HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

MASTER THESIS

Mr.
Adnan Siddique

Clustering of Financial Documents for structure Detection and Future Feature Learning

2019

Faculty of **Applied Computer Sciences and Biosciences**

MASTER THESIS

Clustering of Financial Documents for structure Detection and Future Feature Learning

Author:
Adnan Siddique

Study Programme:
Applied Mathematics in Digital Media

Seminar Group:
MA15w2-M

First Referee:
Prof. Dr. Kristan Schneider

Second Referee:
Mr. Alrik Messner

Mittweida, July 2019

Bibliographic Information

Siddique, Adnan: Clustering of Financial Documents for structure Detection and Future Feature Learning, 57 pages, 12 figures, Hochschule Mittweida, University of Applied Sciences, Faculty of Applied Computer Sciences and Biosciences

Master Thesis, 2019

Abstract

In today's market, the process of dealing with textual data for internal and external processes has become increasingly important and more complex for certain companies. In this context,the thesis aims to support the process of analysis of similarities among textual documents by analyzing relationships among them. The proposed analysis process includes discovering similarities among these financial documents as well as possible patterns. The proposal is based on the exploitation and extension of already existing approaches as well as on their combination with well-known clustering analysis techniques. Moreover, a software tool has been implemented for the evaluation of the proposed approach, and experimented on the EDGAR filings, on the basis of qualitative criteria.

Keywords: TFIDF (Term Frequency-Inverse Document Frequency), LDA (Latent Dirichlet Allocation), LSI (Latent Semantic Indexing), SVM (Support Vector Machine), PCA (Pronciapal Component Analysis)

I. Contents

II. List of Figures

III. List of Tables

1 Introduction

1.1 Background

In today's, the process of dealing with financial information has become increasingly important and more complex for companies. Financial information is quite tough and the accessible data on the market needs to satisfy needs of potential customers.

Document and text clustering is an unsupervised learning technique. Document clustering deals with the unsupervised partitioning of a document collection into meaningful groups based on their textual content, usually for topic categorization; i.e. an individual cluster contains documents on one topic whereas different clusters will contain documents on different topics. Unlike document classification - which is a supervised learning method that requires prior knowledge of document categories to train a classifier, document clustering is an unsupervised learning method that does not rely on prior categorization knowledge.

Natural Language Process (NLP) is a system to read heterogeneous information and gather feature vectors for required data. Firstly, in vectorization each text input represents with real vector representation that can represent the inner concept of text is easy to define. Secondly, for symbolic data similarity is often determined making use of special purpose similarity measures and distance functions, for a great deal depending on the application domain under consideration.

During the thesis different well known clustering algorithms were used to cluster the data into different groups, for example the corpus was classify into different topics on the basis of terms appear in text. To classify the data into these different categories will be similarity and distance between different features. If the similarity matrix of features will be quite same it will classify as same otherwise it will classify in conventional way. Similarly the topics were classified in different categories based on their semantic meaning as well.

To cope with non-vectored information two approaches exist. First, prepossessing approach, textual information will have transferred to real vector used for one standard kernel function. Secondly, kernel function map real vector in feature

space F.

1.2 Problem Statement

Financial Documents clustering is an unsupervised learning system for structure detection. Its detects feature and structure construction, both correct and incorrect, in freely written text in EDGAR filing system. It can also determine bond and not bond, given a text with enough identifying attributes.

Objects that achieve from documents belongs to bond and not bond cluster using different clustering algorithms. These similarities depend on semantics meanings and distance between vectors. Those vectors have shorter distance belongs to one and those have large distance belongs to different clusters.

1.3 Objective

The aim of the research is to establish the composition-property relationship of financial EDGAR documents in order to reduce the time taken for the approval of a financial document with the help of machine learning tools. More specifically if the changes that occur in the documents when there is a change in the agreement parameters can be efficiently predicted, then the probability of that documents is bond or not bond agreement can also be efficiently predicted. To do that the financial documents of different different companies that have been collected from EDGAR filings must be analyzed and machine learning algorithms must be applied to the documents in order to extract useful information.

Information extraction from files and make set of objects for data or vector representation of objects. Correct object gathering and vector representation we used a prepossessing technique.

Natural Language Process (NLP) is a system to read heterogeneous information and gather feature vectors for required data. Firstly, in victories each text input represents with real vector representation that can represent the inner concept of text is easy to define. Secondly, for symbolic data similarity is often determined making use of special purpose similarity measures and distance functions, for a great deal depending on the application domain under consideration.

To cope with non-vectored information two approaches exist. First, prepossessing approach, textual information will have transferred to real vector used for one standard kernel function. Secondly, kernel function map real vector in feature space F.

The scope of my work is the final step, the classification of the files as being typical for a certain stage, for which machine learning (ML) methods, more specially $TFIDF$, PCA (Principal Component Analysis), LDA (Latent Dirichlet allocation) and $K-mean$, have been applied.

