Disclaimer

- The material provided in this document is not my original work and is a summary of some one else's work(s).
- A simple Google search of the title of the document will direct you to the original source of the material.
- I do not guarantee the accuracy, completeness, timeliness, validity, non-omission, merchantability or fitness of the contents of this document for any particular purpose.
-

Document Clustering Techniques

Clustering and Information Retrieval
Presented
By
Najeeb

Outline

- Introduction
- Document Representation
- Distance Measures
- Techniques
  - Hierarchical
  - Partitioning
  - Density Based
  - Model Based
  - Soft Computing Based
- Comparison

Introduction

- Document Clustering
  - Unsupervised classification of documents into groups such that documents in a cluster are similar, whereas documents in different clusters are dissimilar

Document Representation

- A text document is an unstructured data type
- We need to represent each document as a structured data type so that our machine learning algorithms can work on it
- Many structured representations for text documents have been proposed
- The most common of them is the TF-IDF Vector Space Model

Document Representation

- Binary term-document incidence matrix

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

- Each document is represented by a binary vector $\in \{0,1\}^{|V|}$

Document Representation

- Term-Frequencies: The term frequency $tf_{t,d}$ of term *t* in document *d* is defined as the number of times that *t* occurs in *d*

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

- Each document is a count vector in $\mathbb{N}^v$

Document Representation

- Problem with Term-Frequencies
  - A document with 10 occurrences of the term is more relevant than a document with 1 occurrence of the term
  - But not 10 times more relevant
  - Relevance does not increase proportionally with term frequency
  - Many alternatives: Log-frequency weighting, TF-IDF etc.

Document Representation

- Document frequency
- Rare terms are more informative than frequent terms
- We want a high weight for rare terms like “Turing test ” than for frequent terms like “the”
- For rare terms we want higher weights while for frequent terms we want lower weights

Document Representation

- Inverse Document frequency
  - df_t is the document frequency of *t*: the number of documents that contain *t*
  - *We define the idf (inverse document frequency) of t by*

$$\mathrm{idf}_t = \log_{10}(N/\mathrm{df}_t)$$

Document Representation

- Inverse Document frequency

term	df_t	idf_t
calpurnia	1	6
animal	100	4
sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

Document Representation

- tf-idf weighting
  - The tf-idf weight of a term is the product of its tf weight and its idf weight

$$\mathrm{w}_{t,d} = \log(1 + \mathrm{tf}_{t,d}) \times \log_{10}(N / \mathrm{df}_t)$$

  - Best known weighting scheme in information retrieval

Document Representation

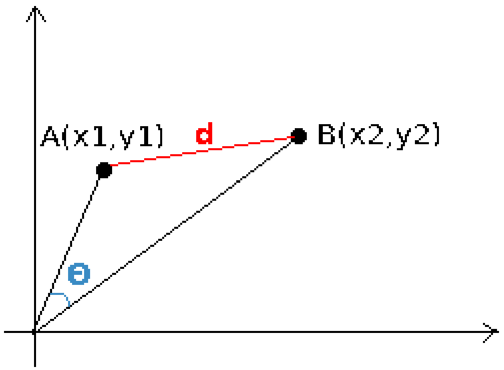
	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

Document Representation

- So we have a |V|-dimensional vector space
- Terms are axes of the space
- Documents are points or vectors in this space
- Very high-dimensional: 5000+ for one of the dataset I used
- These are very sparse vectors - most entries are zero.

