Data Preprocessing for Improving Cluster Analysis and Its Application to Short Text Data

Abstract

Data Preprocessing for Improving Cluster Analysis and Its Application to Short Text Data

Graduate School of
Natural Science & Technology
Kanazawa University

Division of Electrical Engineering
and Computer Science

Student ID No.: 1223112012
Name: Tran Vu Anh
Chief advisor: Professor Kenji Satou
Date of Submission: July 3rd, 2015

CHAPTER I Dissertation introduction

This chapter is to briefly introduce the content and distribution of this dissertation.

CHAPTER II Clustering and data preprocessing for clustering

This chapter briefly presents the background of clustering and its challenges. We then introduce data preprocessing methods in order to deal with challenges in clustering.

2.1 Clustering

As introduced above, clustering task organizes data objects into groups whose members are similar in some way. A cluster is therefore a collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters. Clustering is applied in various fields, e.g., marketing (categorizes the customer), biology (classify the gene expression data), geography (identify the similar zones appropriate for exploitation) and so on.

Clustering has more than 50 years of development. Many clustering algorithms were proposed with different schemas and concepts [1]. Even though with a long history of research and development, there are still several challenges existed for clustering, i.e., the number of clusters, high dimensionality, noise and outlier. Such problems can impact the quality of cluster analysis. However, if the data have been preprocessed appropriately, for example, clusters are well-separated, dense and have no noise, the performance of the clustering algorithms may improve. Data preprocessing is often used to perform such tasks in order to improve the quality of data, and hence improve the performance of clustering.

2.2 Data preprocessing

Real world data usually contain noises and outliers, are high dimensional, hence, strongly impact the performance of clustering. To deal with such problems, data preprocessing methods are employed to improve quality of data and therefore, improve the performance of clustering. The popular tasks of data preprocessing methods in clustering are feature reduction (i.e., PCA [2]), feature selection [3], noise and outlier removal [4].

CHAPTER III Data preprocessing algorithm D-IMPACT

In this chapter, we describe the data preprocessing algorithm D-IMPACT [5] based on concepts underlying the clustering algorithm IMPACT [6]. We aim to improve the accuracy and flexibility of the movement of data points in the IMPACT algorithm by applying the concept of density to various affinity functions. These improvements will be described in the subsequent subsections.

3.1 Gravity-based data preprocessing algorithm

Recent studies have focused on new categories of clustering algorithms which prioritize the application of data preprocessing. SHRINK, a data shrinking process, moves data points along the gradient of the density, generating condensed and widely separated clusters [7]. In CLUES [8], each data point is transformed such that it moves a specific distance toward the center of a cluster. These two shrinking algorithms share the following limitations:

- The process of shifting toward the median of neighbors can easily fracture the cluster.

- The direction of the movement vector is not appropriate in specific cases. For example, if the clusters are adjacent and differ highly in density, the median of the neighbors is likely to be located on another cluster.

We introduce a clustering algorithm based on the simulation of gravity system: moving data points under effect of attractive-force like values to form dense regions that can be easily identified as clusters. The data points movement in IMPACT algorithm can avoid the addressed problems. The next section will explain the algorithm in detailed.

3.2 Clustering algorithm IMPACT

3.2.1 IMPACT algorithm

The IMPACT algorithm is based on the idea of gradually moving all objects closer to similar objects according to the attraction between them until the dataset becomes self-partitioned. The algorithm has two phases. The first phase is for normalizing and denoising the input dataset. In the second phase, IMPACT iteratively moves the data points and identifies clusters.

To evaluate the performance of IMPACT algorithm, we tested it on datasets with different characteristics. The results will be presented in the next section.

3.2.2 Experiment result

In this section, we evaluate the performance of IMPACT and demonstrate its effectiveness for different types of data distributions. The experiment results of IMPACT show that the algorithm is effective in identifying clusters with arbitrary shapes, density, and orientation, without affected by the number of clusters and noise. In addition, IMPACT algorithm is not parameter sensitive. The IMPACT algorithm not only works effectively with two-dimensional datasets but also produces accurate results when dealing with practical datasets. The clustering results on UCI datasets and text dataset shows IMPACT can identify the correct number of clusters and archive high performance of clustering. However, there are several limitations existed for IMPACT algorithm:

- The datasets are not completely denoised.
- In several cases, small parts of clusters are merged.
- IMPACT takes long processing time to cluster the data.