1.4 Limitation

Major clustering performance are being analyzed through cross validation of cluster e.g. similarities and dissimilarity measurements, Value for specific cluster and whole cluster and (e) Correct cluster with x objects should be better than noisy cluster with x object.

1.5 Information About EDGAR Filings

EDGAR, the Electronic Data Gathering, Analysis, and Retrieval system, performs automated collection, validation, indexing, acceptance, and forwarding of submissions by companies and others who are required by law to file forms with the U.S. Securities and Exchange Commission (SEC). Its primary purpose is to increase the efficiency and fairness of the securities market for the benefit of investors, corporations, and the economy by accelerating the receipt, acceptance, dissemination, and analysis of time-sensitive corporate information filed with the agency [1].

Not all documents filed with the Commission by public companies will be available on EDGAR. Companies were phased in to EDGAR filing over a three-year period, ending May $6, 1996$. As of that date, all public domestic companies were required to make their filings on EDGAR, except for filings made in paper because of a hardship exemption. Third-party filings with respect to these companies, such as tender offers and Schedules $13D$, are also filed on EDGAR.

However, some documents are not yet permitted to be filed electronically, and consequently will not be available on EDGAR. Other documents may be filed on

EDGAR voluntarily, and consequently may or may not be available on EDGAR. For example:

1. Form 144 (notice of proposed sale of securities) may be filed on EDGAR at the option of the filer.

2. Forms 3, 4, and 5 (security ownership and transaction reports filed by corporate insiders) filed before June $30, 2003$ may be filed on EDGAR at the option of the filer, but those filed on or after that date must be filed on EDGAR.

3. Filings by foreign companies and foreign governments before November $4, 2002$ either could be made on EDGAR at the option of the filer, or were not permitted to be filed electronically, but from that date on, these filings must be made on EDGAR.

It should also be noted that the actual annual report to shareholders (except in the case of investment companies) need not be submitted on EDGAR, although some companies do so voluntarily. However, the annual report on Form $10-K$ or $10-KSB$, which contains much of the same information, is required to be filed on EDGAR.

Figure 1.1: EDGAR Filing

1.6 Related Work

The thesis is organized as follows. In Chapter 2 we will present the preprocessing which are needed to treat textual data. In Chapter 3 we will look at different similarity measures in data points and extract important features from textual data. In Section 3.6 we focus two different approaches supervised and unsupervised learning to identify the ordering or patterns in a data set.

In Chapter 4 we look at the clustering techniques of the data vectors and the main results on this regard. We will look on different evaluation measure for cluster's. Also check the benefits in different fields of machine learning.

In Chapter 5 we describe the proposed model for textual data mining and comparison of different methods. We will look on implementation on the proposed model and check the results. Also implement semi-supervised classification using through different models.

2 Text Prepossessing

This chapter provide a introduction of text prepossessing. In this phase raw data will be processed and convert to machine understandable. Unnecessary words and semantic word will be remove from text [2]. Pre-processing phase is concerned in producing the machine interpretable representation of given text documents before applying clustering techniques. The main procedure of text prepossessing is represent in diagram 2.

Text pre-processing is an essential part of any NLP system, since the characters, words, and sentences identified at this stage are the fundamental units passed to all further processing stages, from analysis and tagging components, such as morphological analyzers and part-of-speech taggers, through applications, such as information retrieval and machine translation systems [3]. Text documents often contains some unnecessary information like stop words and special formats like number format and date format. So after removal of these word we normally reduce the size of document for better results.

1. To reduce indexing(or data) file size of the Text documents

    a) Stop words accounts $20 - 30\%$ of total word counts in a particular text documents.

    b) Stemming may reduce indexing size as much as $40 - 50\%$.

2. To improve the efficiency and effectiveness of the system

    a) Stop words are not useful for searching or Text mining and they may confuse the retrieval system.

    b) Stemming used for matching the similar words in a text document.

More generally, we are interested in performing some basic task and transformations on the text, in order to be left with artefacts which is used for further processing, more meaningful analytic task afterward [4].

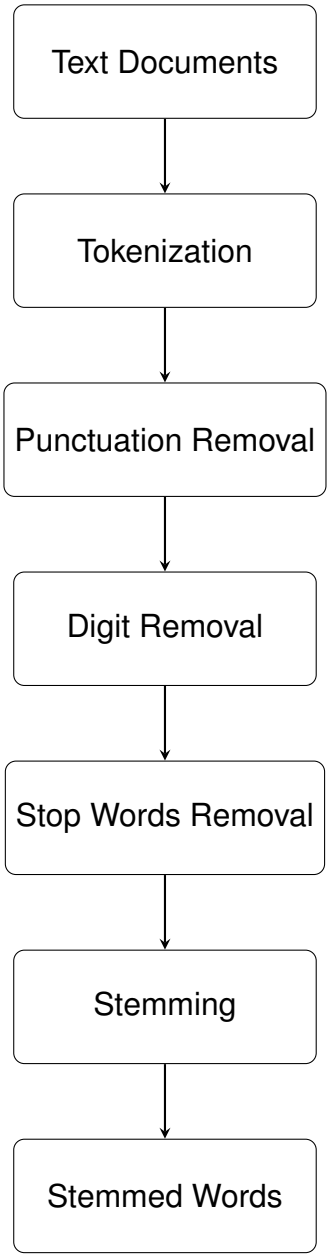

Figure 2.1: Steps of Text Preprocessing

Document	Contents
$D1$	the dr. lives in a blue box
$D2$	Natural language processing (NLP) is a field of computer science
$D3$	My favorite color is blue

