information from music. It is a direct response to the need for entirely new methods for analyzing, describing, distributing, and presenting music due to the rapidly growing amount of music available in digital form, see [32], [39], [47] for the most recent comprehensive surveys of the field. Automatic music recommendation [10] based on some notion of music similarity, for

This work was supported by the Austrian Science Fund (FWF P27082) and the Vienna Science and Technology Fund (WWTF MA14-018).

which the music recommender we will introduce in Section III is an example, is an important goal of MIR.

Hubness was first noted as a problem in audio-based music recommendation [3], more specifically that certain hub songs are being recommended conspicuously often in nearest neighbor-based playlists, while other songs acting as antihubs are never recommended. Hubness is an aspect of the curse of dimensionality and a general problem of learning in high-dimensional spaces [36], [43]. Hubness is related to the phenomenon of concentration of distances, where all pairwise distances are approximately the same for dimensionality approaching infinity [20]. In [36] it has been argued, that in the high-dimensional but still finite case, some points are expected to be closer to the center than other points and are at the same time closer, on average, to all other points. This phenomenon is known as spatial centrality [36] and it is amplified by high dimensionality. Points closer to the center have a high probability of being hubs, i.e. of appearing in nearest neighbor lists of many other points. Points which are further away from the center have a high probability of being 'anti-hubs', i.e. points that never appear in any nearest neighbor list. For unimodal data, hubs are close to the global data center. Real-world data sets are often better described as a mixture of distributions, for which hubs tend to be close to the means of individual distributions. Since the exact mixture of distributions in real-world data is often unknown, k-means clusters [36] and local neighborhoods [25] have previously been used to describe spatial centrality in multimodal data. On the other hand, anti-hubs are typically far from centers and can be considered distance-based outliers [36].

It is important to note that the degree of concentration and hubness is linked to the intrinsic rather than extrinsic dimension of the data space. While extrinsic dimension is simply the actual length of the data vectors of interest, the intrinsic dimension [9] is the, often much smaller, number of dimensions necessary to represent a feature space without loss of information. Previous research [43] demonstrated that real world data with extrinsic dimensionality as small as 34 can already exhibit the negative effects of hubness.

For the real world music recommendation system which we will describe in Section III and analyze in this paper, it has been shown [19] that only two-thirds of the music catalogue is recommended and reachable at all, while only about a third of the songs are likely to being listened to. Based on a random walk analysis simulating potential users, it was 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. Hub objects are also often responsible for misclassification, i.e. the class labels of hub objects and the objects they are nearest neighbors to do not match [43].

Apart from music recommendation, hubness has already been shown to have a negative impact on many more tasks including classification [11], [36], [48], clustering [41], [52], visualization [15] and outlier detection [13], [14], [37]. The connection between hubness and the notion of hub vertices in graphs has also already been explored [1], [2], [19], [36]. There also exist a number of methods to reduce the negative effects of hubness, essentially re-scaling [43], [51] the distance space, centering the data [25], or using alternative distances [16] (see [12] for a comprehensive review and comparison).

The term algorithmic bias or, more generally, bias in computer systems is used to describe systematic and repeatable errors that create unfair outcomes in computer experiments, i.e. generating one result for certain users or certain data and different results for others in a reproducible way [21]. This bias is usually divided into pre-existing, emergent and technical bias.

When computer systems embody biases that exist independently and often prior to creation of a computer system, one speaks of pre-existing bias. Such bias can enter a system as a conscious decision but more often implicitly and unconsciously. Very often pre-existing bias is due to the choice of training data enforcing existing prejudice and discrimination. It has e.g. been shown [8] that application of machine learning to ordinary human language results in human-like semantic biases. The text corpora used for training contain certain imprints of historic societal biases, like e.g. prejudice toward race or gender [7], which are then replicated by the computer systems. For the field of MIR, the CompMusic project has broached a related issue, stating that most "research is being carried out with a western centered approach and as a result, most of our data models, cognition models, user models, interaction models, ontologies, etc., are culturally biased" [45].

Emergent bias becomes visible when real users are interacting with computer systems, most often because user interfaces are used by populations different from those intended by the interface designers. Examples from MIR are populations with different expertise or cultural backgrounds, as has also been discussed within the CompMusic project where different interfaces to MIR systems for different cultures have been advocated [45].