In this study, we propose a data preprocessing algorithm named D-IMPACT (Density-IMPACT) to overcome the limitation of gravity-based preprocessing algorithms by utilizing the idea of IMPACT algorithm and the concept of density [9]. An advantage of our algorithm is its flexibility in relation to various types of data; it is possible to select an affinity function suitable for the characteristic of the dataset. This flexibility improves the quality of cluster analysis even if the dataset is high-dimensional and non-linearly distributed, or includes noisy samples.

3.3 Data preprocessing algorithm D-IMPACT

In this section, we describe the data preprocessing algorithm D-IMPACT based on concepts underlying the IMPACT algorithm. We aim to improve the accuracy and flexibility of the movement of data points in the IMPACT algorithm by applying the concept of density to various affinity functions. These improvements will be described in the subsequent subsections.

3.3.1 Movement of data points

The main difference between the data movement in D-IMPACT and IMPACT algorithms is that the movement of data points can be varied by the density functions, the attraction functions, and an *inertia* value. This helps D-IMPACT detects different types of clusters and avoid many

common clustering problems, i.e., inappropriate movement, border overlapping, and increases the efficient of computation.

3.3.2 D-IMPACT algorithm

D-IMPACT has two phases. The first phase detects noisy and outlier data points, and removes them. The second separates clusters by iteratively moving data points based on attraction and density functions. **Figure 3.1** shows the flow chart of the D-IMPACT algorithm.

3.3.2.1 Noisy points and outlier detection

First step is noise and outlier detection. An outlier is a data point significantly distant from the clusters. We refer to data points which are close to clusters but do not belong to them to as noisy points, or noise, in this manuscript. Both of these data point types are usually located in sparsely-scattered areas, that is, low-density regions. Hence, we can detect them based on density and the distance to clusters. Both outliers and noisy points are output and then removed from the dataset. When this phase is completed, the movement phase commences.

3.3.2.2 Moving data points

In this phase, the data points are iteratively moved until the stop criterion is met. The distances and the densities are calculated first, after which, we compute the components used to determine the movement vectors: attraction, affinity vector, and the *Inertia* value. The data points are then moved in order to shrink the clusters to increase their separation from one another. This process is repeated until the stop condition is satisfied. In D-IMPACT, we adopt various stop criteria as follows:

- Stop after a fixed number of iterations controlled by the parameter n_{iter}.
- Stop based on the average of the densities of all data points.
- Stop when the magnitudes of movement vectors have decreased significantly compared to the previous iteration.

When this phase is complete, the preprocessed dataset is output. The new dataset contains separated and shrunk clusters, with noise and outliers removed.