Table 2.1: List of Documents

2.1 Tokenization

Definition 2.1 It is a process in which we break down the long sentence into words, phrases, symbols, or other meaningful elements are called tokens.

After applying this process we get a list of token that is become input for further processing. Tokenization is useful part for lexical analysis. All processes in information retrieval require the words of the data set. It is sounds trivial because text is already save in machine readable format but still some problems will arise due to hyper links, punctuation mark or some other formats.

$D1$	$D2$	$D3$
the	Natural	My
dr.	language	favorite
lives	processing	color
in	(NLP)	is
a	is a	blue
blue	field	
box	of	
	computer	
	science	

Table 2.2: Tokenization of Documents

we competing strategies such as keeping the punctuation with one part of the word, or discarding it altogether. The main purpose of tokenization is highlight the meaningful words. These is a problem in abbreviations and acronyms which have to be transformed into a standard form.

Tokenizing unsegmented language sentences requires additional lexical and morphological information. Tokenization is also affected by writing system and the typographical structure of the words. Structure of languages can be grouped into three categories [3]:

Isolating: Words do not divide into smaller units. Example: Mandarin Chinese.

Agglutinative: Words divide into smaller units. Example: English, German.

Inflectional: Boundaries between morphemes are not clear and ambiguous in terms of grammatical meaning. Example: Latin.

2.2 Stop Words Removal

Many words in a document or frequently used but they are essential meaningless in text mining. It is used for joining the words in sentence together. Stop words do not contribute to the context or content of textual documents. High frequency of these common words is obstacle to understand the meaning of document content.

Stop words are commonly used and not useful for the classification of document. So, these words will be removed and this process will reduce the text data and improve the performance of algorithm. These include:

1. Set all characters to lowercase
2. Remove numbers (or convert numbers to textual representations)
3. Remove punctuation (generally part of tokenization, but still worth keeping in mind at this stage, even as confirmation)
4. Strip white space (also generally part of tokenization)
5. Remove default stop words (general English stop words)

$D1$	$D2$	$D3$
dr. lives blue box	Natural language processing (NLP) field computer science	favorite color blue

Table 2.3: Removal of Stop Words in Documents

2.3 Stemming

Definition 2.2 Stemming is the process of conflating the variant forms of a word into a common representation, the stem.

For example, the words: "presentation", "presented", "presenting" could all be reduced to a common representation "present". The stemming process remove the morphological and inflexional endings from words. It is widely used in information retrieval for text data.

$D1$	$D2$	$D3$
dr. live blue box	Natural language process (NLP) field computer science	favorite color blue

Table 2.4: Stemming of Words in Documents

There are mainly two errors in stemming.

Over-stemming is when two words with different stems are stemmed to the same root. This is also known as a false positive.

Under-stemming is when two words that should be stemmed to the same root are not. This is also known as a false negative.

2.4 Lemmatization

Lemmatization is related to stemming, differing in that lemmatization is able to capture canonical forms based on a word's lemma.

$$Better- > Good \tag{2.1}$$

For example, stemming the word "better" would fail to return its citation form (another word for lemma); however, lemmatization would result in the following:

It should be easy to see why the implementation of a stemmer would be the less difficult feat of the two.

3 Features Extraction and Similarity Measures

Another important aspect is to identify common features found in several documents. Similarities in these text in terms of content and interest increases with the increase in number of common features [7]. Although to find similar text, which have same kind of interest and same kind of content is a very firm task. The main challenges which makes it hard to identify similarities between is the heterogeneous nature of information coming from content, graph structures, interaction data etc. and distinct kind of nodes. Some nodes have millions of followers and some has very less [9].

Similarity measure between different entities is understood by different types of definitions in the field of statistics and data mining. Similarity measures have been proven as a very important key step in different technologies such as clustering, classification, recommendation engines and anomaly detection. Computing similarity is to compute identicalness between two entities. In data mining, similarity increases with decrease in distance and decreases with the increase in distance between various dimensions constituting features of the entities [10]. Similarity is usually computed scaling from 0 as lowest similarity to 1 for identically similar entities.

- Similarity $= 1$ if $A = B$,
- Similarity $= 0$ if $A ! = B$, (where A, B are two entities)

There are various kinds of methods to evaluate similarity distance and here the five mostly used similarity measures which are Euclidean distance, Cosine similarity, Jaccard similarity, Manhattan distance and Minkowski distance.

