Application of PCA in Text Document Clustering

DWDM PROJECT REPORT

by

Rishi Sharma (171CO135)
Vedant Mehra(171CO250)

Department of Computer Science
National Institute of Technology Karnataka, Surathkal

November 2020

1 Abstract

Text Document Clustering is a very strong text mining approach for discovery of topics from textual documents. It is a useful technique that organises a large number of unordered text documents into a small number of meaningful and coherent clusters, thereby providing a basis for intuitive and informative navigation and browsing mechanisms. Graph-based clustering algorithms, such as Spectral Clustering, are reported performing well on document clustering. They treat the clustering problem as an optimisation process of grouping documents into K clusters so that a particular criterion function is minimised or maximised. Most of the existing text clustering algorithms take in features vector directly as an input and make the corresponding prediction. This research work aims at applying Dimensionality Reduction techniques such as Principle Component Analysis(PCA) on the features vector instead of feeding-in directly as input to Spectral Clustering so that we can reduce dimensions of input features vector to make the computation faster and achieving the same level of accuracy as we were getting when the dimensions were high. *Euclidean distance*, *Gaussian kernel* was used for computing the weights of the adjacency matrix of the graph. The concluding results show that by applying PCA to input data and using *Gaussian Kernel* as a distance measure, we can achieve the same level of accuracy as compared when the dimensions of input features vectors were relatively high.

2 Table of Contents

3 List of Figure

4 List of Tables

4 Chapters

4.1 Introduction

One of the critical questions in areas of information retrieval and text mining is how to explore and utilise the massive amount of text documents. To help users effectively navigate, summarise, and organise text documents, *Text Document Clustering* is one of the most crucial text mining methods which have developed in recent years. A considerable number of records can be easily organised into several meaningful clusters using document clustering. Text document clustering can also be used in many situations to browse a collection of documents or collect the results obtained using some search engine to respond to some user's query[2]. This can increase the recall and precision in information retrieval systems [3], [2], [4], [5], and it is a way which is very efficient to find the nearest neighbours of a document [6]. The problem of document clustering is usually defined as follows: given a set of documents, we would like to divide them into an automatically derived number of clusters, such that the documents assigned to each cluster are more similar to each other than the documents assigned to different clusters. In other words, the documents in one cluster share the same topic, and the documents in different clusters represent various topics.

Document Clustering has been among the most active research topics in *Information Retrieval* and *Natural Language Processing*. Whereas many traditional document clustering algorithms have been developed over the past few decades [8], [9], various types of new ideas in clustering have emerged over the last few years that give very promising results on some challenging tasks. Among them are *Spectral Clustering* [10], [11], [12], [13] and *Path-based Clustering* [14], [15], [16], which have demonstrated excellent performance on some clustering tasks involving highly non-linear and elongated clusters in addition to compact clusters.

It has been reported that there are two general categories of clustering methods: Agglomerative hierarchical and partitional methods. *Agglomerative hierarchical clustering* (AHC) algorithms initially treat each document as a cluster, use different kinds of distance functions to compute the similarity between the pairs of clusters, and then merge the closest pair [17]. This merging step is repeated until the desired number of clusters is obtained. Compared with the bottom-up method of AHC algorithms, the family of *k-means* algorithms [18], [19], [20], [21], which belong to the category of partitional clustering, create one-level partitioning of the documents. The *k-means* algorithm is based on the idea that a centroid can represent a cluster. After selecting *k* initial centroids, each document is assigned to a cluster based on a distance measure, then *k* centroids are recalculated. This step is repeated until an optimal set of *k* clusters are obtained based on a criterion function. Text document clustering is a basic process used in information retrieval, automatic topic extraction and document organization. High quality clustering algorithms play a vital role efficiently in organizing, summarizing and navigating the unstructured documents.

Despite the promising performance of these algorithms demonstrated on some difficult data sets, there exist some other situations when these algorithms do not perform well. Consider some examples in Fig 4.1. Though it is known that spectral clustering works perfectly well on the 2-circle data set (Fig 4.1(a)), it gives very poor results on the 3-spiral data set (Fig 4.1(b)). The poor result of clustering is mainly because of the specific choice of the affinity matrix, which is usually defined in a way which is similar to the *Gaussian Kernel* based on inter-point *Euclidean Distance* in the input space. However, if path-based clustering which is part of path-based criteria are used to define the similarity or dissimilarity between points to form the affinity matrix before spectral clustering is applied, the three clusters in the 3-spiral data set can be found correctly, as shown in Fig 4.1(c).

