Fall 2017

Final Project Report: Hubness and Some Questions

Yimeng Li and Vega Bharadwaj *CS 584 Data Mining*

1 Introduction

The number of k-occurences of point x from dataset D is called its hubness score. High hubness score indicates that this point is close to plenty of other points in the dataset. In a regular clustering algorithm, e.g. K-Means, a cluster medoid is also the point closest to all the points in the cluster. Since both of them, a cluster medoid and a high hubness point, share the same attribute, we believe it's highly possible that high hubness points are more likely to become cluster medoids comparing to points with a lower hubness score. In this project, we unveil the relationship between high hubness points and cluster medoids on both synthetic and real-world datasets.
Besides, low hubness scores also indicates that a point is far away from the majority of its peers, which is big sign of being an outlier or a noise. We happened to learn an algorithm called DBSCAN during the semester, which is quite effective in finding data outliers using density information. The second problem we are trying to solve is to explore the relation between low hubness points and DBSCAN-detected noise points on both synthetic and real-world datasets.

2 Related Work

Hubness has been observed in several fields of applications involving sound and image data during recent years (Aucouturie and Pachet, 2007; Doddington et al., 1998; Hicklin et al., 2005) and, in addition, Jebara et al. Hubness phenomenon in the construction of the neighborhood graph of learning (Tony Jebara et al 2009) [2]. Amina M et al. designed a hubness-based algorithm by introducing hubs into the k-means algorithm (Amina M et al 2015) [3]. Although people does not give much attention to the phenomenon of hubness in data clustering, the k-nearest-neighbor list is widely used in many clusters. The k-nearest-neighbor list computes the density estimate by the volume used to observe the space determined by the k nearest neighbors. Density-based clustering methods generally rely on this density estimation. The main goal of density-based clustering algorithms is to find high-density areas separated by low-density areas [4]. In high-dimensional space, this is often difficult to estimate because data is usually sparse. It is also important to choose the appropriate neighborhood size, because too small or too large k values may cause the density-based approach to fail. The k-nearest-neighbor list is often used to construct k-NN graphs and so it's used in graph clustering.

3 Solutions

To explore the relation between hubs and clusters, I will do experiments on both synthetic datasets and real-world datasets.

First part of the experiment is performed on the synthetic datasets. We run K-Means on each gaussian mixture. In each iteration, we measure the distance from the current cluster centroid to its closest neighbor and to the strongest hub, and scaled by the average intracluster distance. Both minimal and maximal distance from any of the centroids to the corresponding hub and their closes neighbors will be computed.

Second part of the experiment is done on the real-world datasets. The thing about real-world dataset is that the number of clusters is not given. So my idea is to first run DBSCAN on the dataset and count the number of core points as the number of clusters. Then I will run K-Means with the given number of clusters and compute the minimal and maximal distance during each iteration.

To explore the relation between hubs and DBSCAN outliers, experiments will also be done on both synthetic datasets and real-world datasets.

First part of the experiment is performed on the synthetic datasets. We run DBSCAN on each gaussian mixture to find the outliers. And then we compute the hubness score of all the points in the dataset. For the outliers, we check if their hubness scores are two standard deviations lower than the average value.

Second part of the experiment is done on the real-world datasets. I first run DBSCAN on the dataset and find the outliers. Then I will do the same comparison between the outliers hubness scores and the average values.

4 Experiments

4.1 Data

Two types of data will be used including synthetic and real-world data sets.

For synthetic datasets, I mean random generated gaussian mixtures. We generate data from a given list of dimensions: 2, 5, 10, 20, 30, 50, 100. We have two kinds of mixtures, one using five gaussian generators so that it will have 5 data clusters, the other using 10 and it will have 10 data clusters. For the case of a dimension and a number of clusters, we generate 10 different gaussian mixtures. So we will have 7210=140 gaussian mixtures. And we will randomly generate 10000 points for each mixture.

For the real world datasets, I take several UCI datasets. And they are described in an ascending order of their dimension.

Iris Dataset. It is perhaps the best known database to be found in the pattern recognition literature. It has 150 instances and the data dimension is 4. It can be downloaded from https://archive.ics.uci.edu/ml/datasets

Abalone Dataset. The data comes from predicting the age of abalone from physical measurements. It has 4177 instances and the data dimension is 8. It can be downloaded from https://archive.ics.uci.edu/ml/dataset

Breast Cancer Wisconsin (Prognostic) Dataset. Each record represents follow-up data for one breast cancer case. It has 198 instances and the data dimension is 33. It can be downloaded from https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Prognostic).

Sonar Dataset. The data is obtained by bouncing sonar signals off a metal cylinder at various angles and under various conditions. It has 208 instances and the data dimension is 60. It can be downloaded from http://archive.ics.uci.edu/ml/datasets/connectionist+bench+(sonar,+mines+vs.+rocks).

Hill-Valley Data Set. Each record represents 100 points on a two-dimensional graph. When plotted in order as the Y co-ordinate, the points will create either a Hill or a Valley. It has 606 instances and the data dimension is 100. It can be downloaded from http://archive.ics.uci.edu/ml/datasets/hill-valley.

4.2 Experiment Setup

To make sure that the code to calculate hubness scores is correct, I will first replicate the experiment settings mentioned in [1] to see if I get similar results.
To evaluate our experiments on hubs distribution across clusters, we will draw some plots to show how the maximal distance and minimal distance evolve through iterations. Data dimensions below 10 are used to illustrate low-dimensional behavior while dimensions above 10 are used to illustrate high-dimensional behavior. For gaussian mixtures with 5 generators, the neighborhood size is 5 while for gaussian mixtures with 10 generators, the neighborhood size is 10.
To test if my implementation of DBSCAN is correct, I will run it on the synthetic data and see if the found core points are data clusters.
We will draw histograms to show the distribution of outliers. The y-axis will be the number of outliers and the x-axis will show their difference to the average hubness scores using one standard deviations as a unit.

4.3 Experimental Results

5 Conclusion

6 Contribution

Yimeng Li finished exploring the relation between hign-hubness points and cluster modoids. He also finished exploring the relation between low-hubness points and DBSCAN-detected noise points.