3.1 Euclidean distance

Definition 3.1 Euclidean distance is the most common distance measure for identifying similarity or dissimilarity between two points or vectors is euclidean distance as it is simple to calculate and works better for a continuously coming data or very dense data set.

Euclidean distance takes use of Pythagorean theorem to calculate the shortest distance connecting the two points.

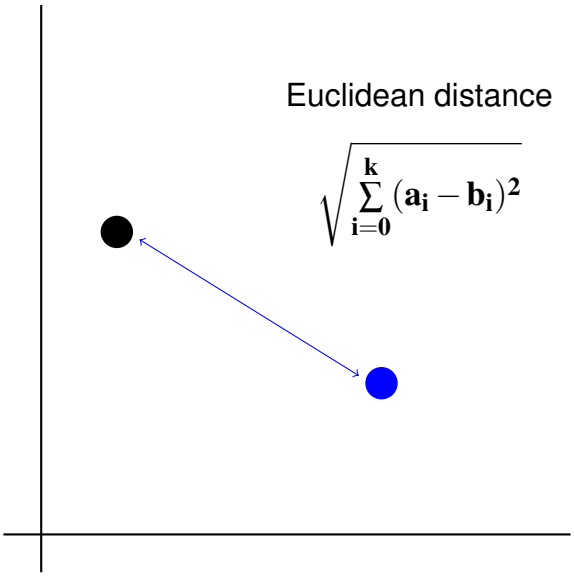

Figure 3.1: Euclidean Distance

As shown in Figure 3.2, the distance between black circle and blue circle is calculated by taking out the sum of the difference between corresponding dimensions of the points or vectors and then taking out the square root of the total sum as shown in eq (3.1). Euclidean distance does not allow data to be normalized to the same level, so it is always good to use euclidean distance when needs to measure data on the same scale. For Euclidean distance, distance between n dimensions of both a and b are calculated by taking square root of the total sum of square of the difference between them.

$$d(a-b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 .... (a_n - b_n)^2} \tag{3.1}$$

Then, the above eq (3.1) can be written in the standard form as:

$$d(a-b) = \sqrt{\sum_{n=0}^{\infty}(a_i - b_i)^2} \tag{3.2}$$

3.2 Cosine similarity

Cosine similarity is a measure of similarity which helps in calculating out the cosine of the angle created in between the inner product measure $(A.B/|A||B|)$ of the two non zero vectors. The highest value of similarity which can be obtained is 1, which is the highest because cosine similarity ranges from 0 to 1. Cosine of 0 degree is equal to 1 and it is always less than 1 for any other angle between the vectors.

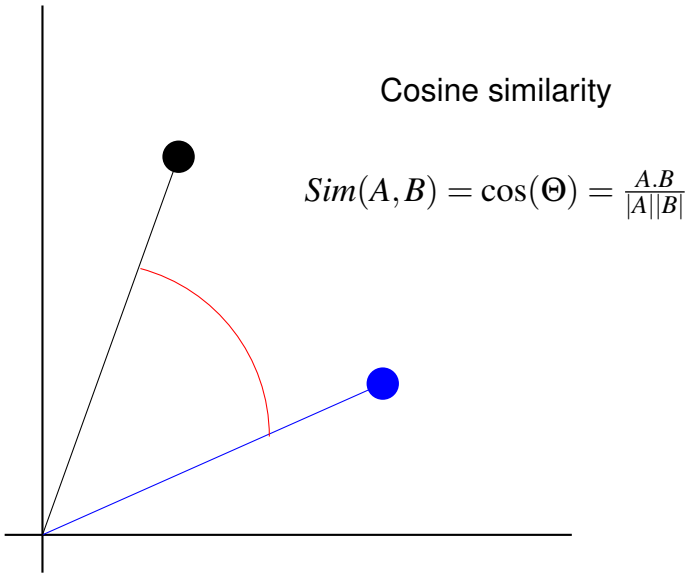

Figure 3.2: Cosine similarity

As shown in equation below, cosine similarity can be calculated by calculating the dot product of A and B and dividing it by the Modulus of A and B.

$$Sim(A,B) = \cos(\Theta) = \frac{A.B}{|A||B|} \tag{3.3}$$

As in Figure 3.2, the cosine similarity is calculated on the basis of the angle created between the black circle and blue circle. Cosine similarity compute the similarity by taking out the normalized dot product of the two given points or vectors as shown in eq (3.3). Cosine similarity is very useful when working with

sparse data and it is usually used in positive space, where outcome will always be in between 0 and 1.

3.3 Jaccard similarity

Jaccard similarity is being used to compute the similarity in between finite number of sets and is calculated by taking out the cardinality of the intersection of the sets and dividing it by the cardinality of the union of the sets as shown in eq (3.4).