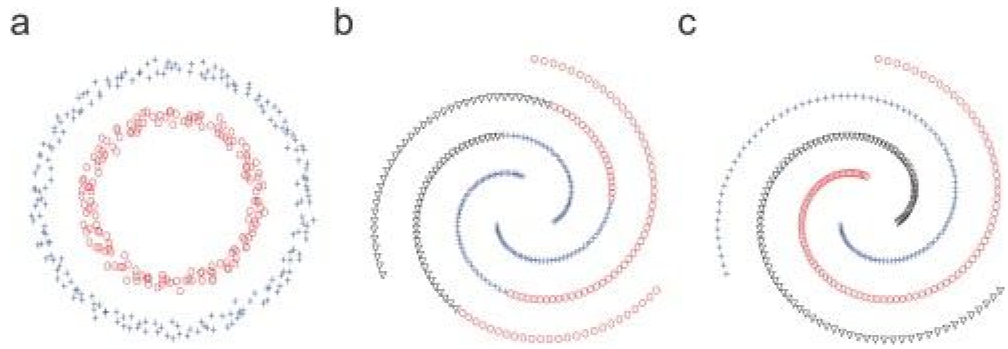

Fig. 4.1 Simple examples of spectral clustering and path-based clustering: (a) spectral clustering result for 2-circle data set; (b) spectral clustering result for 3-spiral data set; and (c) path-based spectral clustering result for 3-spiral data set

In this research work we aim at applying *Principle Component Analysis*(PCA) which is well-known *Dimensionality Reduction* techniques on the input features vector which is very high dimensional vector instead of feeding-in directly as input to *Spectral Clustering* so as to reduce dimensions of input features vector for making the faster computation while achieving the similar level of accuracy when the dimensions of input features vector was high. To achieve this, we made the use of the *Gaussian Kernel* as distance measure and PCA with 270 components.

The rest of this report is organized as follows. Various methods have been explained in Section 4.2 to achieve the results. In Section 4.3, we have shown the results which we have achieved after applying PCA to input features vectors and then applying *Spectral Clustering* on the features vector and showing the accuracy of various types of *Spectral Clustering*. At last, we will give some concluding remarks in Section 4.4.

<u>4.2</u> <u>Materials</u> <u>And</u> <u>Methods</u>

The objective of the text document clustering is to group the text documents into clusters based on the similarity of the content available in a particular document. When we have the document, the first thing we have to do Document preprocessing. It is the significant phase to represent the documents for efficient document clustering . For doing text preprocessing we have to apply several techniques such as *Tokenization*, stop word removal and *Stemmization*. *Tokenization* is defined as the process where splitting a stream of text content into words, terms, symbols or certain other expressive features called tokens takes place. The list of tokens goes into an input for further processing which includes parsing or text mining. There are many words in the documents which occur frequently in the text, but they are basically meaningless words as they are used to connect the words well organized to form a sentence. These words are called stop words which does not denote the content or context of text documents. *Stemming* is defined as a process where reduction of a word to its word stem takes place that affixes to suffixes and prefixes or to the roots of words called as a lemma.

Now the input text has been preprocessed but it is still in the text format, but the algorithm accepts only vectors, so we have to convert the text into vectors so that we can feed it into the algorithm. To convert the text into vectors there are two representations which exist in the literature. They are:

- **Word-count vector**
- **TF-IDF vector**

The *Word-count* vector is the simplest form of text representation in numbers. Like the term itself, we can represent a sentence as a bag of words. It is a way to represent the text where arbitrary length text is converted into fixed-length vectors by counting the frequency of each word, hence the whole process is often referred to as vectorization. Let's understand it with a demo example. Suppose we wanted to convert the following sentences into vector:

- **the cat sat**
- **the cat sat in the hat**
- **the cat with the hat**

In the above 3 text only words: the, cat, sat, in, the, hat and with are present. In order to convert texts to vector, all we have to do is count how many times each word appears:

Document	the	cat	sat	in	hat	with
the cat sat	1	1	1	0	0	0
the cat sat in the hat	2	1	1	1	1	0
the cat with the hat	2	1	0	0	1	1