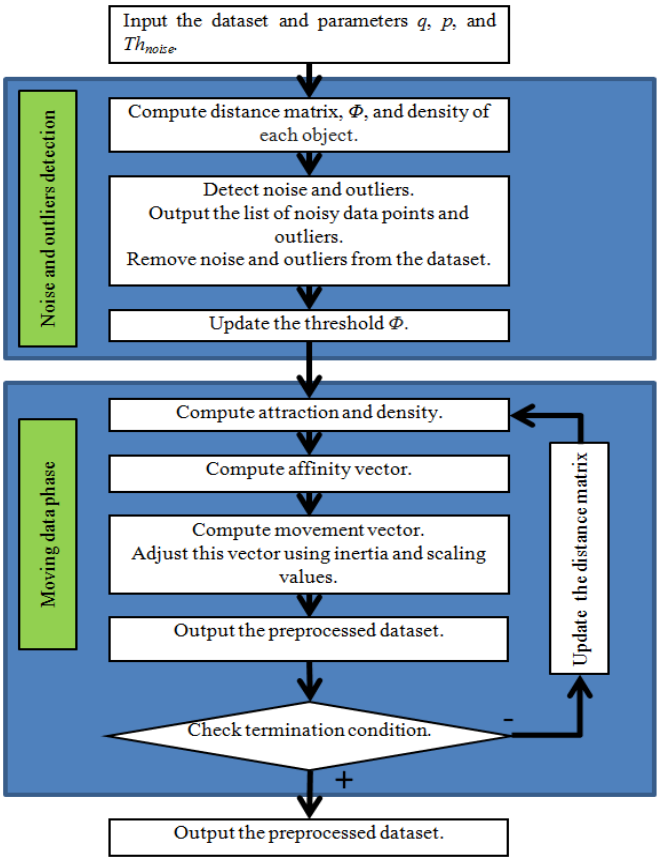


Figure 3.1 Outline of the D-IMPACT algorithm

3.3.2.3 Complexity

D-IMPACT is a computationally-efficient algorithm. The overall complexity of D-IMPACT is $O(m^2n)$. We measured the real processing

time of D-IMPACT on 10 synthetic datasets. For each dataset, the data points were randomly located (uniformly distributed). The results in **Figure 3.2** show the advantage in speed of D-IMPACT in relation to CLUES.

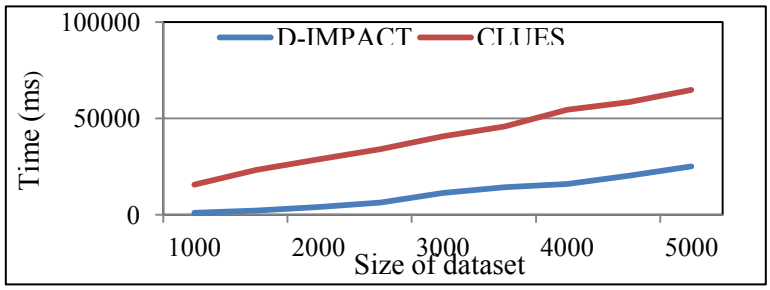


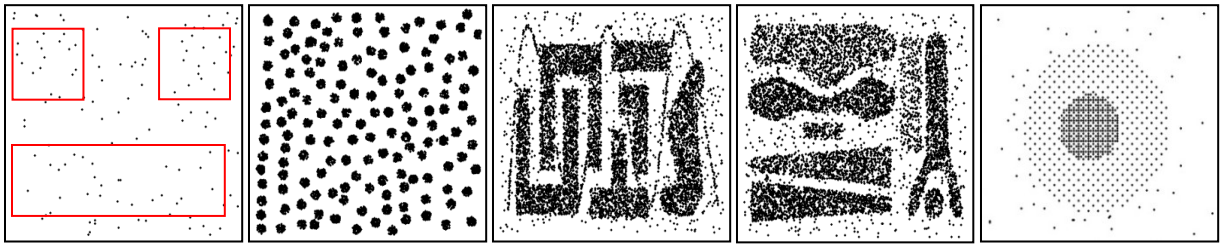
Figure 3.2 Processing times of D-IMPACT and CLUES on test datasets.

3.4 Experiment result

In this section, we compare the effectiveness of D-IMPACT and the shrinking function of CLUES (in short, CLUES) on different types of datasets.

3.4.1 Datasets and method

3.4.1.1 Two-dimensional datasets



a) DM130 b) MultiCL c) t4.8k d) t8.8k e) Planet

Figure 3.3 Visualizations of 2D datasets for validating D-IMPACT algorithm.

To validate the effectiveness of D-IMPACT, we used different types of datasets: two dimensional (2D) datasets taken from the Machine Learning Repository (UCI), and a microarray dataset. **Figure 3.3** shows the 2D datasets used.