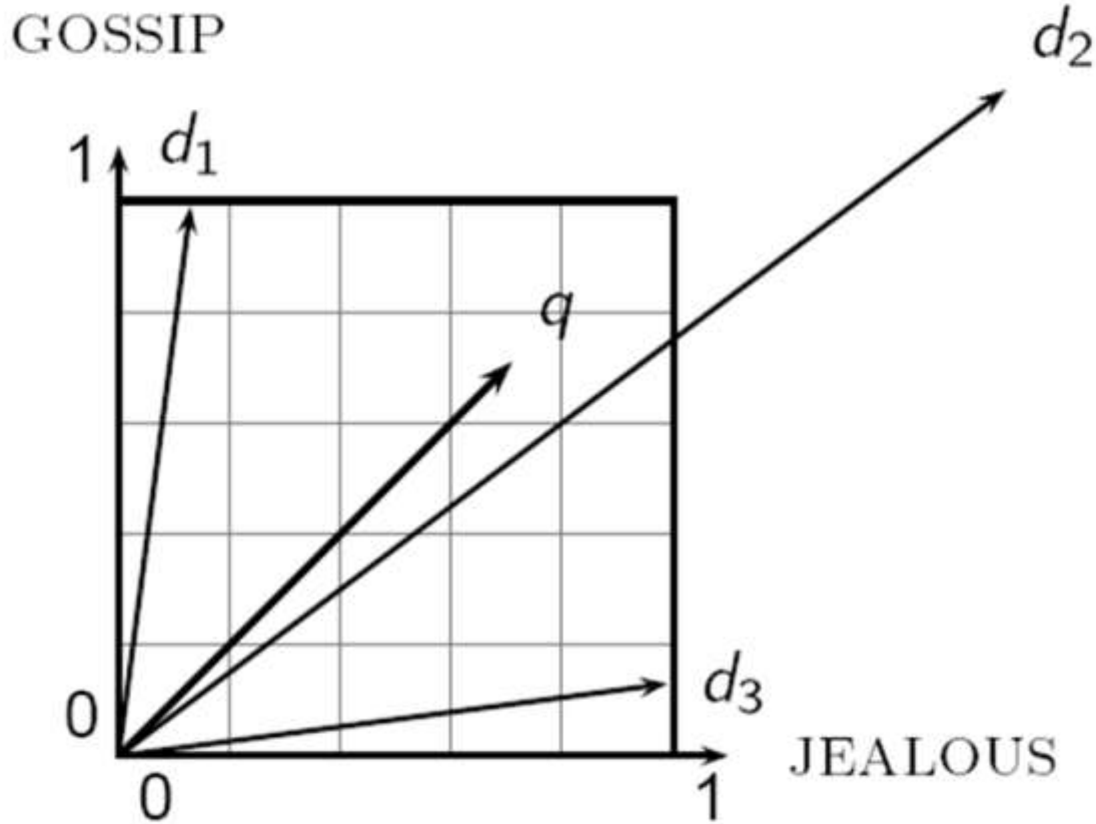
Distance Measures

- Euclidean Distance



Distance Measures

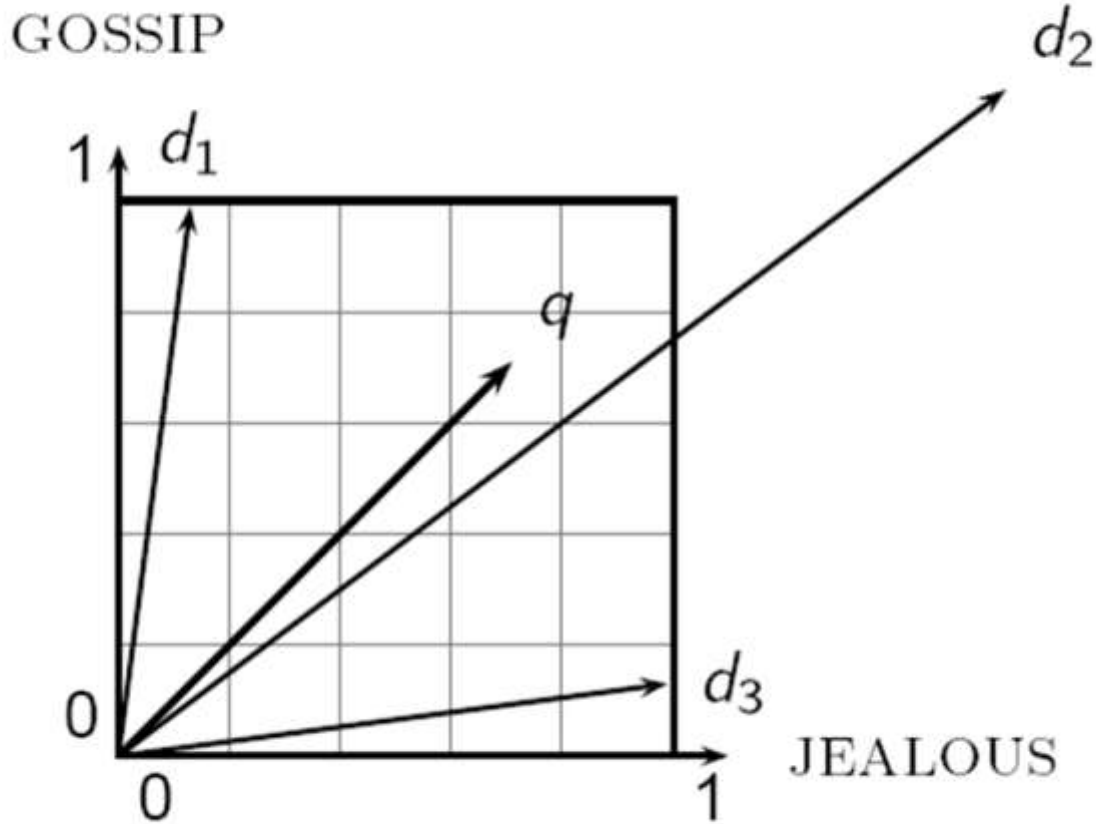
- Euclidean Distance



Distance Measures

- Euclidean Distance
- Cosine Similarity

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



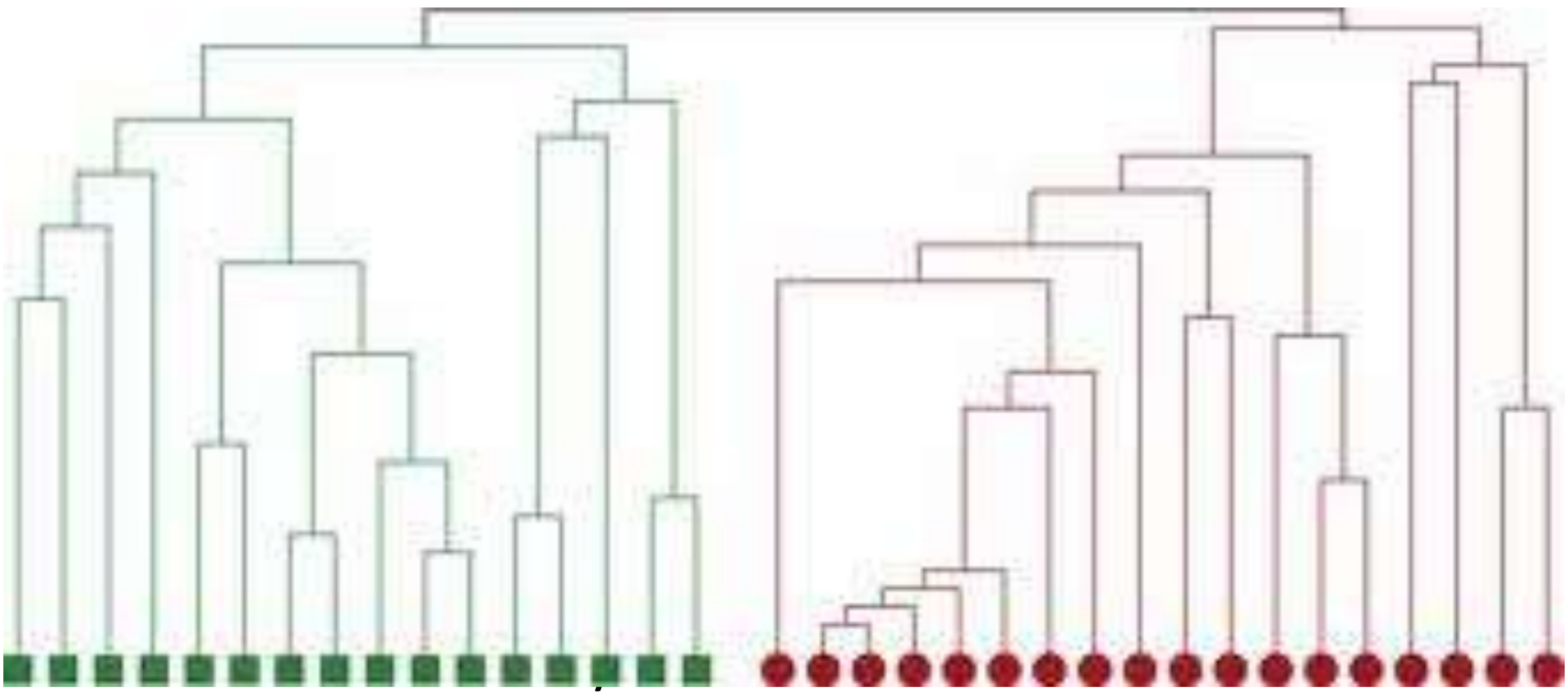
Clustering Methods

- Clustering algorithms can be broadly divided into
  - Hierarchical
  - Partitioning
  - Density Based
  - Model Based
  - Soft Computing Based

Hierarchical Clustering

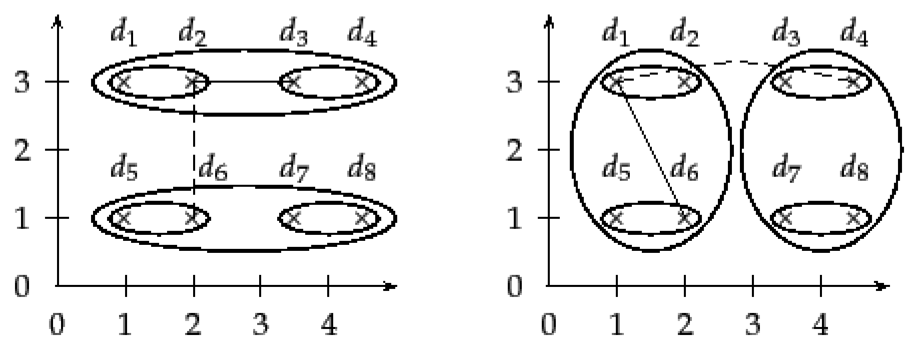
- Agglomerative hierarchical clustering
  - Each object initially represents a clusterof its own
  - Then clusters are successively merged until the desired cluster structure is obtained
- Divisive hierarchical clustering
  - All objects initially belong to one cluster
  - Then the cluster is divided into sub-clusters, which are successively divided into their own subclusters

Hierarchical Clustering



Hierarchical Clustering

- Single Link Clustering (good for non isotropic)
- Complete Link Clustering
- Average link Clustering

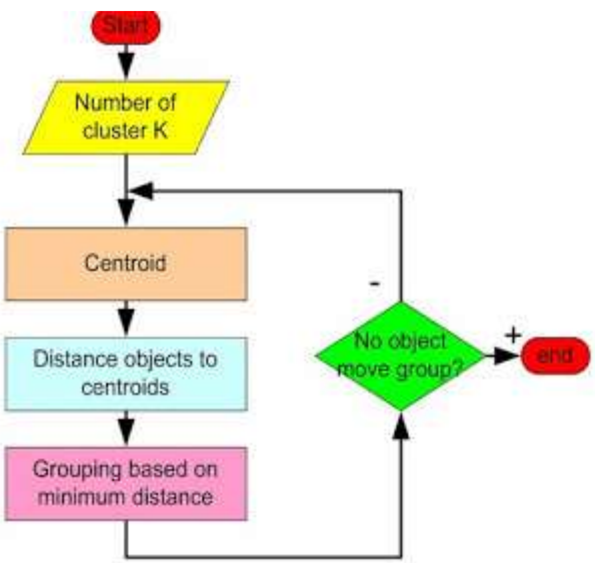


Partitioning Methods

- Partitioning methods relocate instances by moving them from one cluster to another, starting from an initial partitioning
- Partitioning methods usually converge to a local minimum
- Some heuristic is used for achieving global optimality

Partitioning Methods

- K-means
- K-Medoid: Each cluster is represented by the most centric object in the cluster, rather than by the implicit mean that may not belong to the cluster



Density Based Methods

- Density-based methods assume that the points that belong to each cluster are drawn from a specific probability distribution
- Density based clustering algorithms include
  - DBSCAN (Arbitrary Shape)
  - AUTOCLASS (Gaussian, Bernoulli, Poisson, and log-normal distributions)
  - MCLUST
  - SNOB