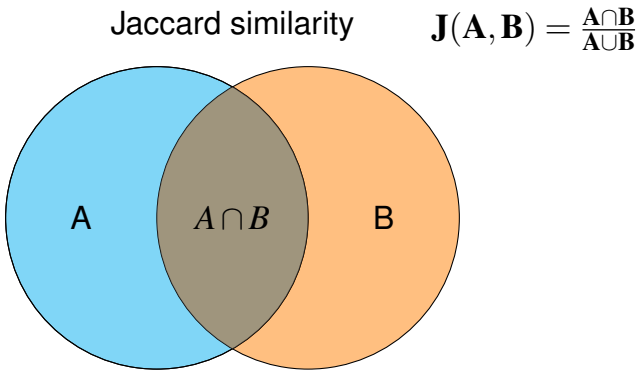

Figure 3.3: Jaccard similarity

$$Sim(A,B) = \frac{A \cap B}{A \cup B} \tag{3.4}$$

As shown in Figure 3.3, Jaccard similarity in between two sets A, B is taken out by taking out the cardinality of the intersection of A and B and dividing it then by Cardinality of their union. Jaccard similarity also has a same similarity range as in cosine similarity, where 0 is the lowest similarity and 1 as the highest similarity between two finite sets or points. As shown in the below Figure 3.6, Jaccard similarity can be calculated by calculating the intersection between A and B and then dividing the result by the union of A and B.

3.4 Manhattan distance

Manhattan distance is the measure of distance between two entities by calculating the absolute difference between all the cartesian coordinates of these entities or in other words it is calculated by calculating the difference between x coordinates and y coordinates respectively. Manhattan is also commonly known as $L1$ norm or distance, rectilinear distance, Manhattan length, Minkowski's $L1$ distance, city block distance, taxi-cab metric and city block distance.

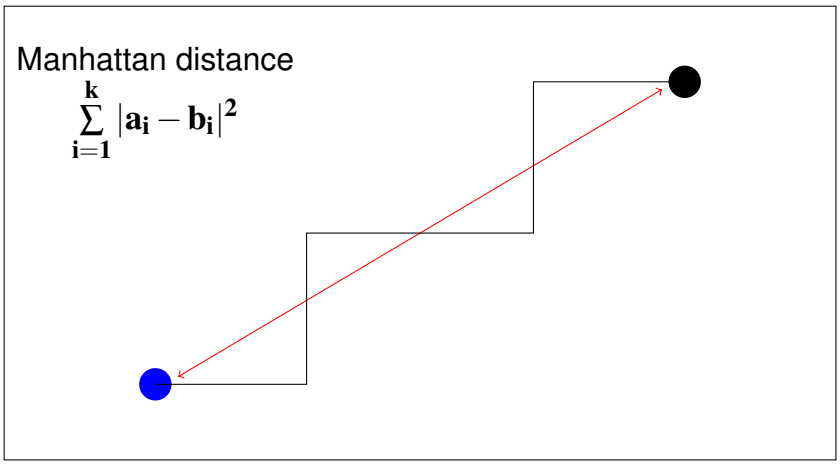

Figure 3.4: Manhattan distance

$$d(a-b) = \sum_{i=0}^{\infty} |a_i - b_i|^2 \tag{3.5}$$

Lets say there are two points, as shown in Figure 3.4 by black and blue circle, Manhattan distance between these two points will be calculated by computing the sum of variations in absolute $x-axis$ and $y-axis$. This is measured through axes at right angles between two points by using eq (3.5), as shown in Figure 3.4, Manhattan is calculated same as Euclidean distance.

3.5 Minkowski distance

Minkowski distance is calculated by generalizing the Manhattan distance and Euclidean distance metric or measure.