Technical algorithmic bias on the other hand arises from specifically technical constraints, which may be due to hardware, software or even peripherals. Technical bias may arise from rather trivial problems like the combination of imperfect random numbers and alphabetical order in a database as mentioned in Section I. It can also be much harder to grasp, e.g. in a legal expert system weighing data points to determine whether a defendant should accept a plea bargain. In such cases information which is hard to formalize, like the impact of emotion on a jury, is ignored or misrepresented in the expert system's weighing algorithm [21]. The concept of technical algorithmic bias, as well as computer bias in general, is closely related to a recent drive in data mining to give prospective users and stakeholders the right to an explanation for algorithmic output [23], allowing to better understand why a certain decision has been made.

In the research field of data mining, basically two solutions to the problem of algorithmic bias have been proposed: (i) discrimination discovery from data bases [33]; (ii) discrimination prevention via fairness-aware data mining [24]. To the best of our knowledge, the problem of technical algorithmic bias due to high dimensionality has not been discussed within the data mining community so far. The hubness problem itself has been researched and documented within the MIR community to a considerable extent [3], [17], [18], [29], [42], [43], but has not been understood as a problem of technical algorithmic bias. In the final report [47] of the most recent substantial re-evaluation of the research compass of the field of MIR ('Roadmap for Music Information ReSearch (MIReS)'), biases in training data are mentioned but not those related to algorithms. In a very recent overview article [26] on ethical dimensions of MIR technology, algorithmic bias is discussed for a hypothetical music recommender where "a large number of artists is never recommended [...] due to a lack of user data or other artifacts that are not completely understood". This paper aims at understanding such failures of music recommendation systems as a problem of technical algorithmic bias due to high dimensionality.

A question which still remains unanswered is what makes a specific piece of music act as a hub or anti-hub within a certain MIR system. The respective theory [36] suggests that hub objects are closer to the center of all data, but there exists no in-depth investigation how this relates to what is being observed in music data. It seems clear that given a certain representation, different songs will act as hubs, e.g. a timefrequency based music representation and one based on the raw audio temporal signal will result in very different data distributions and hence in very different hub songs. Previous research has already shown that the choice of how songs are modeled is an important factor as to whether the overall MIR system is prone to hubness or not [18].

This interplay of hubness, representation and algorithmic bias will be the topic of the rest of this paper.

III. DATA

For our analysis, we use data from a real-world music discovery system¹, 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. To allow a more appealing access to the full database regardless of the song's publication date, a recommendation system using a contentbased music similarity measure was implemented [22]. This similarity measure is based on timbre information computed from the audio. For our experiments we use a development data base of 16583 songs which is organized in a coarse genre taxonomy. The artists themselves choose which of the six genre labels 'Hip Hop, Reggae, Funk, Electronic, Pop and Rock' best describe their music. The distribution of genres is quite unbalanced with 'Electronic' and 'Rock' being the most prominent genres. This is representative of the full data base.

IV. METHODS

The user interface of the music recommender has been implemented as a visualization of a knn graph showing the k = 5 most similar songs to the currently selected track. We describe the feature processing used to model the timbre of a song in Section IV-A and how the distance $D_{KL}(p,q)$ between two songs p and q is computed in Section IV-B. The 5 songs with minimum distance D_{KL} to a query song are being recommended.

A. Signal processing

From the 22050Hz mono audio signals two minutes from the center of each song are used for further analysis. We divide the raw audio data into overlapping frames of short duration and use Mel Frequency Cepstrum Coefficients (MFCC) to represent the spectrum of each frame. MFCCs are a perceptually meaningful and spectrally smoothed representation of audio signals. MFCCs are a standard technique for computation of spectral similarity in music analysis (see e.g. [30]). The following steps compute the MFCCs:

- divide signal into short overlapping segments with frame size 46.4ms (1024 samples) and a hopsize of 23.2ms (512 samples)
- 2) apply a Hann window to each segment
- 3) compute power spectrum matrix using a Fast Fourier Transformation
- 4) transform the power spectrum to the mel-scale using a filter bank consisting of triangular filters
- 5) convert to decibel by taking the logarithm
- 6) apply discrete cosine transform to compress and smooth the mel power spectrum to 20 MFCCs

