**PR3: Text Clustering**

**Name**

Varun Shah

**Student ID**

010823657

**Rank**

30

**NMI**

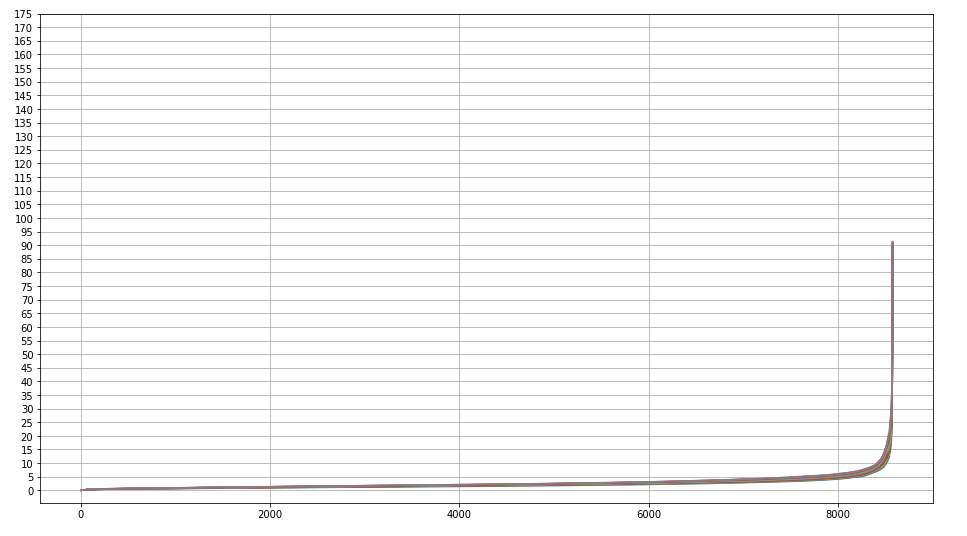
0.1342

**DBSCAN Algorithm PseudoCode**

Following are the steps for performing DBSCAN

1. Once the radius eps and density is decided, first step is to identify which documents are core points. For that:
   1. Sort the row of Matrix and check for column index = ‘density’ is less than or equal to Eps. If it is lesser than Eps than the document is core point else it is a noise or border which will be clustered later. Store core document index and non-core document index separately
2. Now, for clustering of core points following steps are taken:
   1. For current core point add all the other core points that fall within the Eps radius from current core point and mark all these clustered points as visited
   2. Now, for each clustered point mapped to current core point’s cluster, find other unvisited cluster point that fall within its Eps radius
   3. Follow this way until there is no unclustered core point
3. Once, Core points are clustered, points that were marked non-core points needs to be categorized as border point or noise point in following way:
   1. For each non-core point, check for which cluster points it falls within the Eps radius of cluster’s core points and add it to the cluster it is closest to.
   2. All the points that did not fall into the Eps radius of any core points of any cluster, create one cluster identified as noise points cluster

**Determining Radius Eps for MinPts**

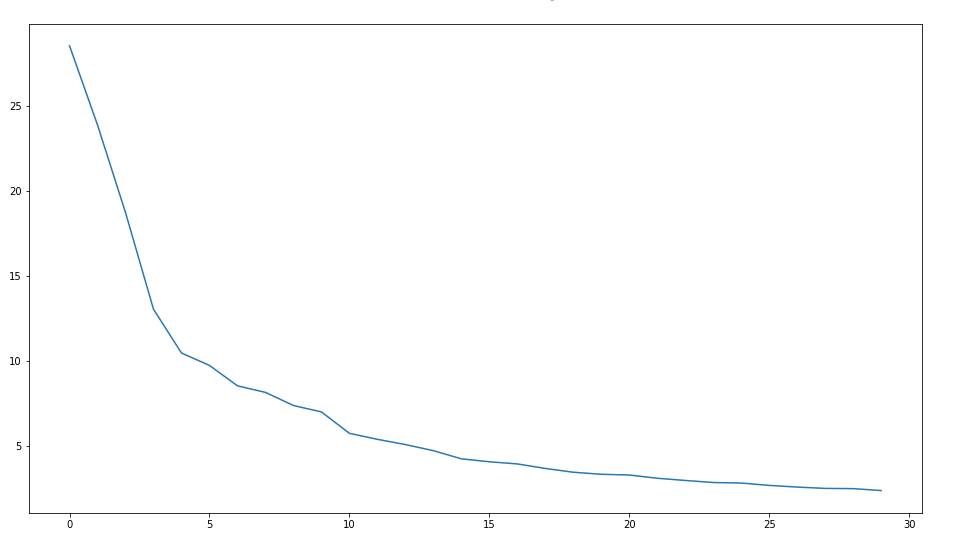


1. As Shown above, graph depicts that, all the documents are coupled and eps variance is low.
2. So, in this assignment Eps value is taken 3.5 for which graph depicts that core points should be around 5800-6000. While computing for identifying core points, 5813 core points are found
3. From above observation, it can be deduced that 5813 are core points for 3.5 Eps radius and density 15

**Feature Selection/Reduction**

Performed following steps for data preprocessing:

1. Read data from “train.dat” and created index, value and pointer arrays for each document. Instead of using keys provided in document, mapped those key with assigned unique index and used that index to track corresponding value. That way it reduces dimensionality from max key-value to total unique keys.
2. Using CSR Matrix of Scipy created CSR matrix of train data from index, value and pointer
3. Using truncated SVD, performed dimensionality reduction to reduce it to 8 components based on following graph:



1. Then, computed pairwise distance of each document to other document and created square distance matrix