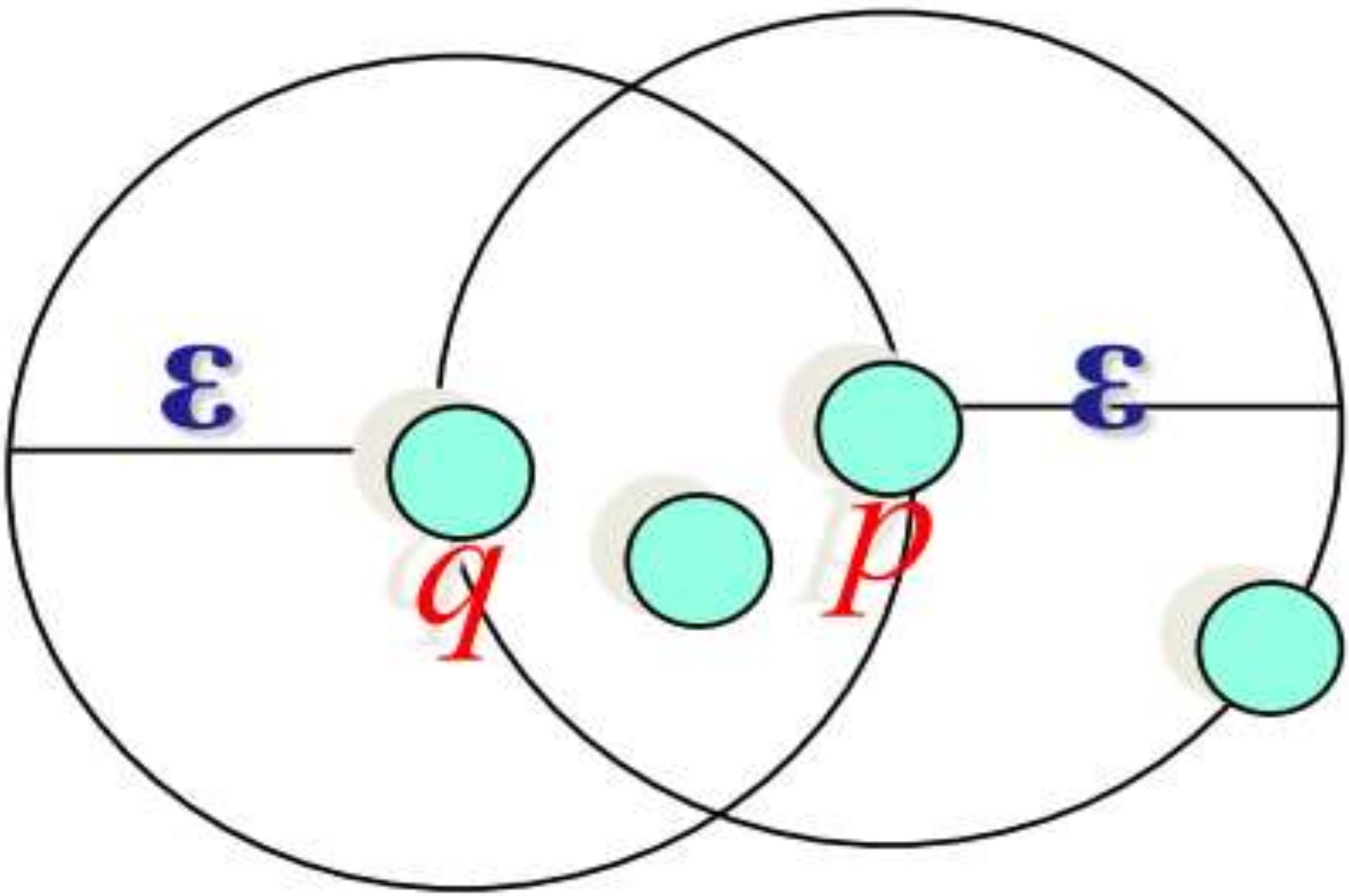
Density-based spatial clustering of applications with noise (DBSCAN)

- Clusters are dense regions in the data space, separated by regions of lower object density
- A cluster is defined as the maximal set of density connected points
- ε-Neighborhood: Objects within a radius of ε from an object

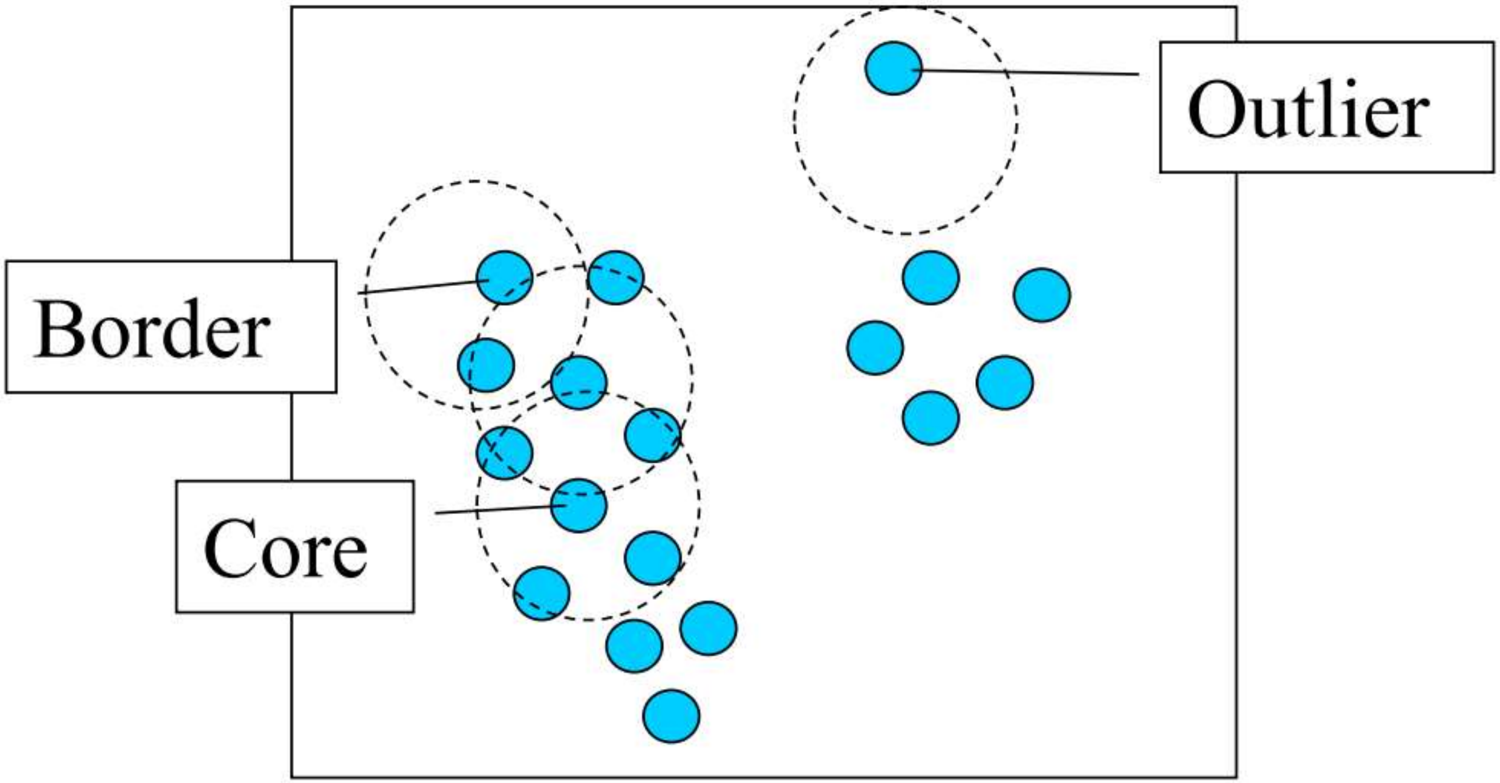
$$N_{\varepsilon}(p):\{q \mid d(p,q) \leq \varepsilon\}$$

- "High density" - ε-Neighborhood of an object contains at least MinPts of objects

Density-based spatial clustering of applications with noise (DBSCAN)



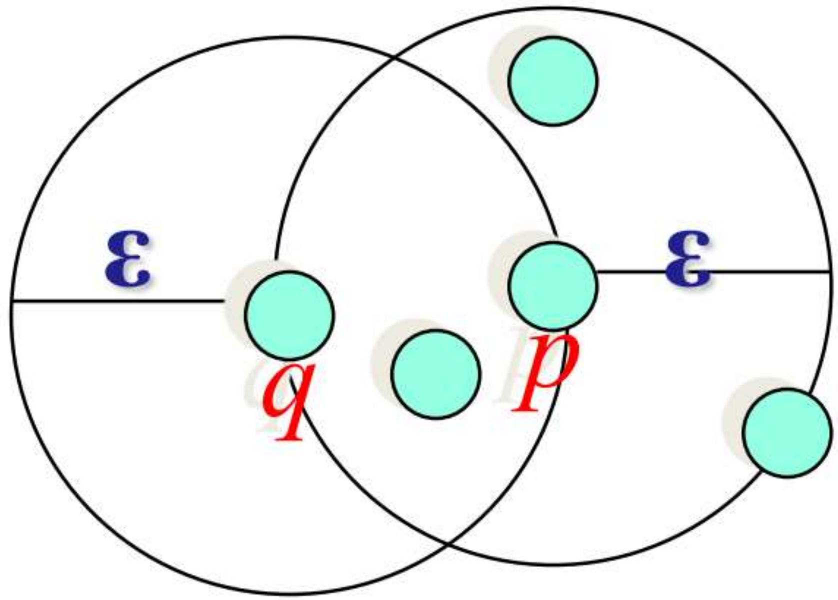
Density-based spatial clustering of applications with noise (DBSCAN)