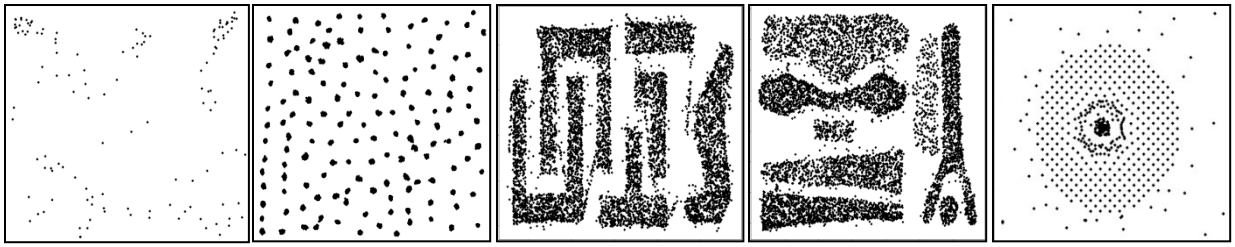
3.4.1.2 Practical datasets

The practical datasets are more complex than the 2D datasets, i.e., the high dimensionality can greatly impact the usefulness of the distance function. We used the Wine, Iris, Water-treatment plant (WTP), and Lung-cancer (LC) datasets from UCI [10], as well as the dataset GSE9712 from the Gene Expression Omnibus [11] to test D-IMPACT and CLUES on high-dimensional datasets.

3.4.1.3 Validating methods

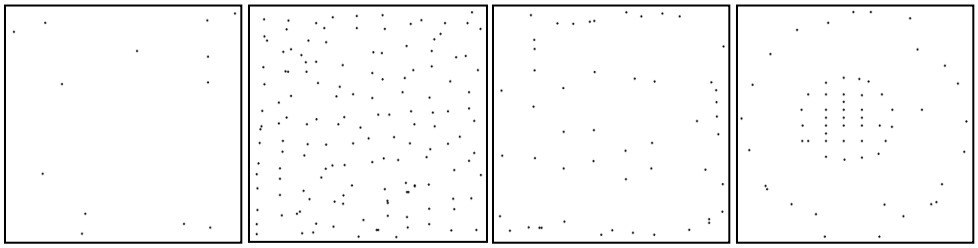
We compare the results of D-IMPACT with CLUES implemented in R [12]. 2D plots are use to visualize the effect of both algorithms in case of 2D datasets. We used two evaluation measures, the Rand Index and adjusted Rand Index (aRI) [13] to evaluate the clustering result from Hierarchical agglomerative clustering (HAC) on practical datasets preprocessed by D-IMPACT and CLUES.

3.4.2 Experimental results of 2D datasets



a) DM130 b) MultiCL c) t4.8k d) t8.8k e) Planet

Figure 3.4 Visualizations of 2D datasets after preprocessed by D-IMPACT.



a) DM130 b) MultiCL c) t4.8k d) Planet

Figure 3.5 Visualizations of 2D datasets after preprocessed by CLUES.

The results of D-IMPACT are shown in **Figure 3.4**. From these results, we can see that IMPACT algorithm can separate clusters, remove noise while retain the global structure of clusters. In contrast, CLUES incorrectly merge and fracture clusters (**Figure 3.5**).

3.4.3 Experimental results of practical datasets

We used HAC to cluster the original and preprocessed Iris and Wine datasets, and then validated the clustering results with aRI. A higher Rand Index score indicates a better clustering result. D-IMPACT outperforms PCA and CLUES in case of Iris, Wine, and GSE9712 datasets (**Table 3.1**). The results produced by D-IMPACT can improve the performance of clustering most among three methods, and more stable. In case of WTP dataset, CLUES incorrectly merger all outliers to other clusters, while D-IMPACT produce a dataset which 8 out of 9 minor clusters (outliers) can be detected easily. D-IMPACT also separates outliers from clusters better than CEE [14] in case of Lung cancer dataset.

Table 3.1 Index scores of clustering results using HAC on the original and preprocessed datasets of IRIS and Wine. The best scores are in bold.

Dataset	Preprocessing algorithm		
	None	CLUES	D-IMPACT
Iris	0.759	0.732	**0.835**
Wine	0.810	**0.899**	0.884
GSE9712	0.330	0.139	**0.330**

3.5 Conclusion

In this study, we proposed a data preprocessing algorithm named D-IMPACT inspired by the IMPACT clustering algorithm. D-IMPACT moves data points based on attraction and density to create a new dataset where noisy points and outliers are removed, and clusters are separated. The experimental results with different types of datasets clearly demonstrated the effectiveness of D-IMPACT. The clustering algorithm employed on the datasets preprocessed by D-IMPACT detected clusters and outliers more accurately.