Fig 4.2 Word-Count Representation of above sentences

Now we have length-6 vector for each sentences

- **the cat sat with vector form [1,1,1,0,0,0]**
- **the cat sat in the hat with vector form [2,1,1,1,0]**
- **the cat with the hat with vector form [2,1,0,0,1,1]**

TF-IDF stands for *Term Frequency-Inverse Document Frequency*. It is an approach to quantify the value of a word in text, we calculate a weight and assign it to a word which will importize the value of words in the text and corpus. This approach is extensively used in *Information Retrieval* and *Text Mining. Term Frequency* increases the weight of the terms whose occurrence is more in the text. Hence its definition is as follows:

`tf(t,d) = F(t,d)`

where `F(t,d)` denotes frequency of term *t* in document

But practically, it looks not very likely that fifty occurrences of a term in a text actually carry fifty times the importance of a single occurrence. So, in order to make it pragmatic, we will scale Term Frequency in logarithmic terms so that as the frequency increases exponentially, we will be increasing the terms weights in an additive way.

`tf(t,d) = log(F(t,d))`

Inverse Document Frequency diminishes the weight of the terms that occur in all the documents of the corpus and similarly increases the weight of the terms whose frequency in rare documents across the corpus. The infrequent keywords get special treatment and stop words get punished.

IDF is defined as:

$$\mathrm{idf}(t,D) = \log(N/N_t \in d)$$

Here, N is the total number of files in the corpus D and $N_t \in d$ is the count of files in where the term t is present. As of now, we agree upon the fact that TF depends on individual documents as it is an intra-document factor and IDF is constant for a corpus as it is a per corpus factor. Finally, We calculate TF-IDF as:

`tf-idf(t,d,D) = tf(t,d) . idf(t,D)`

After converting text into vectors we can feed them into the algorithms for doing clustering. Now comes the part where we have a very high dimensional vector which if fed into the algorithm will take a lot of time for doing the task. So we applied *Dimensionality Reduction* techniques such as *Principle Component Analysis*(PCA) to reduce the dimension from very high numbers to a low value.

PCA is a technique for *feature extraction*, where it mixes our input variables in such a way that we can remove some variables which are not that much important whereas keeping the valuable information which are important variables hence still taking away the most important parts of all the variables. It is an unsupervised learning class of statistical techniques used to explain data in high dimensions using a smaller number of variables called the principal components. All the variables after being applied PCA are all independent of each other which is an added advantage.

We consider this as an advantage as considering the assumptions of the models which are linear they require the independent variables to be independent of one another. We applied PCA on the input text and reduced the features for visualization. The dimensions were reduced from 1000 to 2 as we can plot only in 2D. We used *Genism library* for plotting the *Word Embeddings* of the input text which clusters the similar nearer to each other where unsimilar words of the text are far away in the *Embedding* space.