ε = 1unit, MinPts = 5

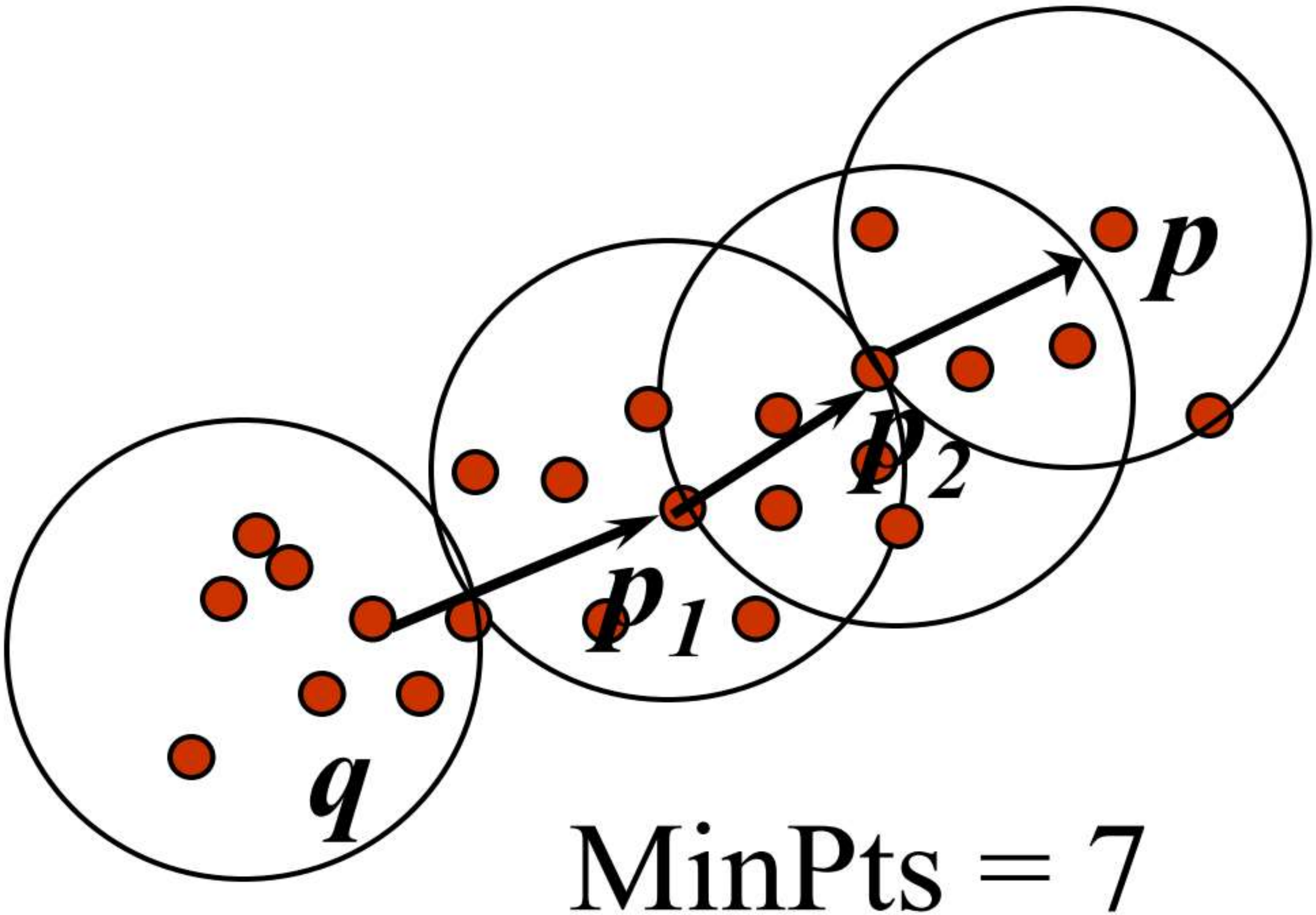
Density-based spatial clustering of applications with noise (DBSCAN)

- Directly Density-Reachable



MinPts = 4

Density-based spatial clustering of applications with noise (DBSCAN)



Density-based spatial clustering of applications with noise (DBSCAN)

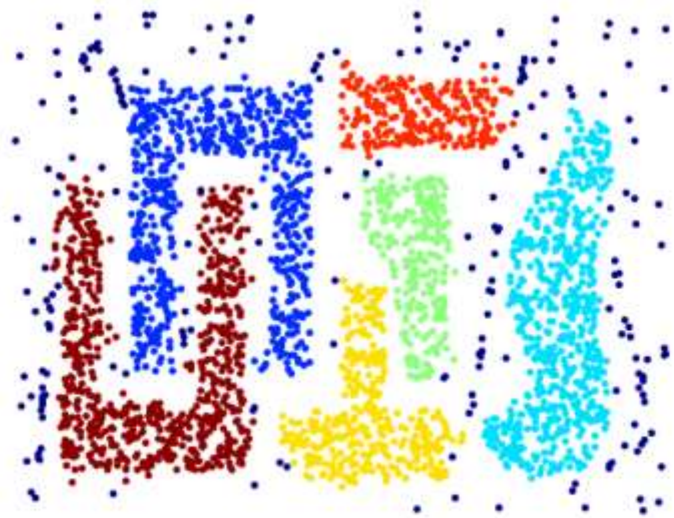
- The algorithm

```
for each o ∈ D do
    if o is not yet classified then
        if o is a core-object then
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE
```

Density-based spatial clustering of applications with noise (DBSCAN)

- Advantages:
  - Resistant to Noise
  - Can handle clusters of different shapes and sizes

Original Points



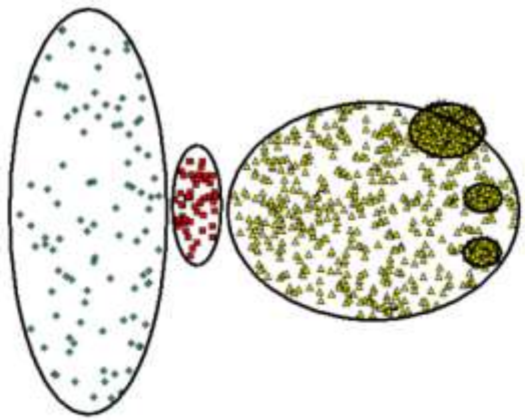
Clusters

Density-based spatial clustering of applications with noise (DBSCAN)

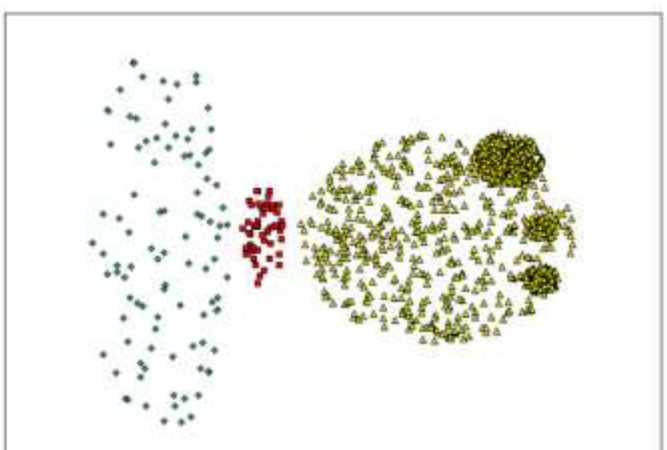
- Disadvantages:
  - Cannot handle varying densities
  - Sensitive to parameters—hard to determine the correct set of parameters

Density-based spatial clustering of applications with noise (DBSCAN)

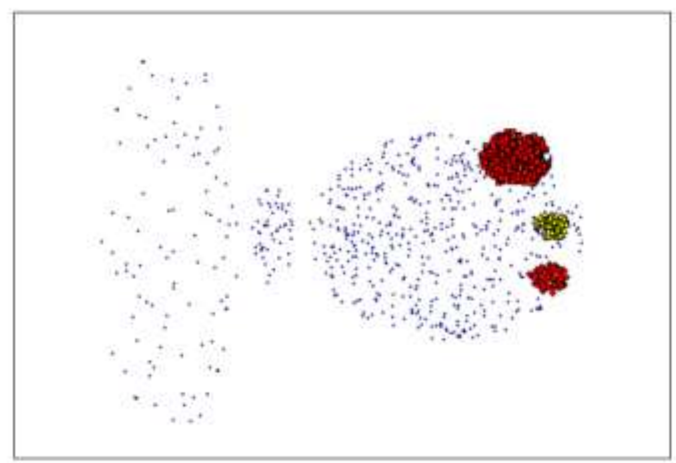
- e



Original Points



(MinPts=4, Eps=9.92).