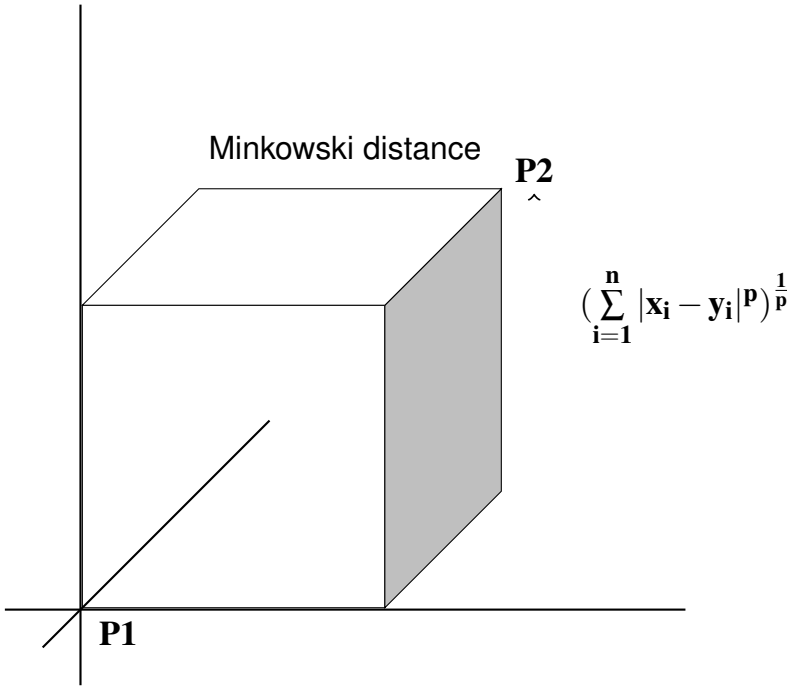

Figure 3.5: Minkowski distance

$$dis(x,y) = (\sum_{n=0}^{\infty} |x_i - y_i|^p)^{\frac{1}{p}} \tag{3.6}$$

As shown in Figure 3.5, Minkowski distance between two points or vectors is measured by computing the different Minkowski metric orders on various variables of the given points by using eq (3.6). Minkowski is known by various names because of its different metric orders, which are:

- $p = 1$, is the Manhattan distance. In this case it is also known by $L1$-norm, city-block distance or Taxicab. Minkowski distance, sometime also called as Foot-ruler distance, when working with ranked ordinal variables of two vectors.

- $p = 2$ is the Euclidean distance. In this case it is also known as Ruler distance or $L2$-norm. Minkowski distance, sometimes also called as Spearman distance, when working with ranked ordinal variables of two vectors.
- $p =$infinity is the chebyshev distance. In this case it is also known as chessboard distance or Lmaxnorm.

3.6 Pattern Identification techniques

Pattern identification or pattern recognition comes from the field of machine learning. Its focus is on identifying the ordering or patterns in a data set. Their are two approaches in pattern identification which includes supervised learning, in which rules or patterns are learned from trained labeled data set and unsupervised learning, in which unknown rules or patterns are discovered from unlabeled data set. Pattern identification algorithm takes statistical variation into account to find out the most possible matching inputs of all the possible inputs taken.

Pattern identification takes all given features of an entity and repeats this process for every entity present in the given data set. Pattern identification algorithm takes use of probabilistic approach and finds out the ordering of features based on highest probability or regularity in the data set and then create a set of rules or ordering of features in decreasing order of probability or regularity in the data set. Pattern identification can be done in different ways:

Association rules help in understanding conclusive association relationships between a group of objects. It helps in discovering set of rules by multiple levels of concepts from the group of objects. For example, it helps in discovering symptoms that are occurring with specific kind of diseases and can identify the reasons for the diseases. In recent time, a lot of interest has been grown in the field of mining association rules from a data set containing different dimensions. It helps in finding fascinating patterns or rules in data set and can be used in medical diagnosis, selective marketing, decision support patterns, financial forecasts and many other different applications [11].

To produce interesting rules, there are various measures that needs to be consider. Most useful and highly accepted measures are confidence, lift, conviction and support which are explained below: Lets take an example as shown in Table 3.6, by which these measures can be explained. Let A be an item set and there is an association rule $A \Rightarrow B$ with L as a set of different l transactions.

Transaction Id	Egg	Pepper	Salt	Chicken	Beer
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	1	0	0
5	0	1	0	0	0
6	1	1	1	1	1

Table 3.1: Transaction table of items

Support helps in discovering the regularity of the set of items from entire data set. Referring to eq (3.7), to find support of A with respect to L is the proportion of transaction l, which consists of item set A. Now, lets compute support of A(Chicken, Beer), this will be $\frac{2}{6}$, because A comes two times in six transactions and now its support will be 0.334.

$$supp(A) = \frac{|t \in T; A \subset t|}{|T|} \tag{3.7}$$

Confidence is the proof of how regularly the rule has been founded true. Referring to eq (3.8), the confidence value for $A \Rightarrow B$ with respect to total transactions, is the proportions of the number of transaction which consists of A and also contains B. Now, lets compute confidence of (Chicken, Beer) $\Rightarrow$ (Pepper), which is $\frac{0.16}{0.334}$ and constitutes to 0.49. It means that around 49 percent times customers buys Bread, if they have bought Beer and Chicken.

$$conf(A \Rightarrow B) = \frac{support(A \cup B)}{support(A)} \tag{3.8}$$

Lift in association rule is the observed support ratio, if support of A and support of B is independent. Considering $A \Rightarrow B$, as shown in eq (3.9), let compute Lift on an example $(Chicken, Beer) \Rightarrow (Pepper)$ is computed by $\frac{0.16}{0.66 \times 0.33}$ because support of $A(Chicken, Beer) \Rightarrow B(Pepper)$ is 0.16 and individually support of $(Chicken, Beer)$ is 0.66 and support of $(Pepper)$ is 0.33, which gives us

the value of lift that is 0.76.

$$lift(A \Rightarrow B) = \frac{support(A \cup B)}{support(A) \times support(B)} \tag{3.9}$$

Another measure of association rule is *Conviction*, which is defined in eq (3.10).

$$conv(A \Rightarrow B) = \frac{1 - support(B)}{1 - conf(A \Rightarrow B)} \tag{3.10}$$

Again using the rule used in above equations $(Chicken, Beer) \Rightarrow (Pepper)$, by referring eq (3.4), the conviction for this rule will be $\frac{1-0.33}{1-0.49}$, that is $\frac{0.67}{0.51} \Rightarrow 1.31$, which explains the ratio of expected number, where A occurs without B, if A and B are independently divided by the observed by the number of false predictions.