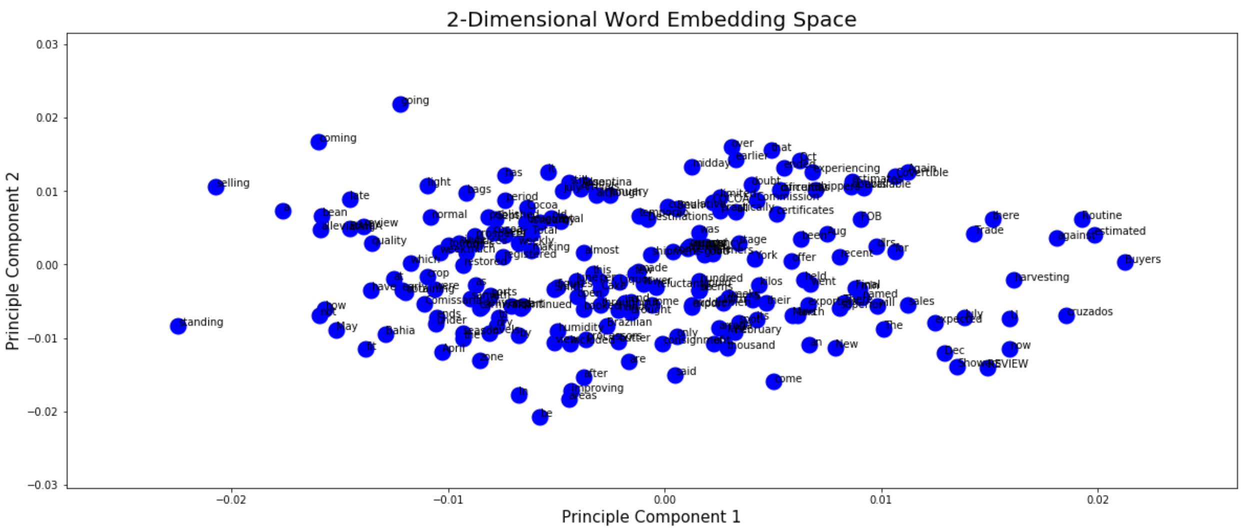

Fig 4.3 Visualization of Input Text in Embedding Space

For fitting the input features vector into the algorithm we tried various values of components such as 2, 50, 100, 150, 250, 270, 300, 500, 750, 900 while applying PCA to input features vectors. The best accuracy is achieved when the features vector has 270 features.

Now we have features vectors with reduced dimension, our vector is ready to feed into *Spectral Clustering* Algorithm. *Spectral Document Clustering* has developed in recent times as a widespread clustering technique, which motivated the emerging criterion functions and the developing algorithm to produce further accurate clusters . It uses the *Eigen Vectors* of the graph matrices which are derived from the documents. This algorithm is based on the concept of the weighted undirected graph. It demonstrates the collection of documents, i.e. document corpus $D = \{d1,d2,……, dn\}$. as an undirected graph $G(Vs,Es,Ma)$ where V_s is a Vertex Set, E_s denotes the Edge Set and M_a denotes the Graph Affinity Matrix. Each vertex $V_i \in V_s$ signifies the ith document and edge $(i, j) \in E_s$ is allocated to an affinity score. This algorithm consists of the following steps. First we have to construct a *Similarity graph* where there are numerous ways to transform a set of documents *{d1,d2,…, dn}* from document collections with the pair wise similarity ps_{ij} or distance wise similarity ds_{ij}, into a graph. Then we have to construct a *Graph Laplacian Matrix*. It is the most important part of spectral clustering. If there is no unique resolution then the matrix is precisely called as *Graph Laplacian*. The frequently used graph Laplacian types are as follows: *Un-normalized Graph Laplacian*, *Normalized Graph Laplacian*, *Normalized Graph Laplacian related to Random Walks.*After computing the *Laplacian matrix* we apply *Particle Swarm Optimization* to the *Eigen Vectors* computed from *Laplacian Matrix*.

Particle Swarm Optimization is a global stochastic optimization technique for incessant methods. The K-means algorithm is suitable for the initial clustering conditions, which can cause this algorithm to converge upon suboptimal solutions. But, the PSO algorithm is less sensitive for the initial conditions because of its population based nature. Hence, the PSO is more likely to find the near optimal solution. The ultimate purpose of this algorithm is to have whole particles come across the optima in a high dimensional data volume. The location of the particle in the high dimensional problem space, epitomize to explain a particular problem. While a particle travels from one location to a new location, another solution will be generated. These solutions are assessed by a fitness function which provides the optimal solution. Each particle will recall its recent coordinates and its velocity which shows the particle movement speed along with the dimensions of a problem space, from which the finest fitness value is established. The best value is coupled with its neighbor's best value, impacts the movement of every particle over the problem space.

For analysing how our algorithm has performed we calculated *Adjusted Rand Index*. The *Adjusted Rand index* in data clustering, is a measure of the similarity between two data clusterings. A form of the *Rand index* may be defined that is adjusted for the chance grouping of elements, this is the *Adjusted Rand Index*. Considering from a mathematical side, *Rand index* is related to accuracy, but is applicable even when class labels are not used. The *Adjusted Rand index* is the corrected-for-chance version of the *Rand index*. Such a correction for chance establishes a baseline by using the expected similarity of all pair-wise comparisons between clusterings specified by a random model.

```
ARI = (RI - Expected_RI) / (max(RI) - Expected_RI)
```

