Improving Recommender Systems by Reducing Hubness

Donthu Vamsi Krishna (15111016)
vamsi@iitk.ac.in
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Dhekane Eeshan Gunesh (13248)
eeshangd@iitk.ac.in
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Project Guide- Prof. Piyush Rai

Date- April 26, 2016

Project Group- 1

1 Abstract

The problem of Hubs occur while dealing with high dimensional data. Due to which hubs tend to be in the nearest neighbors of most of the points which reduces the performance of nearest neighbor methods. Therefore nearest neighbor relations become asymmetric and hence the anti hubs will not be in the nearest neighbors of most of other points. In this project, we analyze the problem of hubs and remove the hubness in the data by making the nearest neighbor relations close to being symmetric.

Keywords– **Hubs, nearest neighbor methods, asymmetric, Anti-hubs**

2 Introduction to Recommender Systems

Recommender systems or recommendation systems (sometimes replacing “system” with a synonym such as platform or engine) are a subclass of information filtering system that seek to predict the ‘rating’ or ‘preference’ that a user would give to an item[5].

In today’s web, the recommender systems play an important role and hence they are becoming more and more advanced by learning a lots of parameters and hence the dimension of data that they deals with increases. Since the recommender systems rely on nearest neighbor methods and deals with high dimensional data, their performance degrades by the occurrence of hubs. In this project we would like to reduce the amount of hubness in the given data sets. For this purpose, we implemented 4 approaches that reduces hubness namely Centering, Weighted Centering, Local Scaling and Global scaling[2][3]. In order to compare these methods, we measured the amount of hubness by measuring the skewness measure[4] of the data and also by calculating Goodman-Kruskal Index [6].

3 Introduction to Hubness Phenomenon

The ***phenomenon of hubs*** is a relatively recent discovery and was formally stated in the year of 2010[4]. It can be described as follows– If the feature space is high-dimensional then some objects of the data set become the so-called hubs. These hubs are those points which appear in the k nearest neighbor lists of majority of other objects.

This hub phenomenon is observed to significantly deteriorate the performance of the algorithms for similarity or nearest neighbor search[2]. Hence, they also degrade the performance of recommender systems that use similarity based nearest neighbor search. This is because these objects are obtained unnecessarily and frequently as the neighbors of test items and hence, are recommended more frequently. Analogous to hubs, another phenomenon of ***anti-hubs*** is observed. These are those points which do not appear in any of the nearest neighbor lists and hence, do not get recommended at all. Experiments show that the phenomenon of hubness is very prominent in real life data sets. As an example, in work done by Gasser and Flexer[7] on Audio-based Music Recommendation it was observed that due to presence of hubs, only 72.6% songs were reachable and only about a third of the songs are likely to be recommended. This shows that about 25% of the data points are never recommended (reached) or have become anti hubs.

The phenomenon of hubs can be related to the ***asymmetries in the nearest neighbor relations***. Consider a point h which is a hub in the given feature space. Now, consider the set S of all other points that contain h as one of the k nearest neighbors. The size of S is very large since h is a hub. Now, if we consider any element e in the k nearest neighbor list of h, then it is very likely that it will not be a nearest neighbor of h. This is because of the large size of S. Hence, the ***nearest neighbor relationship tends to become very asymmetric***.
This connection is extremely useful in devising a very powerful and general remedy for removing hubness. It includes ***changing the distance metric so as to make nearest neighbor relationships more symmetric***.

4 Related Work

Previously done work in this field includes connecting the hub phenomenon with previous works, analyzing the effects of hubness in various real life data sets and coming up with methods to reduce hubness. The last will be considered in the next sections.

4.1 Study of effects of hubs in various domains

In their works, Aucoutier and Pachet[8] studied the phenomenon of hubs in music information retrieval (MIR). They observed certain songs that were similar to a large number of other songs with respect to the used audio-similarity functions. Thus, these songs appeared in the nearest neighbor lists of many other songs and prevented the anti-hub songs from entering the nearest neighbor lists. Work by Oscar Celma[9] showed how hub phenomenon "infected" the performance of collaborative filtering based recommender systems. Doddington et al.[10] studied the effects of hub problems in the problem of image retrieval and found their adverse effects. Some studies[4] also showed the effect of hubs in text retrieval.