CHAPTER IV Data preprocessing algorithm SCF

In this chapter, we describe the data preprocessing algorithm SCF which aims to reduce the number of dimensions without losing the semantic information stored in each feature. The new space produced by SCF will contains the semantic similarity between the keywords of each documents and the concept underlying the corpus.

4.1 Clustering algorithms and data preprocessing methods for text clustering

4.1.1 Text clustering

With the rapid growth of information exchange, a large number of documents are created in everyday, such as emails, news, forum post, social network posts, etc. To help people deal with document overload, many systems apply clustering to help people manage, organize, and organize text data more effective.

4.1.2 Challenges of text clustering in short text data

There are still several challenges existed for clustering, i.e., the number of clusters, high dimensionality. In case of text data, critical problems for text clustering are:

- High dimensionality and sparseness
- Ignoring the semantic relationship between the words.
- Comparing to other kinds of documents, i.e., article, official document, book, the short text is less strict in grammar and may contains a lot of misspellings.
- Short texts may contain pattern repeated in many texts but not contribute to the content of the text.

One of solutions for problems above is employing data preprocessing before doing clustering in order to improve the quality of clustering results. Next section will present the basic of data processing and introduce briefly several data preprocessing algorithms.

4.1.3 Data preprocessing for text clustering

We briefly introduce several popular data preprocessing techniques for text clustering. Latent Semantic Indexing (LSI) [15] applies PCA technique to project word frequency matrix into "latent" semantic space, which can reveal underlying topics in the documents. However, similar to PCA, the new dimensions produced by LSI are just a linear transformation from original word frequency matrix, so they may not correspond to meaningful topic underlying the documents. In addition, these methods are heavy computation, so they are inefficient to be employed on large datasets. Recently, many researches utilized WordNet [16], a thesaurus for English, to do clustering with considering the semantic relationship between words. However, these researches are quite complex, heavy computation, and cannot completely solve the problem of high dimensionality. Next, we introduce WordNet, which plays an important role in this research.

4.2 WordNet and semantic similarity

WordNet is a large lexical database of English and then extended to other languages. Nouns, verbs, adjectives and adverbs are organized into sets of cognitive synonyms (synsets), which are linked by semantic relationship. This network makes WordNet becomes a useful tool for computational linguistics and natural language processing.

4.2.1 WordNet structure

Information in WordNet is organized around sets of cognitive synonyms called synsets and the relationship between them. WordNet 3.0 contains 155287 unique strings (words) organized into 117659 synsets, which has 206941 relationship links (word-sense pairs) between them. Synsets are linked based on the sense relationship between them, providing a hierarchical structure for computing semantic similarity.

4.2.2 Semantic similarity

A number of semantic similarity computation methods are summarized in [17]. In this research, we simply apply path length measure implemented in package WordNet on Python to calculate the semantic similarity [18]. Next, to calculate the semantic similarity between words, we employ first *k* approach to calculate the semantic similarity between two words. The idea is using only first *k*-synsets for each word to calculate the semantic similarity between them. We denote synset(t, k) = $\{s_1, \ldots, s_k\}$ as the set of first *k*-synset for word t, and $ss(x,y)$ as the semantic similarity between two words x and y; the sense-relatedness between two words t_i, t_j is computed as:

$$term\_ss(t_i, t_j, k) = \max(ss(x, y) | x \in \text{synset}(t_i, k), y \in \text{synset}(t_j, k))$$

Based on the formula, the semantic similarity between two words t_i, t_j is the maximum value of semantic similarity between all pairs of synsets belonging to two set of synsets synset(t_i, k) and synset(t_j, k).

4.3 Data preprocessing algorithm SCF

This chapter is to present an algorithm to reduce the number of dimensions by doing semantic-based features clustering and then create semantic conceptual features in order to improve the quality of clustering. In this section, two phases of SCF algorithm will be explained step by step.