Sequential pattern analysis is used for analyzing sequential data, to identify or to discover sequential patterns in the given data set of features. Sequential pattern mining is used to for identifying sub-sequences of interest in the set of sequences, where interest consists of various components such as length, profit and frequency of occurrence [6].

Trend analysis is is also a well known operation to gather information and then identifying a pattern. Trend analysis is usually done on time series data set. Time series is formed by doing same calculation over a fixed interval of time frame on a regular basis.Time series indicates or represents the values taken by an object for a time period such as a year or a month.

Time series can consists of different types of data points such as numerical, categorical or symbolic by nature. Instead of focusing on the each data points in the time series, focus can be made on different segments and by this interesting patterns can be discovered and can identify and understand different trends from it. Trend analysis is very important because of the sequential results of an object for a specific time period can help in forecasting the future characteristics or behavior or even help in discovering the probable causes behind the outcome [12].

3.7 Feature Extraction Methods

Feature selection is more common and easy to apply in the problem of text categorization [13] in which supervision is available for the feature selection process. Also, number of unsupervised methods are available for feature extraction in document clustering. Some of methods are described.

3.7.1 Document Frequency-based Selection

The simplest possible method for feature extraction in document clustering is that of the use of document frequency to filter out irrelevant features. While the use of inverse document frequencies reduces [TF-IDF] the importance of such words, this may not alone be sufficient to reduce the noise effects of very frequent words [15]. Those words which are too much frequent in the corpus of documents can be removed because they are typically common words such as "is", "an", "the", or "of" which are not discriminative from a clustering aspect. Such words are also referred to as stop words. A variety of methods are commonly available in the literature [14] for stop-word removal. Typically commonly available stop word lists of about 300 to 400 words are used for the retrieval process. In addition, words which occur extremely infrequently can also be removed from the collection. This is because such words do not add anything to the similarity computations which are used in most clustering methods.In some cases, such words may be misspellings or typographical errors in documents. Noisy bag of words which are derived from the web, blogs or social networks are more likely to contain such terms. We note that some lines of research define document frequency based selection purely on the basis of very infrequent terms, because these terms contribute the least to the similarity calculations. However, it should be emphasized that very frequent words should also be removed, especially if they are not discriminative between clusters.

The $tf-idf$ score for a term at position i in document j is computed as

$$(tf-idf)_{ij} = tf_{ij} \times idf_i \tag{3.11}$$

Where tf_{ij} is the term frequency for term i in document j. idf_i, is the inverse

document frequency for a term t_i expressed as

$$idf_i = log\frac{|D|}{|\{d : t_i \in d\}|} \tag{3.12}$$

$|D|$ denotes the total number of documents in the corpus and the denominator, $|\{d : t_i \in d\}|$ are the number of documents in which therm t_i exists.

The $tf - idf$ weighting scheme will increase the weight of terms that have frequent occurrence in a smaller set of documents and lower the weight of those terms that are frequently occurring over the entire corpus. $tf - idf$ is just one out of many different weighting schemes [16].

Note that the $td - idf$ weighting method can also naturally ïňĄlter out very common words in a "soft" way. Clearly, the standard set of stop words provide a valid set of words to prune. Nevertheless, we would like a way of quantifying the importance of a term directly to the clustering process, which is essential for more aggressive pruning.

4 Clustering Approaches

Clustering or cluster analysis is used to group together unlabeled data into similar groups known as clusters. Cluster analysis have many definitions like it helps in finding the groups or clusters when their is no knowledge about the categories of data set, it helps in dividing a set of entities into compact group of small sets by using similarity on different dimensions of these entities to form homogeneous clusters or organizing sets of multidimensional patterns into different groups or clusters etc. Clustering is used for identifying a group of objects or grouping them together such that, the objects in the same group are very much similar to each other with respect to the users which are present in any other group.