4.2 Connections with previous works

Some initial works[11] showed that hubs can be viewed as False Positives when are considered in context of classification problem. It is thought that there are connections between hub problems and the high dimensionality of the feature space. Important work by Radovanović et al.[4] linked the hub property to the phenomenon of concentration of distances. It is showed that concentration of distances in high dimensional feature space takes some points close to the data mean and at the same time, closer on average to all the other data points.

In addition, it can be seen that hubs can be linked very naturally to the technique of Shared Nearest Neighbors (SNN), which consider the size of intersection nearest neighbor lists of two points as a measure of the similarity between the two objects.

5 Measuring Hubenss

5.1 Skewness Measure[4]

We have stated the similarity between hubness and asymmetry of nearest neighbor relation. This can be used to define a measure of hubness in the given data space with respect to a given distance metric [4]. Suppose that S is the set of all feature vector points. We first fix some k, which is the size of nearest neighbor list. We randomly select $n+1$ number of points– $x, x_1, \ldots, x_n$ from the data. Now, we define an indicator function $I_{k,i}(x)$ as follows–

$$I_{k,i}(x) = \begin{cases} 1 & \text{if } x \text{ appears in the } k \text{ nearest neighbors list of } x_i \\ 0 & \text{otherwise} \end{cases}$$

Now, we define a measure $N_k(x)$ for how often does x appear in the k nearest neighbors of other elements as follows–

$$N_k(x) = \sum_{i=1}^{n} I_{k,i}(x) \tag{1}$$

As mentioned, it gives an idea of how often does the element x occur in the k nearest neighbors of other elements. Now, we view N_k as a function– $N_k : S \to \mathbb{R}$. It is assigning a real number to each of the feature vectors. If the hubness is very high, then the nearest neighbor relation will be highly asymmetric and hence, the distribution of N_k will be highly skewed. Hence, define skewness of N_k distribution, denoted by S_{N_k} as one measure of hubness–

$$S_{N_k} = \mathbb{E}\left[\frac{(N_k - \mu_{N_k})}{\sigma_{N_k}}\right]^3 \tag{2}$$

where μ_{N_k} and σ_{N_k} are the mean and standard deviation of distribution of N_k. $\mathbb{E}$ denotes the probabilistic expectation.

5.2 Goodman-Kruskal Index[2]

Goodman-Kruskal Index is a measure of the quality of clusters formed in a feature space provided a distance metric d is given. It works by counting the number of concordant and discordant tuples in a distance matrix. Consider feature vector space given as $S = \{x_1, \ldots, x_n\}$. Consider any tuple $T = (i, j, k, l)$. T is called as ***concordant*** iff x_i, x_j belong to the same class, x_k, x_l belong to different classes and $d(i,j) < d(k,l)$. This shows that a concordant tuple is an instance when the intra-class distance is smaller in comparison with the inter-class distance. T is called as ***discordant*** iff x_i, x_j are in the same class, x_k, x_l are in different classes and $d(i,j) > d(k,l)$. A tuple that is neither concordant, nor discordant is not counted. Let N_C, N_D be the total number of concordant and discordant tuples respectively. Then, Goodman-Kruskal Index, denoted by I_GK is defined as–

$$I_{GK} = \frac{N_C - N_D}{N_C + N_D} \tag{3}$$

It can be clearly seen that $I_{GK} \in [-1, 1]$. Also, a larger value of I_{GK} would imply that more concordant tuples exist than discordant and hence, better will be the clustering quality.

6 Methods of Reducing Hubness

We consider mainly two types of methods for reducing hubness as follows–

6.1 Centering and Weighted Centering[3]

Consider that we are given with a feature space in high dimensions. The idea comes from the heuristic of Concentration of distances and Radovanović's observation[4] that it leads to some points being more close to the mean of the data and at the same time, closer on average to all the data points. Thus, in ***centering*** we shift the origin of the feature space to the center of the data. Thus, each of the feature vectors will be transformed accordingly. It is proved[3] that doing so reduces the probability of a point remaining to be a hub. The idea is that before centering the hub was having almost the same distances with all the other points. But after the transformation, these distances will be changed significantly as the hub is now the origin. This method works when ***cosine distance metric*** is used.