(MinPts=4, Eps=9.75)

Model Based

- These methods attempt to optimize the fit between the given data and some mathematical models
  - Decision Trees
  - Self Organizing Maps

Soft-computing Methods (Fuzzy Clustering)

- Traditional clustering approaches generate partitions; in a partition, each instance belongs to one and only one cluster
- In fuzzy clustering each pattern is associated with every cluster using some sort of membership function, namely, each cluster is a fuzzy set of all the patterns
- Larger membership values indicate higher confidence in the assignment of the pattern to the cluster

Soft-computing Methods (Fuzzy Clustering)

- Fuzzy C Means

objective function

$$J_h = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij} d_{ij}^2.$$

$$J_f = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2.$$

new membership weights:

$$u_{ij} = \begin{cases} 1, & \text{if } i = \operatorname{argmin}_{l=1}^{c} d_{lj} \\ 0, & \text{otherwise} \end{cases},$$

$$u_{ij} = \frac{1}{\sum_{l=1}^{c} \left(\frac{d_{ij}^2}{d_{lj}^2} \right)^{\frac{1}{m-1}}} = \frac{d_{ij}^{\frac{-2}{m-1}}}{\sum_{l=1}^{c} d_{lj}^{\frac{-2}{m-1}}},$$

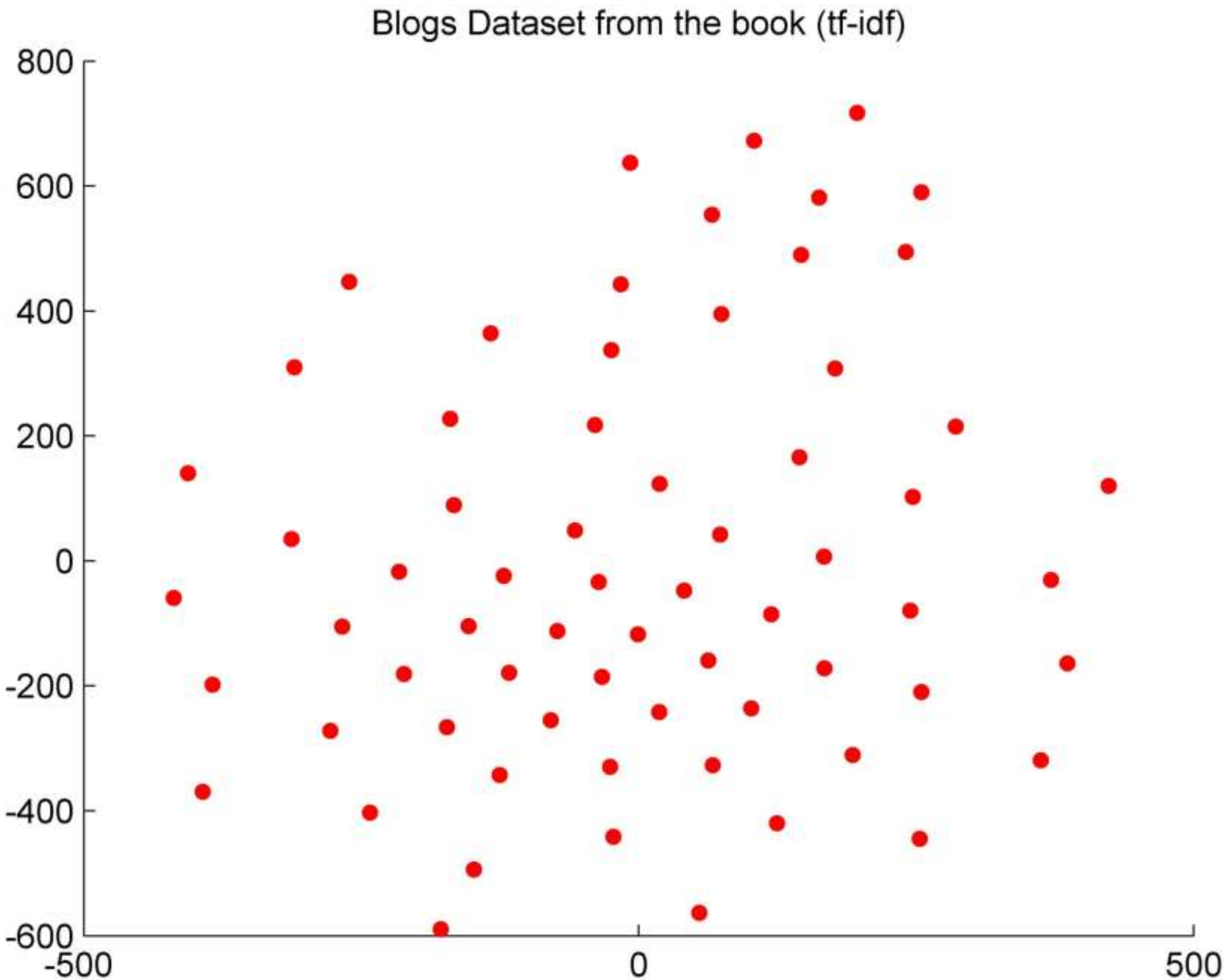
new cluster centres:

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij} \mathbf{x}_j}{\sum_{j=1}^{n} u_{ij}}$$

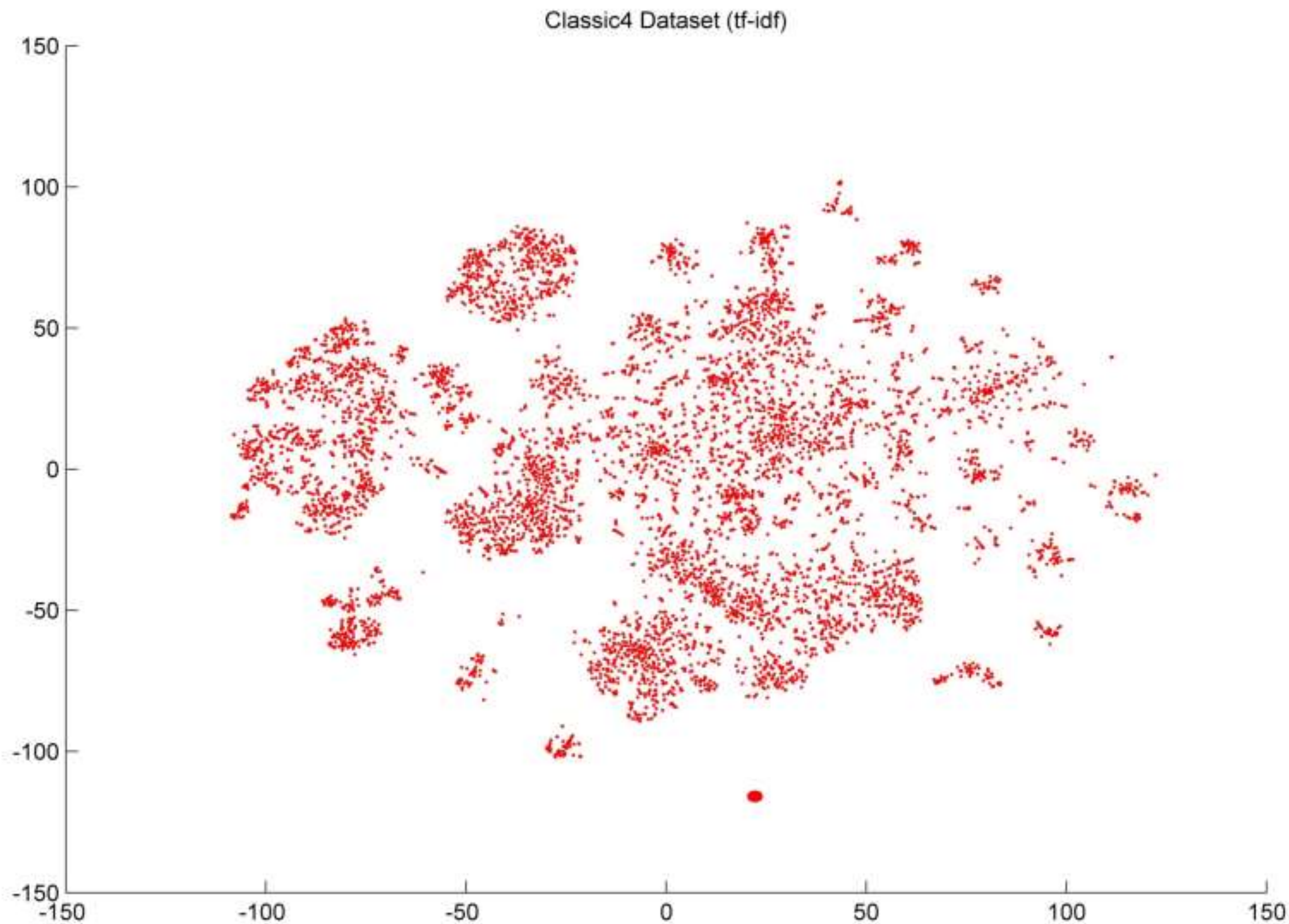
$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^{n} u_{ij}^m}.$$

DataSet

- Originally I tried to work with the dataset given in this book
- It contains 75 blogs however only 65 of them can be retrieved currently
- The distribution of the dataset is somewhat uniform
- (too small and too bad)