4.3.1 Phase 1: Word pruning and clustering.

In this phase, unnecessary words are removed and replace semantically related words by a representative word in order to reduce the number of dimensions. Firstly, we discards all words not include in WordNet in this step, which mostly are misspellings and jargons. Next, extreme-high document frequency words are automatically detected by clustering and then discarded.

Next, we do clustering on the remaining words based on the semantic similarity between them. From the clustering result, clusters can reveal the groups of semantically related words and the centroid of the clusters is considered as the representative words for all the words belonging to that cluster. Then, we create representative word frequency matrix (RWFM). The feature space of this matrix is representative words (centroids of the clusters) identified in the previous step and can present the frequency and the semantic relationship of all the words belonging to the group of semantically related words..

4.3.2 Phase 2: keywords and Semantic related Conceptual Feature (SCF) matrix construction

It is not necessary to use all the words or representative words to describe the main topics in the document. Actually, the main topics can be identified via several words, for examples, keywords in a scientific article, or tags in a news article. In this research, we define such words are keywords. To automatically identify the keywords, we firstly apply TF-IDF [19] (term frequency–inverse document frequency) to weight the representative words frequency matrix RWFM. Then, we apply clustering on the term frequency of each document to identify keywords. Base on the distribution of keywords, unnecessary keywords will be discarded. After identifying keywords and selecting important representative features, we discover the concepts (main topics)

underlying the documents by doing clustering on the covariance matrix of important representative words to find the groups of high co-occurrence important representative words, which are the concepts underlying the corpus.

Finally, the Semantic related Conceptual Feature (SCF) matrix will be constructed based on the keywords of each document and the concepts underlying the corpus. The final matrix *SCF* only presents the semantic similarity between keywords of each document (which are representative word that important to present the content underlying the document and corpus) and the concepts underlying the corpus. Therefore, SCF should improve the performance of clustering.

4.4 Experiment result

In this chapter, we evaluate the performance of the proposed algorithm SCF and compare it with other methods.

4.4.1 Datasets and text processing

In our study, we used two short text corpuses to evaluate the proposed algorithm: Enron and 20 newsgroups corpus. These datasets are widely used for experiment in text mining research, such as text classification and text clustering. UC Berkeley Enron dataset [20] contains 1702 emails and classify into 8 classes. The second dataset, 20 newsgroups dataset [21], contains approximately 20000 newsgroups belonging to 20 different classes. **Table 4.1** summaries the characteristic of these two datasets. To do the text processing, we used NLTK package on Python [18] for tokenizing and doing POS-tagging. We then selected all nouns and verbs for creating term frequency matrix. We used the first *k* synsets approach described in section 4.2.2 to calculate the semantic similarity. The value *k* is set to 2 for the sake of word clustering.

TABLE 4.1 UC Berkeley Enron dataset and 20 newsgroups dataset

Name	No. documents	No. classes	Size of clusters	No. nouns	No. verbs
Enron	1546	6	727, 36, 92, 474, 74,143	5489	2212
20 newsgroups	19918	20	~500 for each	50399	12255

4.4.2 Experiment result

We employed SCF on UC Berkeley Enron and 20 newsgroups datasets. For doing word clustering, the value of the threshold *th* is set to 0.5 (hypernym/hyponym relationship). To find the concepts underlying the corpus, the value of the threshold *th* is set to 0.7. The result of number reduction is showed in the **Table 4.2**. Both two methods word clustering