B. Computing similarity of songs

We now describe the approach to computing music similarity based on spectral similarity. For a given music collection of songs, it consists of the following steps:

- 1) for each song, compute MFCCs for short overlapping frames as described in Section IV-A
- 2) train a single Gaussian (G1) to model each of the songs
- compute a distance matrix between all songs using the Kullback-Leibler divergence between respective G1 models

Single Gaussian (G1) with full covariance are used to represent the MFCCs of each song [31]. For single Gaussians, $p(x) = \mathcal{N}(x; \mu_p, \sigma_p)$ and $q(x) = \mathcal{N}(x; \mu_q, \sigma_q)$, there is a closed form of the Kullback-Leibler divergence [34]:

$$KL_N(p||q) = 0.5 \log\left(\frac{\det(\Sigma_p)}{\det(\Sigma_q)}\right) + 0.5Tr\left(\Sigma_p^{-1}\Sigma_q\right) + 0.5\left(\mu_p - \mu_q\right)'\Sigma_p^{-1}\left(\mu_q - \mu_p\right) - \frac{d}{2}$$
(1)

where Tr(M) denotes the trace of the matrix M, $Tr(M) = \sum_{i=1..n} m_{i,i}$. Dropping constants and symmetrizing the divergence yields the following approximation [40]:

$$D_{KL}(p,q) = Tr\left(\Sigma_{p}^{-1}\Sigma_{q}\right) + Tr\left(\Sigma_{q}^{-1}\Sigma_{p}\right) + Tr\left(\left(\Sigma_{p}^{-1} + \Sigma_{q}^{-1}\right)(\mu_{p} - \mu_{q})(\mu_{q} - \mu_{p})'\right)$$
(2)

¹http://fm4.orf.at/soundpark

	#	mean	std	min	max
hub	653	51.63	4.55	44.10	76.36
normal	9977	68.60	67.01	46.49	2171.6
anti	5953	76.22	44.67	48.56	1784.9

 TABLE I

 The number (#) of hub, normal and anti-hub objects and their

 mean distance to the center of the data, plus the standard

 deviation and max and min distances.

Please note that this approximation is symmetric, i.e. $D_{KL}(p,q) = D_{KL}(q,p)$, and that the self-similarity is non-zero, i.e. $D_{KL}(p,p) \neq 0$. Actually, $D_{KL}(p,p) = 2d$ with d being the dimensionality of the data vectors (20 MFCCs in our case).

V. RESULTS

The major evaluation measure to characterize hubness in the data is the k-occurrence O^k . It is the number of times a song occurs in the first k nearest neighbors of all the other songs in the database (see e.g. [3]). The mean 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 = 5because our music recommender always shows the five most similar songs. As has been done before [43], 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 koccurrence, 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). These statistics are given in Table I, with the number of hubs being 653 and the number of anti-hubs 5953, which means that more than a third of the data are never being recommended. The largest hub maxhub appears in the recommendation lists of 620 other songs. The 653 hub songs cover 40.11% of all the recommendation lists, although they themselves constitute only about 4% of the data base.

A. What makes a song a hub song?

We first try to verify the theoretical result (see Section II) that hub objects are close to the center of the full data set. To compute the center of the data set we use kmeans-clustering based on the weighted symmetrized Kullback-Leibler centroid [4], [44] with the number of clusters set to one.

In Figure 1 we plot the k-occurrence O^k versus the distance (KL divergence as in Equation 2) to the data center. As can be seen, the hub songs (blue dots) show the smallest distance and of course largest O^k . Normal songs (green dots) and anti-hub songs (red dots) do not come as close to the data center and show large variance of distances. This becomes more apparent when looking at the mean distances and their standard deviation in Table I. The average KL divergence of hub songs to the center is 51.63, that of normal songs 68.60 and that of anti-hubs already 76.22. From looking at Figure 1 and at the standard deviations, as well as at minimum and maximum KL divergences given in Table I, it is also clear



Fig. 1. O^k (y-axis) occurrences versus KL divergence (x-axis, log scaled) to the center of all data. Hub songs are depicted in blue, normal songs in green and anti-hub songs in red.

that there is a considerable overlap of the distances to the center for hubs, normal and anti-hub data. The Spearman rank correlation between the k-occurrences O^k and the KL divergences is -0.48, indicating a moderate negative relation with k-occurrences being larger for smaller KL divergences.