Blogs Dataset from the book (tf-idf)
800
600
400
200
0
-200
-400
-600
-500
0
500

DataSet

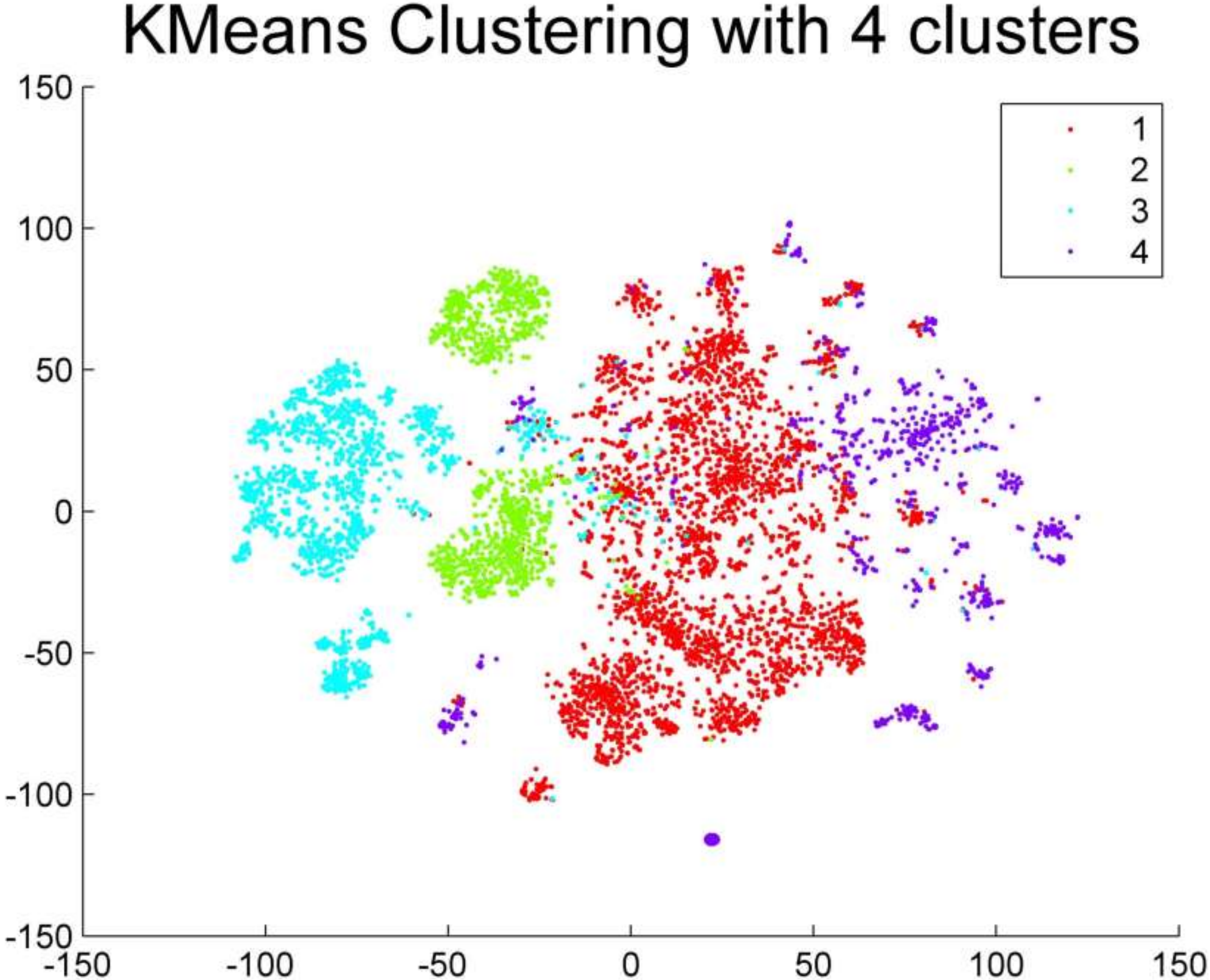


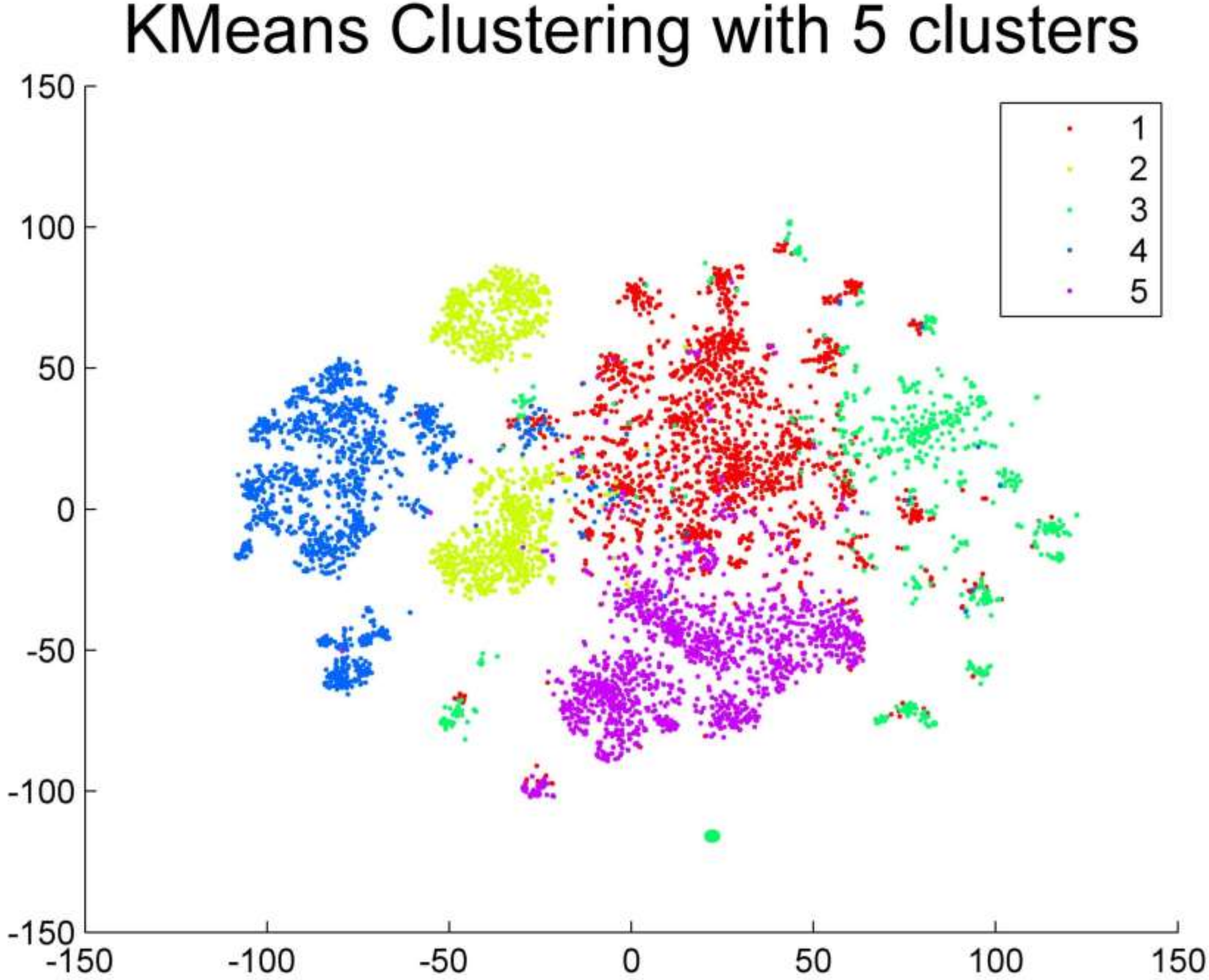
Classic4 Dataset (tf-idf)
150
100
50
0
-50
-100
-150
-150
-100
-50
0
50
100
150