The whole above process is summarized in the Fig 4.4

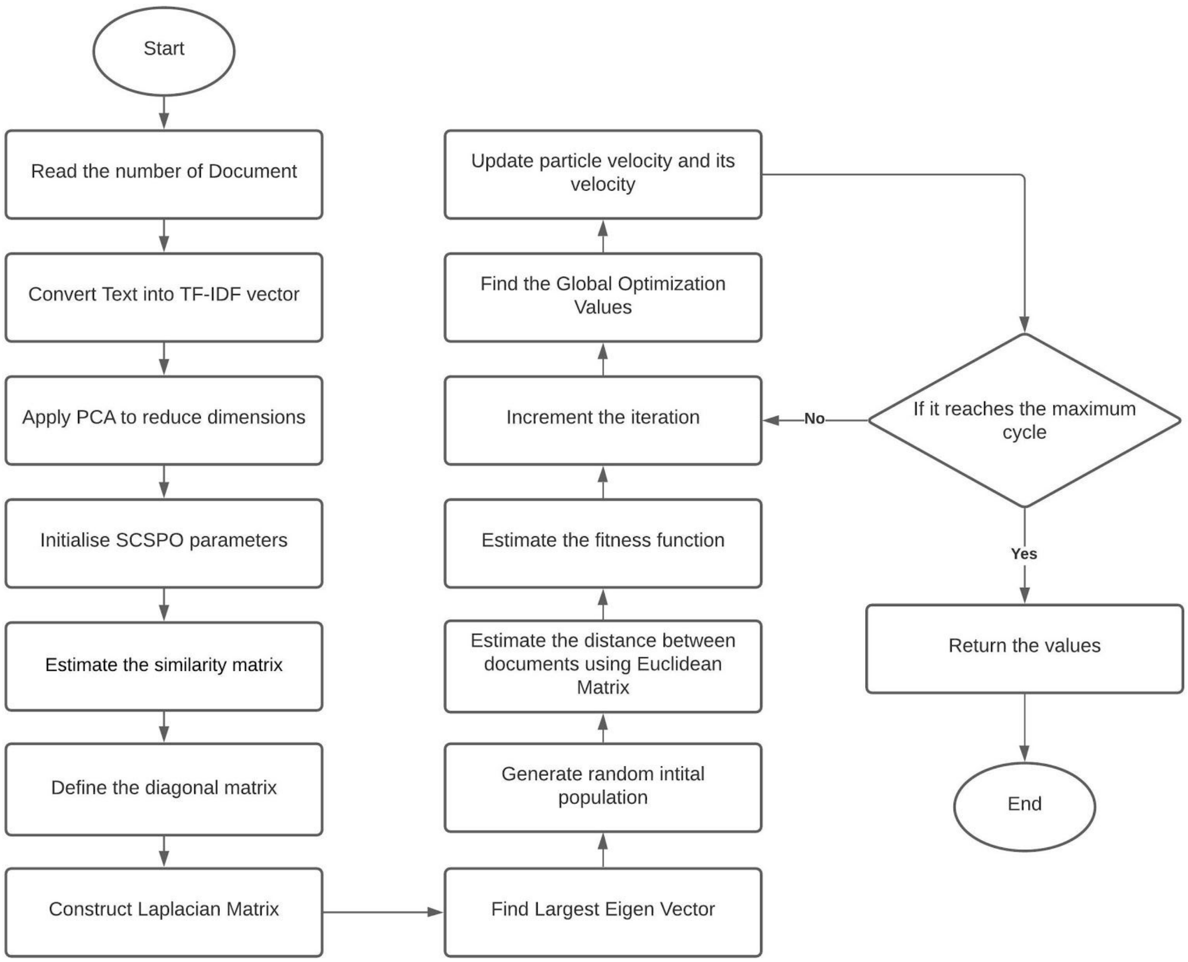

Fig 4.4 Flow chart for Text Document Clustering using while using PCA as Dimensionality Reduction and Spectral Clustering for clustering with Particle Swarm Optimization

4.3 Results & Discussion

Document clustering is a significant problem in the area of text mining and retrieving information. This is an open problem on which contributions are being added from time to time.

Spectral Clustering with *Particle Swarm Optimization* (SCPSO) is a beneficial method as compared to the existing models like *K-means*.

The figure given below shows the adjusted rand index for different clustering models we applied.