Next we try to answer the question, whether hub songs have a specific sound, i.e. whether they have a specific semantic meaning. In Figure 2 we plot the FFT power spectra for 10 seconds of the largest hub song, a normal and an anti-hub song. To allow better comparability of plots all three songs are from the 'Electronic' genre and contain vocals, just as the largest hub song. As can be seen from the plots, there are no distinctive features that would enable differentiation between the hub, normal or anti-hub song. Also listening to these examples², as well as many other randomly chosen songs, does not offer any clues as to what makes a song act as a hub song.

Looking at boxplots of the mean MFCC values for hubs, normal and anti-hub songs in Figure 3 shows that there are also no peculiarities in the MFCCs either. The mean values for all three types of songs are very comparable, the main difference being the larger variation for normal and anti-hub songs compared to hub songs. Comparison of MFCC covariances does also not offer any indication how to distinguish the songs.

We now argue in more detail, whether a song's property of being or not being a hub can, in the context of the method for computing similarity as described above, depend on meaningful, semantic content of that particular song. A waveform, which is perceptually equivalent to the original sound, and thus bears the same semantic content, can in principle be reconstructed from a sufficiently redundant spectrogram [35],

 $^{^2} The three songs can be listened to at this site: http://ofai.at/~arthur.flexer/hdm2018.html$



Fig. 2. FFT power spectrum (y-axis) for 10 seconds (x-axis) of the largest hub song, a normal and an anti-hub song. All songs are from the 'electronica' genre with vocals.



Fig. 3. Box plots for mean vectors of Gaussian models for hub, normal and anti-hub songs.

[50]. Synthesis of a meaningful approximation to the original sound can also still be obtained from a mel-spectrogram, i.e. a version of spectrogram coefficients in which averaging has been performed in frequency [46], [54]. Since the cosine transform is invertible, a similar reconstruction would also be possible from MFCCs, albeit with a certain loss of information since only a limited number (20 in our case) of MFCCs are retained. On the other hand, using a single Gaussian to model the spectral content of a song (see Section IV-B) essentially averages over all the MFCCs in time, reducing the observation to mean and variance statistics and discarding temporal structure [27]. The remaining average information does not allow to make any meaningful statements about the

original waveform that was used for computing the MFCCs. The same average can be obtained through data with very large or very small variation around the mean values, just to give a trivial example. As a consequence, very different audio signals can be close to the center of all data and hence very different audio signals can become hub songs.

B. What if we delete central or hub songs from the data base?

Next we explore what happens if we remove certain songs from the data base. We removed the 653 most central songs (i.e. those with minimal KL divergence to the center of the data), or we removed the 653 hub songs (i.e. every song with $O^k > 25$), or, as a control, 653 random songs. In Figure 4 we plotted the results with the k-occurrences before deletion on the x-axis and the k-occurrences after deletion in the now slightly smaller data base on the y-axis. Looking at the left plot giving the results for deletion of the most central songs, one can see that k-occurrences increase after deletion, i.e. most points in the plot are above the dashed diagonal axis. Since many of the deleted most central points are also hub songs, other songs now take over the role of these hub songs. After deletion of the most central points there are 650 hub songs with an O^k larger than 25, which is almost identical to the 653 hub songs before deletion. The largest hub maxhub now appears in the recommendation lists of 145 other songs, as opposed to 620 before deletion. The new 650 hub songs cover 34.85% of all the recommendation lists, opposed to 40.11%before deletion. The hub songs after deletion take over the role of the hub songs in the full data base to a large extent, but not fully so. There also is a high Pearson correlation of 0.97 between O^k before and after deletion, indicating that songs do not completely change their role with respect to being hubs, normal or anti-hub data.

Looking at the center plot giving the results for deletion of the 653 hub songs, one can again see that k-occurrences increase after deletion. After deletion of all hub songs there are 629 new hub songs with an O^k larger than 25, which is close to the 653 hub songs before deletion. The largest hub maxhub now appears in the recommendation lists of 88 other songs, as opposed to 620 before deletion. The new 629 hub songs cover 27.80% of all the recommendation lists, which is less than before deletion and less than with deletion of the most central songs, but still quite considerable. The Pearson correlation of O^k before and after deletion is again very high at 0.93.