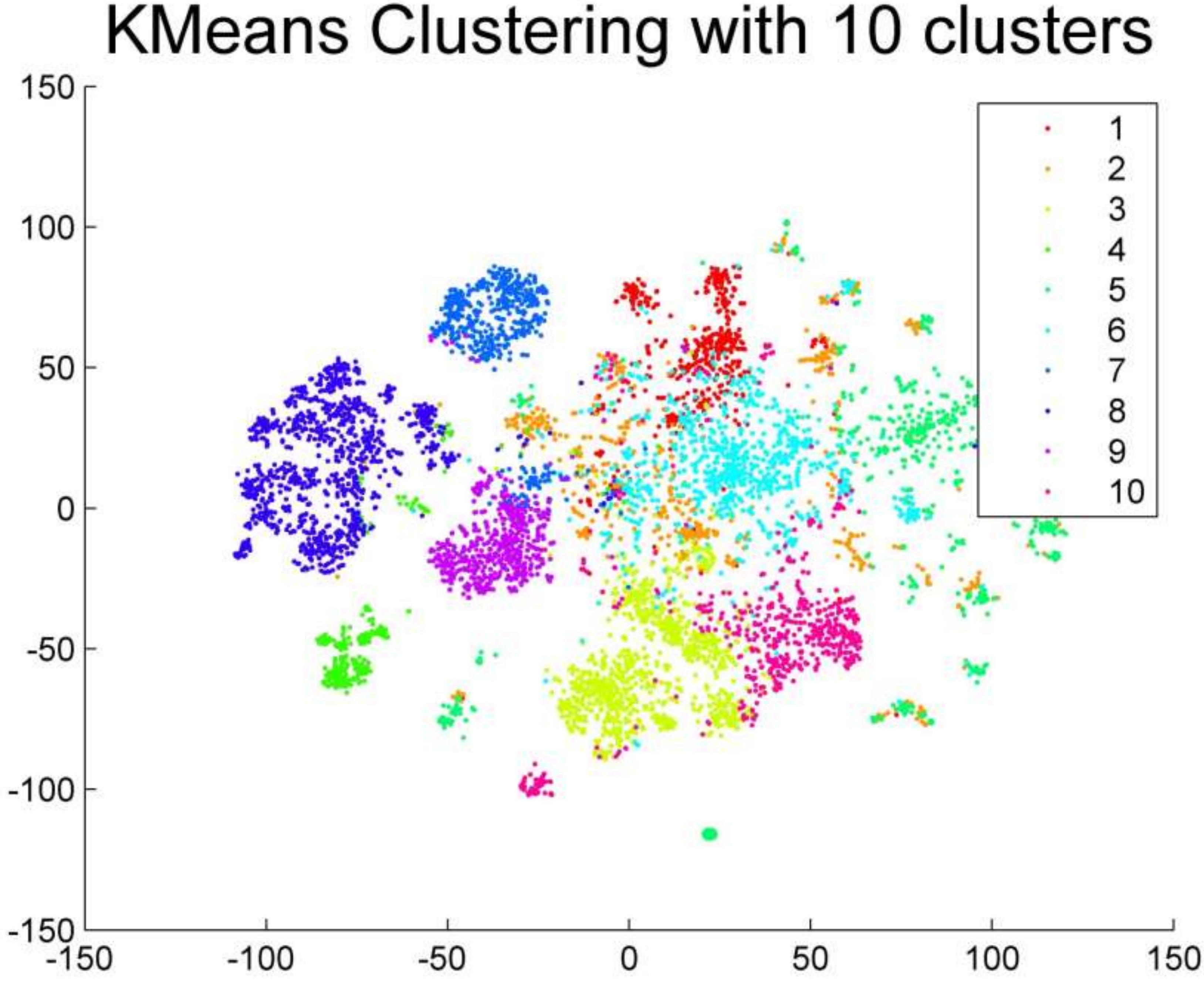
DataSet

- Many papers used very large datasets including
  - Classic4
  - Newsgroup20
  - Reuters-21578
- I have chosen the Classic4 dataset
- After keeping only stem words and removing stop words
  - There are 7095 documents
  - 5896 terms ☹ (unique words)

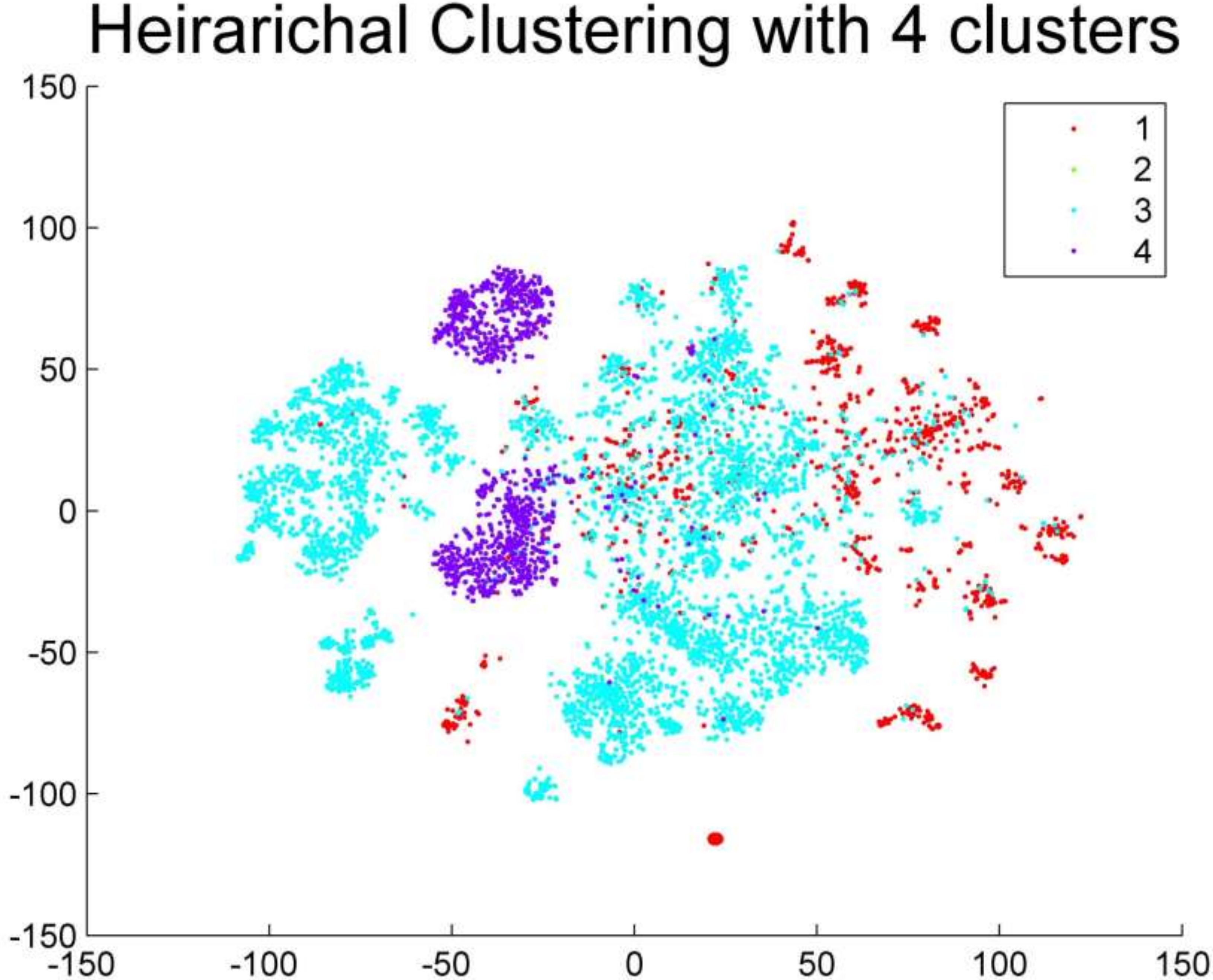
Results


KMeans Clustering with 4 clusters
150
100
50
0
-50
-100
-150
-150
-100
-50
0
50
100
150
1
2
3
4