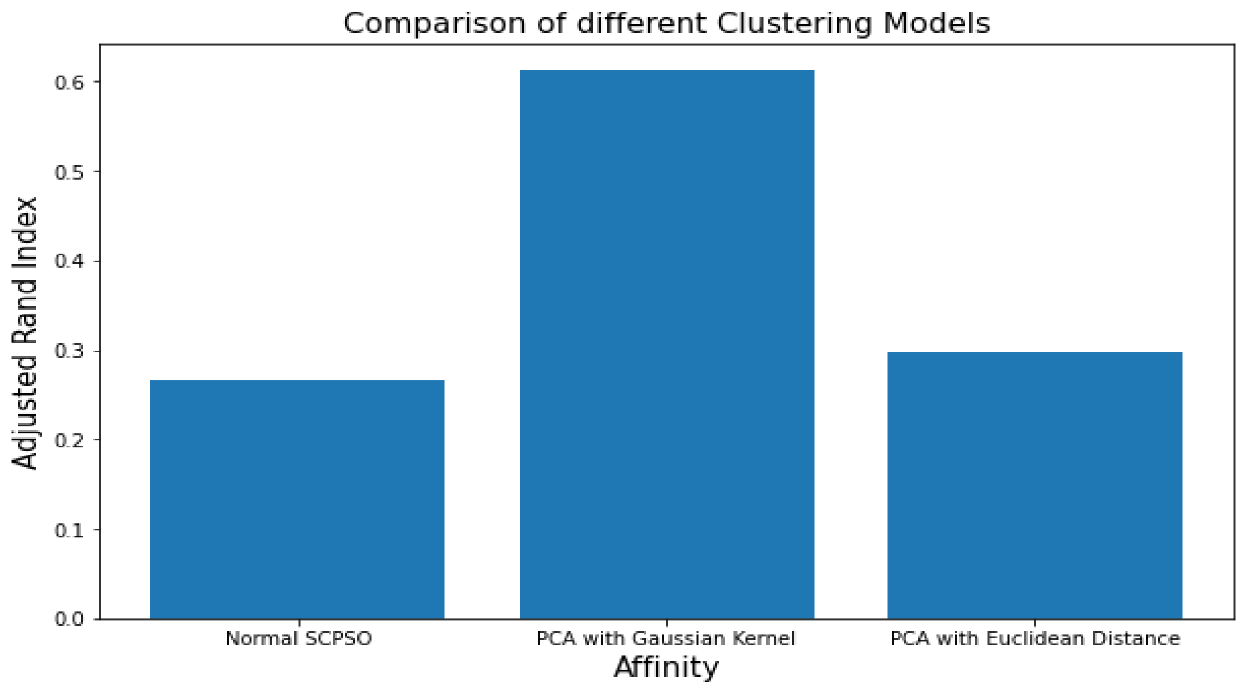

Fig 4.5 Comparing various Clustering Models

For experimentation, the clustering is being performed on the *Reuters* dataset which consists of text documents where each document belongs to some categories. This document corpus contains 90 categories with a total number of 21,578 documents. For the dataset, we applied preprocessing technique which is explained in the above section. For comparing various models, we have used the *Adjusted Rand Index* which gives the idea of how significant is a particular clustering model. A form of the *Rand index* may be defined that is adjusted for the chance grouping of elements; this is the *Adjusted Rand index*. Considering from a mathematical side, the Rand index is related to accuracy but is applicable even when class labels are not used. The *Adjusted Rand index* is the corrected-for-chance version of the Rand index. Such a correction for chance establishes a baseline by using the expected similarity of all pairwise comparisons between clusterings specified by a random model.

```
ARI = (RI - Expected_RI) / (max(RI) - Expected_RI)
```

We have also applied *Principal Component Analysis* (PCA) which is used for reducing the dimensionality while preserving as much information possible. Example - If the dataset has 100 attributes, applying PCA we can get ten features which may maintain over 99% of the information; thus the information is almost entirely retained but reducing dimensionality enables training the model better and faster. PCA has been applied on the TF-IDF vector.

Table 1
Performance measures comparison

Clustering Model	Adjusted Rand Index(ARI)
Normal SCPSO	0.2831796062227219
PCA applied SCPSO with distance metric as Euclidean distance	0.2474655533222179
PCA applied SCPSO with distance metric as Gaussian Kernel	0.6120316575561298

Table 1 reveals the comparison of ARI values on *Reuters* datasets after applying PCA to input features vectors and without applying PCA i.e., existing algorithms. This test shows that the applying PCA on features vectors and then applying SCPSO with distance metric as *Gaussian Kernel* algorithm outperforms the others in terms of ARI. The effectiveness of the proposed algorithm is slightly increased when compared to the PSO clustering algorithm.