Looking at the right plot giving the results for deletion of 653 random songs, one can see that k-occurrences basically remain identical with a perfect Pearson correlation of almost 1. The number of hub songs at 636 and the maxhub of size 615 remain hardly changed. The hub songs cover 40.20% of all recommendation lists which is almost identical to the full data base. It seems that because only random songs are being deleted and not hub songs or songs from the center, the role of the songs concerning their being hubs, anti-hubs or normal songs is hardly changed at all.



Fig. 4. k-occurrences before (x-axis) and after (y-axis) deletion of 653 most central songs (left plot), all 653 hub songs (middle plot), 653 random songs (right plot).

VI. DISCUSSION

In discussing our results we first want to summarize our major findings. In Section V-A we were able to corroborate the theoretical result that hub songs are indeed on average closer to the center of all data than other songs. This relationship is not completely clear, which might be due to the fact that the data distribution perhaps is multi-modal and we worked with one center of the data only. Additional experiments looking at the relationship of the distance of hub objects to multiple centers obtained via clustering might help in this respect.

By listening to hub, normal and anti-hub songs, by inspecting their time-frequency representation and their respective Gaussian models it became clear that hub songs do not carry a distinct semantic meaning. Since modeling the MFCC representation via single Gaussians essentially computes an average in time entailing a severe loss of information, very different audio waveforms can end up being close to the center of the data. This is further confirmed by our experiments deleting all hub songs or an equal number of central songs from the data base in Section V-B. In this case other songs take over the role of hub songs, with all their negative implications of dominating the recommendation lists.

In summary it can be said that hubness should be seen as a form of technical algorithmic bias in high dimensional machine learning, causing unfair treatment of music by the recommender system. This recommendation system is based on Mel Frequency Cepstrum Coefficients (MFCCs), a timefrequency representation of the spectral information of a signal, which is at the heart of many MIR systems (see e.g. [53] for an influential example using MFCCs for music classification). The frequency distributions of musical recordings are determined by the instruments and voices used as well as technical equalization choices made by the recording and mixing engineer. Although modeling the MFCCs computed from a song via single Gaussians entails a severe loss of information, at least average information concerning the frequency distributions still persists. It is the problem of measuring distances in high dimensional spaces that causes unfair treatment of certain songs, either by recommending them too often if they are close to the center of all data, or by never recommending them in case they are far from the center. This property of being close or far from the center of the data can seemingly not be connected to any semantic content of the underlying audio.

The resulting unfair treatment of certain songs will cause more or less exposure to interested audiences. In case this recommendation system were at the heart of a music streaming service distributing revenue in relation to listening time, the unfair treatment would result in monetary gains and losses for the respective artists who created the songs. These gains and losses would be due to the technical algorithmic bias of the system, but not to the semantic content or level of popularity of the songs. The fact whether a song is close to the center of the whole data base, given a certain representation and method of computing music similarity, is a major factor deciding whether a song will be recommended very often or never at all. Since this closeness to the center of all data bears no clear relation to semantic musical meaning, it is a failure of the recommendation system due to technical algorithmic bias.

One could even go a step further and try to take advantage of this technical algorithmic bias by trying to move the model of a song closer to the center of the data thereby turning it into a hub song gaining more recommendations. For music genre classification systems, it has been shown [49] that irrelevant audio filtering transformations of the music signal can be used to both deflate and inflate the performance, as measured with e.g. genre classification accuracy, to no better than chance level or perfect 100% respectively. The irrelevance is ascertained via listening tests, with the transformation being audible but not changing the clear impression of a certain musical genre. It is probably not straightforward but still conceivable that similar application of minor equalization filtering, almost in-audible to a listener, could shift a song's model closer to the center of the data.