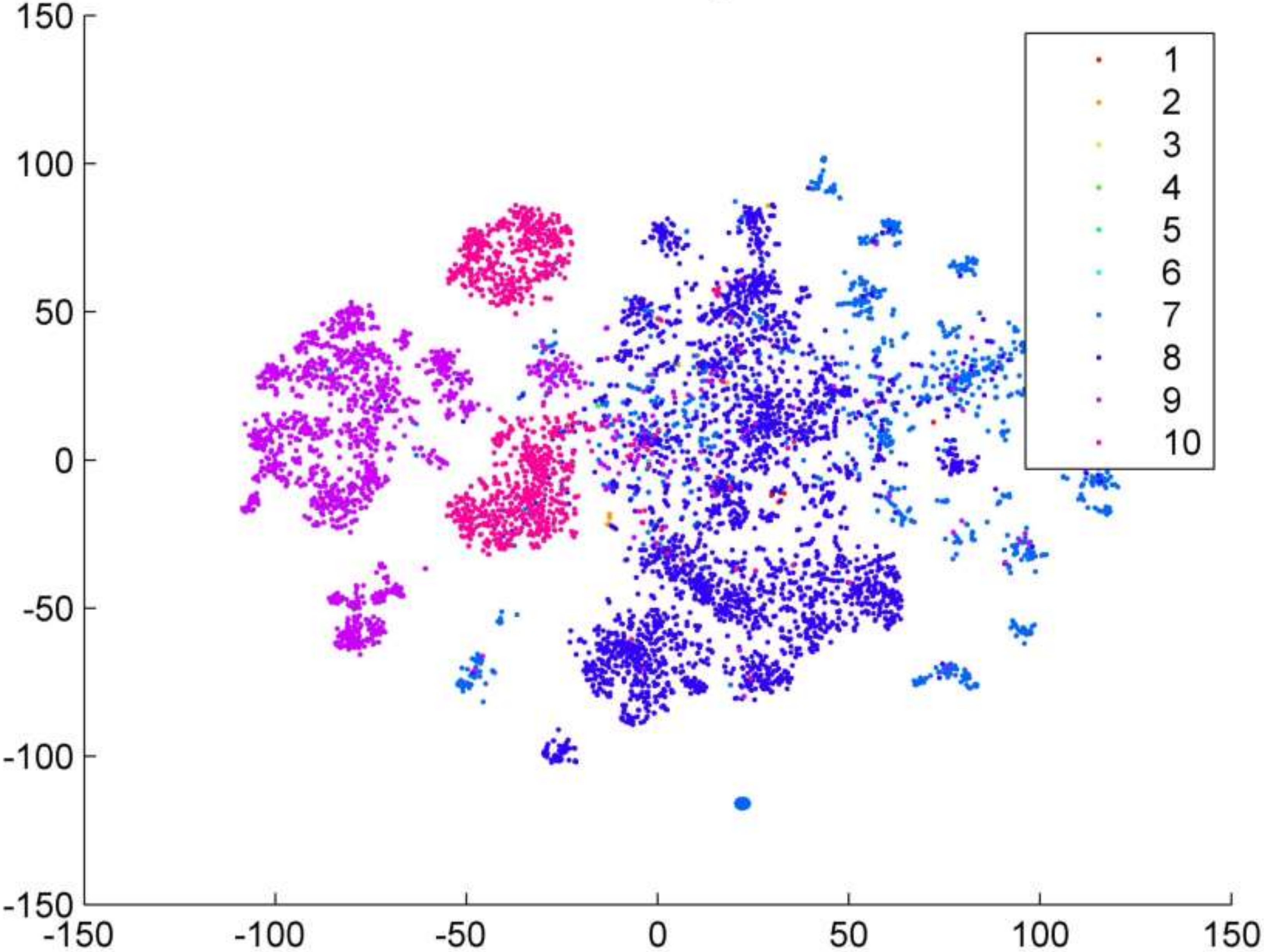

KMeans Clustering with 5 clusters
150
100
50
0
-50
-100
-150
-150
-100
-50
0
50
100
150
1
2
3
4
5

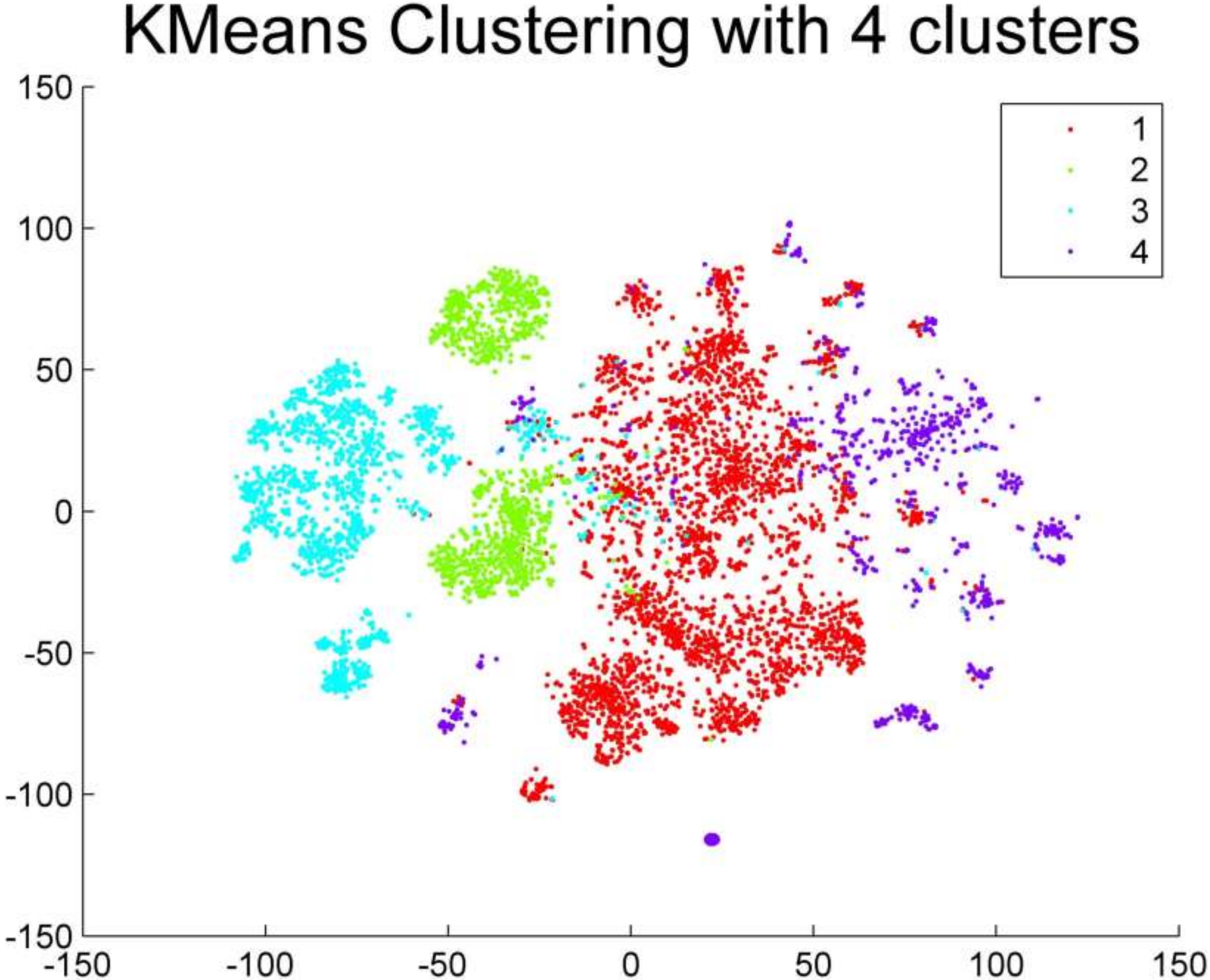


KMeans Clustering with 10 clusters
150
100
50
0
-50
-100
-150
-150
-100
-50
0
50
100
150
1
2
3
4
5
6
7
8
9
10


Heirarichal Clustering with 4 clusters
150
100
50
0
-50
-100
-150
-150
-100
-50
0
50
100
150
1
2
3
4

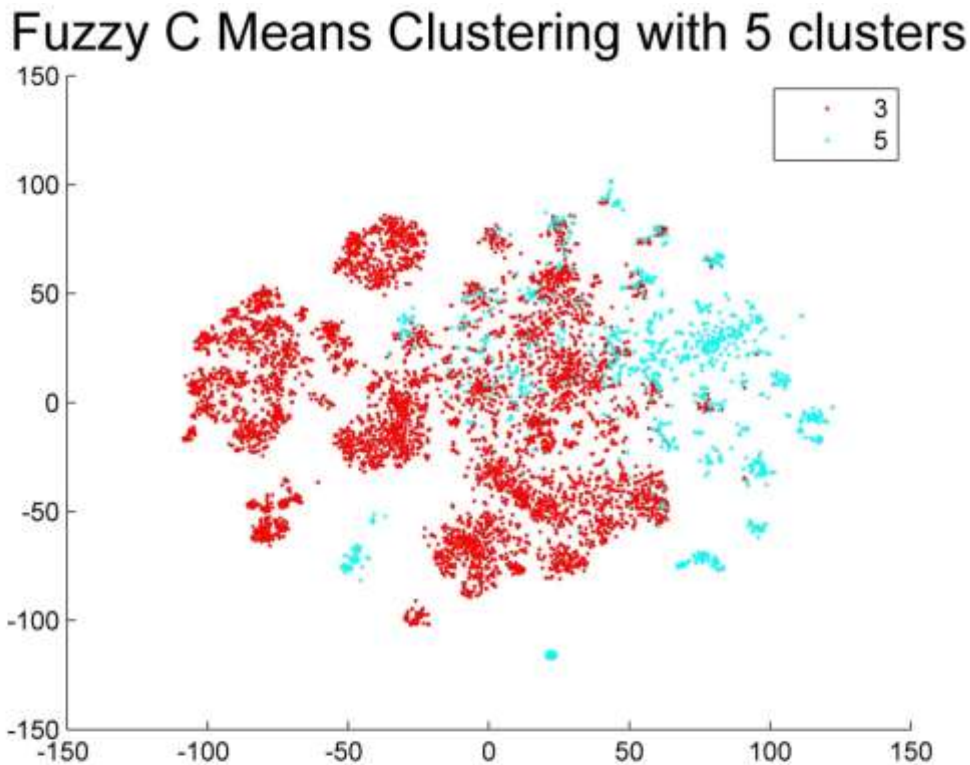
Heirarichal Clustering with 10 clusters




KMeans Clustering with 4 clusters
1
2
3
4
150
100
50
0
-50
-100
-150
-150
-100
-50
0
50
100
150

Fuzzy C-Means

- I also applied the FCM algorithm but for some reason its result is not correct



Refrences

- Rokach, Lior. "A survey of clustering algorithms." Data mining and knowledge discovery handbook. Springer US, 2010. 269-298.
- Premalatha, K., and A. M. Natarajan. "A literature review on document clustering."*Information Technology Journal* 9.5 (2010): 993-1002.
- Singh, Vivek Kumar, Nisha Tiwari, and Shekhar Garg. "Document Clustering using K-means, Heuristic K-means and Fuzzy C-means." Computational Intelligence and Communication Networks (CICN), 2011 International Conference on. IEEE, 2011.
- Jursic, Matjaz, and Nada Lavrac. "Fuzzy clustering of documents." (2008).

Thanks