Suppose the feature space is S. Then the origin is first moved to the center or mean of the data and then the coordinates of each point are changed accordingly–

$$x_{mean} = \sum_{x \in S} x$$
$$x_{centered} = x - x_{mean} \tag{4}$$

Note that there can be multiple hubs in data which are centered around multiple hubs (concentrated, multi-modal data). Thus, centering can fail in these cases.

But to overcome this issue to some extent, centering can be done in steps, instead of a single step. The method of centering can be modified to be done in steps to overcome the shortcomings of the previous method.

In ***weighted centering***, the origin is moved progressively towards the hub objects in the data set which are identified using a function measuring the similarity of a point with every other point.

$$x_{weighted\ mean} = x_{wm} = \sum_{y \in S} w_y y$$
$$x_{weighted\ centered} = x_{wc} = x - x_{wm} \tag{5}$$

To identify hubs, the similarity of every point will all the other points needs to be calculated. This is measured using the term $J(x)$ as follows–

$$J(x) = \sum_{y \in S} \langle x, y \rangle$$
$$J(x) = \sum_{y \in S} \langle x, y \rangle = |S| \langle x, x_{mean} \rangle \tag{6}$$

Note that this shows that we do not need to compute all the $|S|^2$ inner products, but just $|S|$ inner products to compute $J-$values for all the feature points.

The weights need to be fixed so that the center is moved more aggressively towards the hub objects. And hub objects are those that are similar to large number of points. Hence, these will be the points having a large value of J. Hence, we define the weight w_y corresponding to the feature vector y as follows–

$$w_y = \frac{J(y)^\gamma}{\sum_{x \in S} J(x)^\gamma} \tag{7}$$

where $\gamma > 0$ is a parameter that dictates how much emphasis needs to be given to the effect of $J(y)$. Note that the weights are normalized, that is to say–

$$\sum_{x \in S} w_x = 1 \tag{8}$$

Note that $y = 0$ gives the special case of regular non-weighted centering.

6.2 Scaling Approaches

As seen earlier, nearest neighbor asymmetry and hubness are related and a major approach to reduce hubness is to modify the distance metric so that the nearest neighbor relations become symmetric.

6.2.1 Local Scaling

In local scaling [12], we consider that some distance metric d and a feature space S is given. The distance between two points $d_{x,y}$ is then transformed to the entity $LS(d_{x,y})$ as follows–

$$LS(d_{x,y}) = \exp\left(-\frac{d_{x,y}^2}{\sigma_x \sigma_y}\right) \tag{9}$$

where, σ_z is the standard deviation in the distance of point z from its k nearest neighbors.
The heuristic behind this transform is that $LS(d_{x,y})$ will be large when $d_{x,y}$ is much lesser than both σ_x and σ_y. Now the standard deviations represent the extent to which the nearest neighbors of the point are spread in the space and hence, the condition translates to saying that the distance between x, y has to be smaller than the spread of nearest neighbors of x and y. This would imply that if x is in nearest neighbor list of y then y is also in the nearest neighbor list of x. Thus, high value of LS would mean that the nearest neighbor relation is very symmetric. Local scaling only looks at the k nearest neighbors of all the points. And hence, the approach is affected by local properties of feature space.