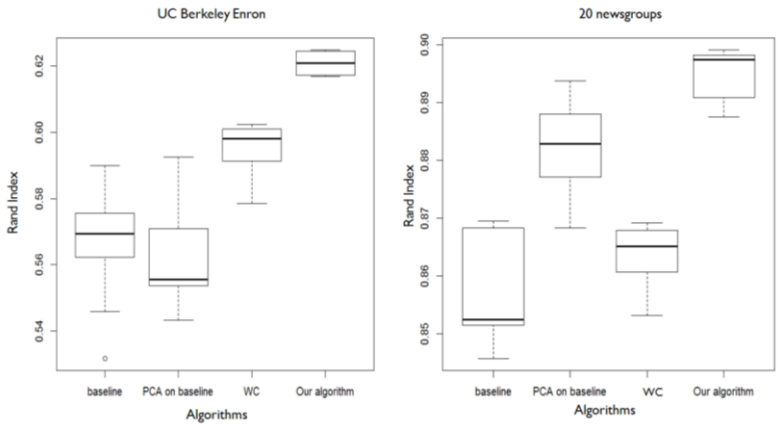


Figure 4.1 Comparisons of clustering performances on UC Berkeley Enron and 20 newsgroups datasets.

and SCF algorithm can greatly reduce the number of feature.

Table 4.2 Result of feature reduction by word clustering and SCF algorithm

Name	Words	Baseline	Phase 1	Phase 2	% reduction	
UC Berkeley Enron	Nouns	5489	1964	168	97.122	96.652
	Verbs	2212	748	90	95.931	
20 newsgroups	Nouns	50399	7006	592	98.825	98.722
	Verbs	12255	1613	209	98.294	

The results of clustering performed on the matrices produced by word clustering and SCF algorithm are showed in **Figure 4.1**. We employed *k*-means (and giving the correct number of clusters in both cases of UC Berkeley Enron and 20 newsgroups datasets) 10 times for each experiment on four matrices: the baseline (contains nouns and verbs after doing stopword removal), term frequency transformed by PCA (with the number of pc varied from 2 to 30), the RWFM (by word clustering), and SCF matrix. The clustering results are then evaluated by Rand Index. The results show that both word clustering can improve the performance of clustering compared to using the original term frequency matrix. In case of UC Berkeley Enron dataset, PCA degenerate the performance of clustering, because most of the topics in this dataset are highly related to a main theme: business. In case of 20 newsgroups dataset, PCA can improve the quality of clustering, however the improvement is lower than the result of SFC algorithm.

4.5 Conclusion

In this research, we proposed a data preprocessing algorithm named SCF to reduce the number of dimensions without losing the semantic information stored in each feature. The new space produced by SCF algorithm presents the semantic similarity between the keywords of each documents and the concept underlying the corpus, hence can present the content of the topics underlying the corpus clearer. The experiment results show that SCF algorithm can create a new space of a small number of features but can improve the performance of clustering result preformed on SCF matrix.

CHAPTER V Conclusion

In this literature, we introduced two data preprocessing methods named D-IMPACT and SCF. D-IMPACT algorithm focuses on removing noises/outliers and separating clusters based on moving data points. SCF algorithm focuses on feature reduction and improving quality of data by computing semantic similarity between keywords of each document and the concepts underlying corpus. The experiment results clearly show effectiveness of both D-IMPACT and SCF algorithm.

In the future, we can improve the algorithm D-IMPACT by employing new formulas to compute the density, attraction and vectors in data objects moving phase to improve its effectiveness. Similar to D-IMPACT, we would like to validate the effect of different semantic similarity functions to find the best measure for SCF algorithm.