On the other hand, methods to reduce hubness in machine learning systems have already been developed [16], [25],

[43], [51] and have been applied successfully [19] to the music recommendation system analyzed in this paper. Hubness reduction increased the reachability, i.e. the percentage of recommended songs, from less than two thirds of all songs to more than 90%. Additional analysis showed 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% before. Hubness reduction also increased the semantic correctness of music recommendations with the percentage of non-matching music genre labels between a music track and its nearest neighbors dropping from 56% to 38%. Therefore it could be said, that a fairness-aware data mining approach [24] to hubness already exists, which has been developed without prior understanding of hubness as a problem of technical algorithmic bias and fairness. This understanding is provided in this paper for the first time.

It took the effort of three research teams and a time period of seven years from the discovery of the hubness phenomenon [3] in 2004, to understanding it is a problem of learning in high-dimensional spaces [36], to developing a first hubness reduction method [42] in 2011. This should make it clear that only the researchers developing these methods are able to discover and understand a technical algorithmic bias existing in their methods. In a very recent report [38] from the French 'Commission for the Ethics of Research in Information Sciences and Technologies (CERNA)' on 'Research Ethics in Machine Learning', the ethical responsibility of machine learning researchers has been made very clear by stating that "The originator or the designer is responsible if the system is poorly designed, the user is responsible if he or she has misused the system". This responsibility is illustrated with a user being clumsy and hitting himself with a hammer on the finger versus the designer being responsible in case the head of the hammer flies off and knocks the user out. This paper tried to raise awareness that high-dimensional data mining produces potentially dangerous 'hammers' too and that it is our duty as data mining researchers to design our systems in a safe and fair way.

VII. CONCLUSION

The intention of this paper was to bring the ethical responsibility to produce fair and unbiased systems to the attention of the high-dimensional data mining community. This was done by presenting an example of technical algorithmic bias, where a music recommendation system, due to a problem of high-dimensional data mining, favors a small group of songs in its recommendations. These so-called hub songs dominate the recommendation lists not because they have a specific sound or semantic meaning, but because the algorithmic bias of the system favors songs close to the center of the data set, a requirement which these hub songs fulfill almost by accident. Deletion of songs close to the center of the data distribution promotes other songs to hub songs. It is our hope that this paper will trigger an in-depth discussion about the technical biases built into data mining algorithms operating in high dimensional spaces.

REFERENCES

- Angiulli F.: On the Behavior of Intrinsically High-Dimensional Spaces: Distances, Direct and Reverse Nearest Neighbors, and Hubness, *Journal of Machine Learning Research*, 18(170):1-60, 2018.
- [2] Aucouturier J.-J., Pachet F.: A scale-free distribution of false positives for a large class of audio similarity measures, *Pattern Recognition*, 41, 272-284, 2008.
- [3] Aucouturier J.-J., Pachet F.: Improving Timbre Similarity: How high is the sky?, *Journal of Negative Results in Speech and Audio Sciences*, 1(1), 2004.
- [4] Banerjee A., Merugu S., Dhillon I.S., Ghosh J.: Clustering with bregman divergences, *Journal of Machine Learning Research*, 6:1705-1749, 2005.
- [5] Bellman R.E.: Adaptive control processes: a guided tour, Princeton University Press, Princeton, 1961.
- [6] Beyer K., Goldstein J., Ramakrishnan R., Shaft U.: When is "nearest neighbor" meaningful? *International conference on database theory*, Springer, Berlin, Heidelberg, pp. 217-235, 1999.
- [7] Bolukbasi T., Chang K.W., Zou J.Y., Saligrama V., Kalai A.T.: Man is to computer programmer as woman is to homemaker? debiasing word embeddings, *Advances in Neural Information Processing Systems*, pp. 4349-4357, 2016.
- [8] Caliskan A., Bryson J.J., Narayanan A.: Semantics derived automatically from language corpora contain human-like biases, *Science*, Vol. 356, Issue 6334, pp. 183-186, 2017.
- [9] Camastra F., Staiano A.: Intrinsic dimension estimation: Advances and open problems, *Information Sciences*, 328, 26-41, 2016.
- [10] Celma O.: Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space, Springer Science & Business Media, 2010.
- [11] Dinu G., Lazaridou A., Baroni M.: Improving zero-shot learning by mitigating the hubness problem, in *Proc. of International Conference on Learning Representations*, workshop track, 2015.
- [12] Feldbauer R., Flexer A.: A comprehensive empirical comparison of hubness reduction in high-dimensional spaces, *Knowlege and Information Systems*, published online 18th of May, 2018. https: //doi.org/10.1007/s10115-018-1205-y
- [13] Flexer A.: Hubness-aware outlier detection for music genre recognition, in Proc. of the 19th International Conference on Digital Audio Effects, pp. 69-75, 2016.
- [14] Flexer A.: 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, 2016.
- [15] Flexer A.: Improving visualization of high-dimensional music similarity spaces, in *Proc. of the 16th Intern. Society for Music Information Retrieval Conference*, 2015.
- [16] Flexer A., Schnitzer D.: Choosing l^p norms in high-dimensional spaces based on hub analysis, *Neurocomputing*, Volume 169, pp. 281-287, 2015.
- [17] Flexer A., Schnitzer D., Gasser M., Pohle T.: Combining features reduces hubness in audio similarity, in *Proc. of the Eleventh International Society for Music Information Retrieval Conference*, 2010.
- [18] Flexer A., Schnitzer D., Schlüter J.: A MIREX meta-analysis of hubness in audio music similarity, in *Proc. of the 13th International Society for Music Information Retrieval Conference*, 2012.
- [19] Flexer A., Stevens J.: Mutual proximity graphs for improved reachability in music recommendation, *Journal of New Music Research*, 47(1), pp. 17-28, 2018.
- [20] Francois D., Wertz V., Verleysen M.: The concentration of fractional distances, *IEEE Trans. on Knowledge and Data Engineering*, Vol. 19, No. 7, pp. 873-886, 2007.
- [21] Friedman B., Nissenbaum H.: Bias in Computer Systems, ACM Trans. on Information Systems, 14 (3): 330-347, 1996.
- [22] Gasser M., Flexer A.: FM4 Soundpark: Audio-based Music Recommendation in Everyday Use, Proc. of the 6th Sound and Music Computing Conf., 2009.
- [23] Goodman B., Flaxman S.: European Union Regulations on Algorithmic Decision-Making and a 'Right to Explanation', AI Magazine, Vol. 38, No. 3, pp. 50-57, 2017.