6.2.2 Mutual Proximity/ Global Scaling[2]

Global Scaling tries to transform the distance between two points x, y to an entity which is the probability that y is the nearest neighbor of x given the distribution of distances of all points to x in the feature space. Then combining these probabilities yields an entity called ***mutual proximity***, which is a measure of probability that x, y are mutual nearest neighbors. Suppose that the distances of a point x_i from all other feature points x_j are denoted by $d_{i,j}$ and that their distribution is denoted by $P(X)$. Then, probability that a randomly selected point w from the feature space will be closer to x than another particular point y is given by $P(X > d_{x,y})$. Now, we define the mutual proximity, denoted by $MP(d_{x,y})$ of x, y as follows–

$$MP(d_{x,y}) = Pr\left(X > d_{x,y} \text{ and } Y > d_{y,x}\right) \tag{10}$$

For simplicity in calculations we assumed that the probability distributions are independent. Actually they are not, but experimental results show that this assumption works fine. Thus,

$$MP(d_{x,y}) = Pr(X > d_{x,y}) \cdot Pr(Y > d_{y,x}) \tag{11}$$

And we measure each of the probabilities by using simple approximation of favorable cases against total cases.

$$Pr(Z > d_{z,w}) = \frac{\text{Number of Instances where } Z > d_{z,w}}{|S|}$$

These are the hubness reducing techniques used in this project.

7 Data Sets

We have used the MovieLens data set [13] available at the website- "*grouplens.org/datasets/movielens/*". We have used three data sets- 100k, 1m and 10m. In these data sets the number of ratings, approximately equal to 100000, 1000000 and 10000000 respectively, are given by certain number of users and for certain number of movies. This is the ***real life data set*** used in the experiment. We have also used IRIS flower dataset which contains 150 feature vectors and corresponding type of the flower [14]. We have multiplied this number of feature vectors to 15000 times by adding randomly generated small real numbers to each feature vector to get many more. In this process, the type of the flower (the class) is maintained. This is the ***synthetic data set*** used by us.

8 Methodology and Experiments

We describe the methodology used in our project in this section.

- The MovieLens data set was first converted into feature vectors by concatenation of user and item profile. This was done for 100k, 1m and 10m data sets. The IRIS data set was converted into feature vectors.
- Programs were written to compute the hubness measures- i. Skewness Index and ii. Goodman-Kruskal Index.
- Programs were written to reduce the hubness in the feature space. The methods include- i. Centering, ii. Weighted Centering, iii. Local Scaling and iv. Mutual Proximity/ Global Scaling.
- Distance Matrix was evaluated for IRIS, MovieLens 100k data sets. The 1m and 10m data sets were not used this step onwards because of distance matrices of the corresponding sizes resulted in memory errors.
- Centered Feature vectors were evaluated for methods of centering and weighted centering.
- Local and Global Scaling Algorithm was run for IRIS and MovieLens 100k to generate transformed distance matrices.
- Each of the measures of hubness was first applied on the original data and then on the transformed data. The weighted centering was carried out on the values of 0.2, 0.4, 0.6, 0.8, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0 for the parameter γ appearing in the weighted centering algorithm.
- The results are tabularized and analyzed for patterns.

  Note that for all the readings, five trial readings were taken and then their average was recorded as the answer. Only in the cases of weighted centering on MovieLens 100k data set, three trials were conducted. This was because of considerably large time was taken by each trial.

9 Results

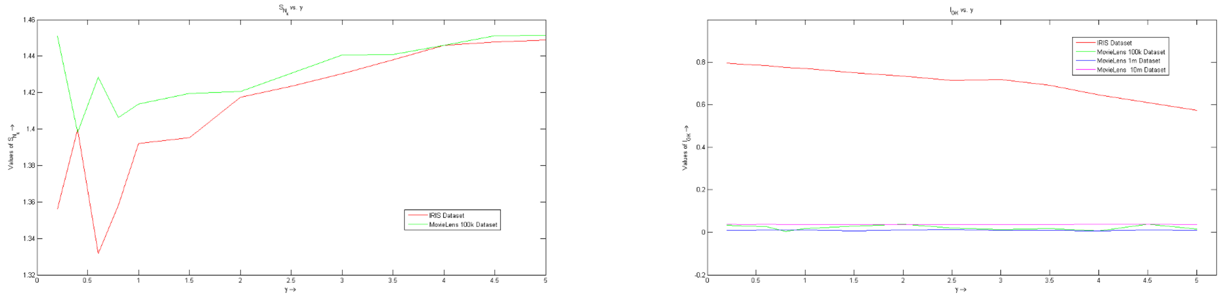
For IRIS data set, the results are displayed in the table below–