Bibliography

[1] Berkhin, P. “Survey of clustering data mining techniques.” Technical report, Accrue Software, San Jose, CA, 2002.
[2] Abdi, H., Williams, L.J. "Principal component analysis." *Wiley Interdisciplinary Reviews: Computational Statistics*, **2(4)**, 2010, 433–459.
[3] Alelyani, S., Tang, J., and Liu, H. "Feature Selection for Clustering: A Review." *Data Clustering: Algorithms and Applications*, **29**, 2013.
[4] Hodge, V.J., Austin, J. "A survey of outlier detection methodologies." *Artificial Intelligence Review*, **22(2)**, 2004, 85-126.
[5] Tran, V.A., et al. “D-IMPACT: A Data Preprocessing Algorithm to Improve the Performance of Clustering.” *Journal of Software Engineering and Applications*, **7**, 2014, 639-654.
[6] Tran, V.A., et al. "IMPACT: A Novel Clustering Algorithm based on Attraction." *Journal of Computers*, **7(3)**, 2012, 653-665.
[7] Shi, Y., Song, Y., Zhang, A. “A shrinking-based clustering approach for multidimensional data.” *IEEE Trans. Knowl. Data Eng.*, **17(10)**, 2005, 1389 – 1403.
[8] Chang, F., Qiu, W., Zamar, R.H. ”CLUES: A Non-Parametric Clustering Method Based on Local Shrinking." *Computational Statistics & Data Analysis*, **52(1)**, 2007, 286-298.
[9] Ester, M., Kriegel, H.P., Sander, J., Xu, X. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.” *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, 1996, 226–231.
[10] The UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/datasets.
[11] Radioresistant and radiosensitive tumors and cell lines. http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE9712.
[12] Chang, F., Qiu, W., Zamar, R.H., Lazarus, R., Wang, X. “clues: An R Package for Nonparametric Clustering Based on Local Shrinking.” *Journal of Statistical Software*, **33(4)**, 2010, 1-16.
[13] Hubert, L., Arabie, P. “Comparing Partitions.” *Journal of Classification*, **2(1)**, Springer, 1985, 193–218.
[14] Visakh, R., and Lakshmipathi, B. “Constraint based Cluster Ensemble to Detect Outliers in Medical Datasets,” *International Journal of Computer Applications*, **45(15)**, 2012, 9-15.
[15] Deerwester, S. “Improving Information Retrieval with Latent Semantic Indexing”. Proceedings of the 51st Annual Meeting of the American Society for Information Science **25**, 1988, 36–40.
[16] Miller, G.A. "WordNet: a lexical database for English." *Communications of the ACM*, **38(11)**, 1995, 39-41.
[17] Meng, L., Huang, R., and Gu, J. "A review of semantic similarity measures in wordnet." *International Journal of Hybrid Information Technology*, **6(1)**, 2013, 1-12.
[18] Perkins, J. *Python 3 Text Processing with NLTK 3 Cookbook*. Packt Publishing Ltd, 2014.
[19] Berger, A., and Lafferty, J. "Information retrieval as statistical translation." *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 1999, 222-229.
[20] U C Berkeley Enron Email Analysis. http://bailando.sims.berkeley.edu/enron_email.html
[21] 20 newsgroups. http://qwone.com/~jason/20Newsgroups/

平成２７年　７月２８日

学位論文審査報告書（甲）

１．学位論文題目（外国語の場合は和訳を付けること。）

Data Preprocessing for Improving Cluster Analysis and Its Application to Short Text Data（データの前処理によるクラスター分析の改善とショートテキストデータへの応用）

２．論文提出者　(1) 所　　属　　電子情報科学　専攻

(2) 氏（ふり）　名（がな）　Tran Vu Anh（ちゃん ぶー あん）

３．審査結果の要旨（600～650 字）

平成２７年７月２８日に第１回学位論文審査委員会を開催、同日に口頭発表、その後に第２回審査委員会を開催し、慎重審議の結果、以下の通り判定した。なお、口頭発表における質疑を最終試験に代えるものとした。

クラスタリングは教師無し学習の一種であり、データ解析の重要な手法として古くから研究されてきたが、適切なクラスタ数の決定方法や、特殊な分布を持つデータへの対応、高次元データの取り扱いなど、未解決の問題が多い。本研究では、クラスタリングの前に行うデータ前処理手法として、D-IMPACT および SCF を開発した。D-IMPACT はデータの密度に基づいてノイズや外れ値の除去を行った後、データ間の引力と密度に基づいて繰り返しデータを移動することにより、クラスタ間の分離を明確にし、クラスタリングの精度を向上することができた。一方、SCF は電子メールなどの短いテキストを対象とし、元のデータを意味的概念空間に変換することにより、テキスト中のキーワードや概念をより良く表現し、テキストクラスタリングの精度を向上することができた。

以上の研究成果は、データの前処理によるクラスタリングの精度向上に大きく貢献するものであり、本論文は博士（工学）に値するものと判定した。

４．審査結果　(1) 判　　定（いずれかに○印）　（合　格）・　不合格

(2) 授与学位　　博　士（工学）