- [24] Hajian S., Domingo-Ferrer J.: A methodology for direct and indirect discrimination prevention in data mining, *IEEE transactions* on knowledge and data engineering, 25(7), 1445-1459, 2013.
- [25] Hara K., Suzuki I., Shimbo M., Kobayashi K., Fukumizu K., Radovanović M.: Localized centering: Reducing hubness in largesample data hubness in high-dimensional data, in *Proc. of the 29th* AAAI Conference on Artificial Intelligence, pp. 2645-2651, 2015.
- [26] Holzapfel A., Sturm B., Coeckelbergh M.: Ethical Dimensions of Music Information Retrieval Technology, Transactions of the International Society for Music Information Retrieval, in press, 2018.
- [27] Humphrey E.J., Bello J P., LeCun Y.: Feature learning and deep architectures: New directions for music informatics, *Journal of Intelligent Information Systems*, 41(3), 461-481, 2013.
- [28] Kabán, A.: Non-parametric detection of meaningless distances in high dimensional data, *Statistics and Computing*, 22(2), 375-385, 2012.
- [29] Karydis I., Radovanović M., Nanopoulos A., Ivanović M.: Looking through the "glass ceiling": A conceptual framework for the problems of spectral similarity, in *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pp. 267-272, 2010.
- [30] Logan B.: Music Recommendation from Song Sets, Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04), Barcelona, Spain, October 10-14, 2004.
- [31] Mandel M.I., Ellis D.P.W.: Song-Level Features and Support Vector Machines for Music Classification, *Proceedings of the 6th International Conference on Music Information Retrieval*, London, UK, September 11-15, 2005.
- [32] Müller M.: Fundamentals of music processing: Audio, analysis, algorithms, applications, Springer, 2015.
- [33] Pedreshi D., Ruggieri S., Turini F.: Discrimination-aware data mining, Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 560-568, 2008.
- [34] Penny W.D.: Kullback-Liebler Divergences of Normal, Gamma, Dirichlet and Wishart Densities, Wellcome Department of Cognitive Neurology, 2001.
- [35] Průša Z., Balazs P., Søndergaard P.L.: A Noniterative Method for Reconstruction of Phase From STFT Magnitude, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 25, Issue 5, pp. 1154-1164, 2017.
- [36] Radovanović M., Nanopoulos A., Ivanović M.: Hubs in space: Popular nearest neighbors in high-dimensional data, *Journal of Machine Learning Research*, 11:2487-2531, 2010.
- [37] Radovanović M., Nanopoulos A., Ivanović M.: Reverse nearest neighbors in unsupervised distance-based outlier detection, *IEEE Trans. on Knowledge and Data Engineering*, 27(5), 1369-1382, 2015.
- [38] Research Ethics Board of Allistene, the Digital Sciences and Technologies Alliance: Research Ethics in Machine Learning, *CERNA Report*, February 2018. http://cerna-ethics-allistene.org/ digitalAssets/54/54730_cerna_2017_machine_learning.pdf
- [39] Schedl M., Gómez E., Urbano J.: Music Information Retrieval: Recent Developments and Applications, *Foundations and Trends in Information Retrieval*, Volume 8, Number 2-3, pp. 127-261, 2014.
- [40] Schnitzer D.: Mirage High-Performance Music Similarity Computation and Automatic Playlist Generation, Vienna University of Technology, Austria, Master Thesis, 2007.
- [41] Schnitzer D., Flexer A.: The Unbalancing Effect of Hubs on Kmedoids Clustering in High-Dimensional Spaces, in Proc. of the International Joint Conference on Neural Networks, 2015.
- [42] Schnitzer D., Flexer A., Schedl M., Widmer G.: Using Mutual Proximity to Improve Content-Based Audio Similarity, in *Proc.* of the 12th International Society for Music Information Retrieval Conference, 2011.
- [43] Schnitzer D., Flexer A., Schedl M., Widmer G.: Local and Global Scaling Reduce Hubs in Space, *Journal of Machine Learning Research*, 13:2871-2902, 2012.
- [44] Schnitzer D., Flexer A., Widmer G., Gasser M.: Islands of Gaussians: The Self Organizing Map and Gaussian Music Similarity Features, in *Proceedings of the Eleventh International Society for Music Information Retrieval Conference*, 2010.