Method	γ	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Result
Original	None	1.437406665	1.391048009	1.41879077	1.45676048	1.53896259	1.448593703
Original	None	1.337101336	1.495413682	1.449115157	1.431763698	1.346745443	1.412027863
Centering	None	1.345110887	1.411364679	1.34283454	1.337079236	1.36573438	1.360424744
Weighted	$\gamma = 0.2$	1.359064116	1.381709147	1.335057124	1.403428631	1.302338025	1.356319409
Weighted	$\gamma = 0.4$	1.348359846	1.362869626	1.410255573	1.456306059	1.421590021	1.399876225
Weighted	$\gamma = 0.6$	1.34459165	1.377464253	1.347831499	1.281529843	1.306913212	1.331666092
Weighted	$\gamma = 0.8$	1.37574702	1.343456455	1.360256443	1.343873794	1.368563671	1.358379476
Weighted	$\gamma = 1$	1.385181094	1.414711666	1.34401905	1.358049429	1.458601226	1.392112493
Weighted	$\gamma = 1.5$	1.397795401	1.442237403	1.387755102	1.367779784	1.380948564	1.395303251
Weighted	$\gamma = 2$	1.469106967	1.427350598	1.392417647	1.396512184	1.401102839	1.417298047
Weighted	$\gamma = 2.5$	1.380446571	1.379171033	1.397661346	1.465023402	1.495134833	1.423487437
Weighted	$\gamma = 3$	1.308352161	1.497251468	1.509386569	1.459770305	1.376558095	1.43026372
Weighted	$\gamma = 3.5$	1.406395735	1.424989897	1.447589002	1.451734446	1.458683656	1.437878547
Weighted	$\gamma = 4$	1.447983221	1.458318489	1.447863563	1.437764196	1.43810238	1.44600637
Weighted	$\gamma = 4.5$	1.362941964	1.505308549	1.390533324	1.562364331	1.41818262	1.447866158
Weighted	$\gamma = 5$	1.487915358	1.432332658	1.43881357	1.476796239	1.408030461	1.448777657
Local	None	1.233118738	1.233661193	1.234243537	1.234314626	1.232473443	1.233562308
Global	None	0.610595567	0.611382771	0.60744476	0.615145389	0.610118881	0.610937473

The results for MovieLens 100k are shown in the table below–

Method	γ	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Result
Original	None	1.47112373	1.452887176	1.461017441	1.417197044	1.466140576	1.453673193
Centering	None	1.401222569	1.444588522	1.420946423	-	-	1.422252505
Weighted	$\gamma = 0.2$	1.455877448	1.449170242	1.447919235	-	-	1.450988975
Weighted	$\gamma = 0.4$	1.404257943	1.426974703	1.363664067	-	-	1.398298904
Weighted	$\gamma = 0.6$	1.425002081	1.410408674	1.449902316	-	-	1.42843769
Weighted	$\gamma = 0.8$	1.399042149	1.421984777	1.398261207	-	-	1.406429377
Weighted	$\gamma = 1$	1.406582156	1.424331227	1.410082541	-	-	1.413665308
Weighted	$\gamma = 1.5$	1.427205364	1.409761044	1.421272103	-	-	1.419412837
Weighted	$\gamma = 2$	1.420623746	1.398494821	1.442728975	-	-	1.420615847
Weighted	$\gamma = 2.5$	1.435613995	1.437737429	1.418141774	-	-	1.430497733
Weighted	$\gamma = 3$	1.458956138	1.412363388	1.450605048	-	-	1.440641525
Weighted	$\gamma = 3.5$	1.422672117	1.401399803	1.498970397	-	-	1.441014105
Weighted	$\gamma = 4$	1.446606982	1.481339274	1.409971126	-	-	1.44597246
Weighted	$\gamma = 4.5$	1.460720676	1.444191336	1.448447057	-	-	1.45111969
Weighted	$\gamma = 5$	1.439256974	1.451268537	1.463375627	-	-	1.451300379
Local	None	1.153964617	1.153964275	1.153965255	1.15396179	1.153972509	1.153965689
Global	None	0.693127156	0.693874422	0.692035019	0.698998525	0.685079565	0.692622937

The graphs in Fig. 1(a), Fig. 1(b) below shows the variation of skewness index S_{N_k} and Goodman-Kruskal Index I_{GK} as a function of parameter γ for all the data sets.