4.4 Conclusion

In the research work, we have applied efforts to make the computation faster by applying Dimensionality *Reduction* techniques such as PCA to reduce the dimensions of features vectors while achieving the same level of accuracy as before. Above results shows that the best outcome is shown by the PCA with *Gaussian Kernel* followed by PCA with *Euclidean Distance* and finally by normal SCPSO. The ideas of our experiments proved to be useful as they showed improved results as compared to the typical SCPSO algorithm, the accuracy is still a bit lower as compared to the results in the research paper. We know that the problems that can be solved by *K-means* can be solved by *Spectral Clustering* but not the other way around, also applying PCA is very helpful as it is drastically reducing the dimensions of features vector and making the computation faster while keeping the accuracy as high as earlier. Hence we can conclude from the results, observations and the working of our algorithm that the additional ideas used, proved to be useful for the problem that we were aiming to solve. In future, experiments can include the use of parallel processing for improvement in the training period. *Hybrid clustering* can be applied to increase the robustness of the algorithm. Instead of PCA, other *Dimensionality Reduction* techniques such as *Linear Discriminant Analysis* (LDA) can be applied, and other ideas like transforming the data differently altogether can also be done. Distinct intentions can be introduced to attain the accomplished results of text document clustering. In addition to the above suggestions, changes can be applied to the spectral and optimization algorithms.

4.5 List of References

REFERENCES

1. R. Janani , Dr. S. Vijayarani, Text document clustering using Spectral Clustering algorithm with Particle Swarm Optimization, (2019)
2. Y. Li, S.M. Chung, J.D. Holt, Text document clustering based on frequent word meaning sequence, Data and Knowledge Engineering, 64 (1) (2008), pp. 381-404
3. J.D. Holt, S.M. Chung, Y. Li, Usage of mined word associations for text retrieval, in Proc. of IEEE Int'l Conf. on Tools with Artificial Intelligence (ICTAI-2007), vol. 2, 2007, pp. 45–49
4. Y. Li, C. Luo, S.M. Chung, Text clustering with feature selection by using statistical data, IEEE Transactions on Knowledge and Data Engineering, 20 (5) (2008), pp. 641-652
5. C.J. van Rijsbergen, Information Retrieval, (second ed.), Buttersworth, London (1979)
6. C. Buckley, A.F. Lewitt, Optimization of inverted vector searches, in Proc. of ACM SIGIR Conf. on Research and Development in Information Retrieval, 1985, pp. 97–110
7. Wei Xu, Xin Lui, Yibong Gong Document Clustering based on non-negative matrix factorization, (2003)
8. R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley, New York, NY, USA (2001)
9. A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, USA (1988)
10. A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: T.G. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, vol. 14, MIT Press, Cambridge, MA, USA, 2002, pp. 849–856.
11. J. Shi, J. Malik, Normalized cuts and image segmentation, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, 17–19 June 1997, pp. 731–737.
12. J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell., 22 (8) (2000), pp. 888-905
13. Y. Weiss, Segmentation using eigenvectors: a unifying view, in: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, Kerkyra, Greece, 20–27 September 1999, pp. 975–982.

14. B. Fischer, J.M. Buhmann, Path-based clustering for grouping of smooth curves and texture segmentation, IEEE Trans. Pattern Anal. Mach. Intell., 25 (4) (2003), pp. 513-518
15. B. Fischer, V. Roth, J.M. Buhmann, Clustering with the connectivity kernel, in: S. Thrun, L. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems, vol. 16, MIT Press, Cambridge, MA, USA, 2004.
16. B. Fischer, T. Zöller, J.M. Buhmann, Path based pairwise data clustering with application to texture segmentation, in: M.A.T. Figueiredo, J. Zerubia, A.K. Jain (Eds.), Proceedings of the Third International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, Sophia Antipolis, France, 3–5 September 2001, pp. 235–250.
17. A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs (1988)
18. D.R. Cutting, D.R. Karger, J.O. Pedersen, J.W. Tukey, Scatter/gather: a cluster-based approach to browsing large document collections, in: Proc. of ACM SIGIR Conf. on Research and Development in Information Retrieval, 1992, pp. 318–329.
19. L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons (1990)
20. B. Larsen, C. Aone, Fast and effective text mining using linear-time document clustering, in: Proc. of ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, 1999, pp. 16–22.
21. C. Ordonez, E. Omiecinski, Efficient disk-based *k*-means clustering for relational databases, IEEE Transactions on Knowledge and Data Engineering, 16 (8) (2004), pp. 909-921