- [45] Serra X.: A Multicultural approach in music information research, in Proc. of the 12th International Society for Music Information Retrieval Conference, pp. 151-156, 2011.
- [46] Serra X., Smith J.: Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition, *Computer Music Journal*, 14(4), 12-24, 1990.
- [47] Serra X., Magas M., Benetos E., Chudy M., Dixon S., Flexer A., Gomez E., Gouyon F., Herrera P., Jorda S., Paytuvi O., Peeters G., Schlüter J., Vinet H., Widmer G.: *Roadmap for Music Information ReSearch*, Peeters G. (ed.), 2013, Creative Commons BY-NC-ND 3.0 license, ISBN: 978-2-9540351-1-6.
- [48] Shigeto Y., Suzuki I., Hara K., Shimbo M., Matsumoto Y.: Ridge regression, hubness, and zero-shot learning, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 135-151, 2015.
- [49] Sturm B.L.: A simple method to determine if a music information retrieval system is a "horse", *IEEE Trans. on Multimedia*, 16(6), pp. 1636-1644, 2014.
- [50] Sturmel N., L Daudet L.: Signal reconstruction from STFT magnitude: A state of the art, in *Proc. International Conference on Digital Audio Effects*, pp. 375-386, 2011.
- [51] Tomašev N., Mladenić D.: Hubness-aware shared neighbor distances for high-dimensional k-nearest neighbor classification, *Hybrid Artificial Intelligent Systems*, pp. 116-127, Springer, 2012.
- [52] Tomašev N., Radovanović M., Mladenić D., Ivanović M.: The Role of Hubness in Clustering High-dimensional Data, *IEEE Trans. on Knowledge and Data Engineering*, Volume 26, Issue 3, 2013.
- [53] Tzanetakis G., Cook P.: Musical genre classification of audio signals, *IEEE Transactions on speech and audio processing*, 10(5), pp. 293-302, 2002.
- [54] Waldspurger I.: Phase retrieval for wavelet transforms, *IEEE Transactions on Information Theory*, 63(5), pp. 2993-3009, 2017.