(a) (b)

Figure 1: (a) S_{N_k} vs. γ. (b) I_{GK} vs. γ.

10 Conclusions

1. In the case of IRIS Dataset, we see that ***centering reduces the hubness only to some extent***. Theoretically, centering is supposed to reduce hubness of data with single hub. But this is not the case for IRIS dataset, in which there are 3 clusters. And hence, the decrease in hubness is only to some extent.

2. For IRIS dataset, ***weighted centering performs a little better than centering in reducing hubness***. This is as expected. Weighted centering works by moving the center progressively towards the data hubs according to their strength to become hubs. Thus, it is more robust to the possibility of multiple hubs than the regular centering. This is observed in the results.

3. For IRIS dataset, ***local scaling performs better than centering and weighted centering in reducing hubness***. This shows the better performance of local scaling technique, as expected.

4. For IRIS dataset, ***global scaling performs the best among all the methods, by reducing the hubness by large amounts (approx. 45 percent)***. It is in accordance with what is expected. Local scaling just looks at the local neighborhood of each of the points and adjusts the distances accordingly. Global scaling/ mutual proximity technique considers all the data points simultaneously and tries to find their effect on two points being nearest neighbors or not. And hence, this technique includes the effects of locality of data points. Hence, it is expected that global scaling performs better, which is the case.

5. Hence, we see that for the synthetic dataset IRIS, the results are as expected.

6. In the case of MovieLens 100k (ML) Dataset, we see that ***centering reduces the hubness only to some extent***. This is expected since the ML dataset has multiple hubs, and the reason is similar to one given for IRIS dataset.

7. For ML dataset, ***weighted centering performs a little better than centering in reducing hubness***. This is as expected. This is as expected and the reason is mentioned above.

8. For ML dataset, ***local scaling performs better than centering and weighted centering in reducing hubness***. This is in accordance with what is expected.

9. For ML dataset, ***global scaling performs the best among all the methods, by reducing the hubness by large amounts (approx. 45 percent)***. The explanation to this can be found above, in the result for IRIS dataset.

10. Hence, we see that for the real-life dataset ML also, the results are as expected.

11. For IRIS dataset, Goodman-Kruskal (GK) index of centered and weighted centered dataset is lesser than that of the original dataset. Smaller GK values shows that the cluster quality of the data is lesser. This shows that due to centering and weighted centering the clustering quality is decreased.

12. Local scaling further decreases the clustering quality and global scaling takes it to a big negative value, which indicates that the data is not clustered.

13. For ML dataset, centering and weighted centering slightly improve the cluster quality.

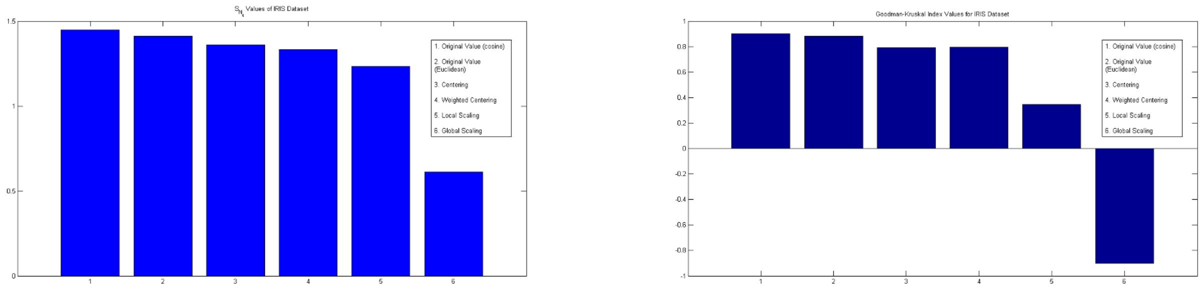
14. For ML dataset, global scaling has no significant effect on clustering quality.

15. Local scaling imporves clustering quality by larger amounts.

16. Hence, ***the effect of various techniques for reduction in hubness on the clustering quality is inconclusive and depends strongly on the particular datasets.***

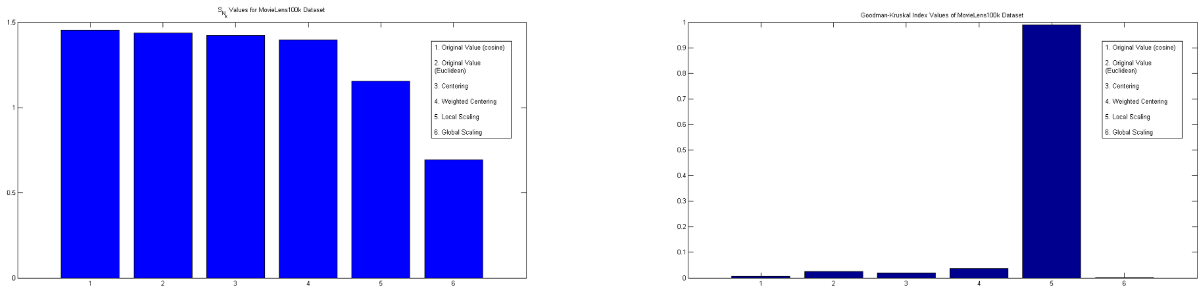
17. Hence, ***for both real and synthetic data, the hubness is reduces as expected*** and hence, the conducted experiments were successful.

We consider the graphical results in Fig. 2. It shows the S_{N_k} value variation and I_{GK} value variation for the IRIS dataset. Fig. 3 does the same for ML dataset.



(a) (b)

Figure 2: For IRIS dataset, (a)S_{N_K} index variation . (b) I_{GK} index variation.



(a) (b)

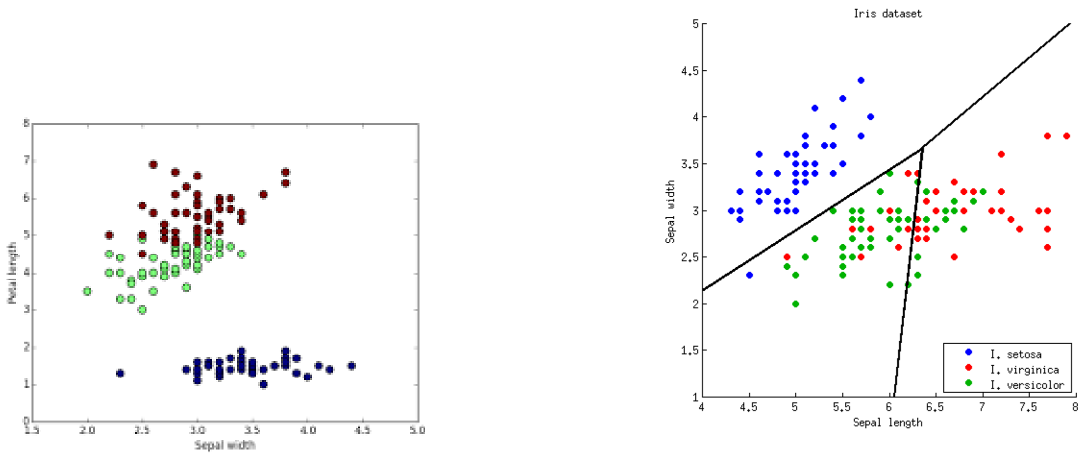
Figure 3: For ML dataset, (a)S_{N_K} index variation . (b) I_{GK} index variation.

10.1 Explanation for Goodman-Kruskal Index Results

We try to explain why Goodman-Kruskal index gives the observed results on the IRIS dataset. The dataset can be visualized in the two dimensions as given in Fig. 4. It has three classes of flowers corresponding to the 3 clusters and they are placed in space as shown. All the classes have the same "diameter" size and two of them are very close to one another while the third is far away. This drives the Goodman-Kruskal index to be driven in another way and is explaned below.

Goodman-Kruskal index works on the pairs of Concordant and Discordant points and hence, the points from the two different close clusters which lie on the boundary have lesser distance than two average points from any cluster. This is a possible reason because of which the I_{GK} value turn out different than expected.

This also shows the dependence of I_{GK} on the particular dataset.



(a) (b)

Figure 4: For IRIS dataset, (a) Visualization 1. (b) Visualization 2 with Flower Classes.

11 Future Plans

Provided we get more time to work on this project we would like to implement one idea which we have thought of and discussed with our mentor. We were unable to implement the idea due to time constraint. But we have run a very simplistic code of the same on MATLAB. The idea includes to consider measure(s) of similarity and dissimilarity between two objects and defining a similarity force and a dissimilarity force between every pair of objects. The former would try to bring the objects closer and the later would try to repel them away. In this way, from an initial state of distances (the initial distance matrix), we can get a new globally scaled distance matrix.

References

[1] Arthur Flexer, "The impact of hubness on music recommendation," *Machine Learning for Music Discovery Workshop at the 32nd International Conference on Machine Learning*, Lille, France, 2015.

[2] Dominik Schnitzer, Arthur Flexer, Markus Schedl, Gerhard Widmer, “Local and Global Scaling Reduce Hubs in Space,” *The Journal of Machine Learning Research*, Vol. 13, Issue 1, January 2012, Pages 2871-2902.

[3] Ikumi Suzuki, Kazuo Hara, Masashi Shimbo, Marko Saerens, Kenji Fukumizu, “Centering Similarity Measures to Reduce Hubs,” *Conference on Empirical Methods in Natural Language Processing*, Main Conference, 2013.

[4] Miloš Radovanović, Alexandro Nanopoulos, Mirjana Ivanović, “Hubs in Space- Popular Nearest Neighbors in High-Dimensional Data,” *The Journal of Machine Learning Research*, Vol. 11, 3/1/2010, Pages 2487-2531.

[5] Wikipedia contributors, “Recommender system,” Wikipedia, The Free Encyclopedia, *https://en.wikipedia.org/w/index.php?title=Recommendersystem&oldid=714461920* (accessed April 21, 2016).

[6] Simon Günter, Horst Bunke, “Validation indices for graph clustering,” *Pattern Recognition Letters*, 24(8):1107 – 1113, 2003, ISSN 0167-8655, DOI:10.1016/S0167-8655(02)00257-X.

[7] Martin Gasser, Arthur Flexer, “FM4 SoundPark: audio-based music recommendation in everyday use,” *In Proceedings of the 6th Sound and Music Computing Conference*, pp. 23–25, 2009.

[8] Jean-Julien Aucouturier , François Pachet, “Improving timbre similarity: How high is the sky?,” *Journal of Negative Results in Speech and Audio Sciences*, 1(1), 2004.

[9] Óscar Celma, “Music Recommendation and Discovery in the Long Tail,” PhD Thesis, Universitat Pompeu Fabra, Barcelona, 2008.

[10] George Doddington, Walter Liggett, Alvin Martin, Mark Przybocki, Douglas Reynolds, “Sheep, goats, lambs and wolves a statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation,” *In Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP-98)*, 1998.

[11] Arthur Flexer, Dominik Schnitzer, Martin Gasser, Tim Pohle, “Combining features reduces hubness in audio similarity,” *In Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, 2010.

[12] Lihi Zelnik-Manor, Pietro Perona, “Self-tuning spectral clustering,” *In Advances in Neural Information Processing Systems*, Vol. 17, pages 1601–1608, MIT Press, Cambridge, MA, 2005.

[13] F. Maxwell Harper, Joseph A. Konstan, “The MovieLens Datasets: History and Context,” *ACM Transactions on Interactive Intelligent Systems (TiiS)*, Article 19 (December 2015), 19 pages. DOI=“*http://dx.doi.org/10.1145/2827872*”.

[14] M. Lichman, UCI Machine Learning Repository, “*http://archive.ics.uci.edu/ml*”. Irvine, CA: University of California, School of Information and Computer Science.