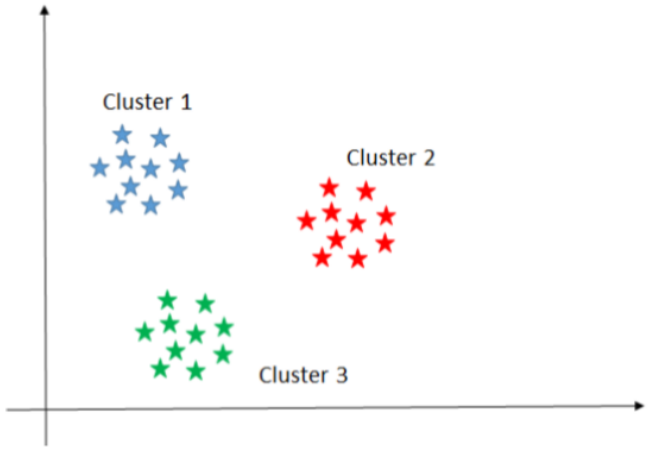

Figure 4.1: Cluster analysis

As shown in Figure 4 Clustering is a one of the main field of data mining and is very commonly used in statistical analysis of data set. It is very commonly used in the field of machine learning, image analysis, computer graphics, bio-informatics, information retrieval tasks and image analysis. Clustering approaches are compared to each other by the help of certain properties that are separation,dimension, density, shape and variance. Less variance produces very compact and tight clusters [18].

4.1 Clustering Requirements

There are different measures required to perform cluster analysis on data that are shown in Table 4.1, consists of Data presentation, Objects choice, Variables choice, What to cluster, Normalization, Similarity (proximity) measures, Clustering criteria, Handling missing data, Clustering algorithms, Clusters number and Result interpretation. Some of them are explained below:

Proximity measure is used for performing clustering, which can be either similarity measure or dissimilarity (or distance) measure. As it is shown in figure 4.1, large distance between objects have lower similarity and on the other side objects with smaller distance to each other have higher similarity. For proximity measure, different distance measures are required, which was explained earlier in similarities section.

Figure 4.2: Proximity measure

Parametric design should be chose according to nature of data. Assumptions will be made according to the form of the distribution used to model the data by the cluster analysis. For example, it is convenient the data will be modeled by multi-variant Gaussian mixture model [18].

Shape of cluster is depends on the position, size and density of n elements in certain groups. These parameters have impact on different clustering algorithms, as the description of the algorithms will show. Therefore, changing of the clustering algorithm directly impact the design parameters.

K-number of clusters could be fix if desired number is known or could be varied to find optimal number of clusters. As Duda $et\ al.\ (2000)$ state, 'In theory, the clustering problem can be solved by exhaustive, so the sample set is finite, so there is finite number of partitions are possible; in practice, such an approach is unthinkable for all but the simplest problems'.

Error; words have different meanings, so that can involve in different classes. This is only happens when we used soft clustering algorithm, which define probability of cluster object that can member of cluster. Hard clustering algorithm have yes or no probability that can only member a one class and have lesser chance of error.

Requirements	**Definition**
Data presentation	It includes data pre-processing and formatting, so that it can be presented in a usable format.
Objects choice	It is the process of selecting the entities or objects which needs to be grouped or clustered.
Variables choice	It states the choice of different dimension of the objects, on which clustering will be performed and it has to be same for all the objects in the data set.
What to cluster	It states what needs to be cluster, data units or objects variables.
Normalization	Adjusting different values of variables or dimension to the same scale of measurement.
Similarity measure	Need to have a similarity measure by which it can find similarity between different objects or entities.
Clustering criterion	Provide criterion parameter to the clustering algorithm, so that it can produce very compact clusters, which are homogeneous in nature
Handling missing data	Data set can have missing data, so there should be some method by which it can handle or approximate missing data.
Clustering algorithms	It is the final part of clustering, where clustering algorithm have to be applied to perform cluster analysis on given data set.
Clusters number	It is optional, in some algorithm to provide number of cluster to be made and in some it is not required.
Result interpretation	Need to understand the result produced by the clustering algorithm, so that it can be compared with results of different algorithms and can have the most optimal solution for analysis.

Table 4.1: Clustering requirements

4.2 Clustering criterion

There is always a clustering criterion to perform clustering between different objects. These are the different dimensions or features of the object, which are taken into account when performing cluster analysis. Cluster analysis must be performed on the same feature of different objects, otherwise it will produce false results or no result.

4.3 Clustering Algorithms

There are various algorithms, which are needed to perform clustering. The most fitting algorithm to be used, depends upon the task and can be identified by experimenting and analyzing their results. So, there is no clustering algorithm which is always correct, it can happen that an algorithm which is producing good result for a data set and can produce different or bad results for a different kind of data set.

4.3.1 Hierarchical Algorithms

Hierarchical clustering is a connectivity based algorithm, which perform clustering based on previously based clusters. Hierarchical clustering make clusters by connecting the objects based on their distance. Dendrogram is used to represent different clusters based on their distances. This type of clustering does not perform clustering by single partition of data set, but provide an comprehensive hierarchy of different clusters that integrate with each other at some distances. For some kind of data set where it needs to analyze the tree kind structure of the results, by which hierarchy of related objects like family structure or biological genetics etc, can be obtained. Hierarchical clustering is preferred over flat clustering. As, shown in Figure 4.3.1, this algorithm works in Agglomerative (bottom-up) or Dicisive (top-down) way. There are two types of linkage methods:

1. Maximum or complete linkage minimizes the maximum distance between observations of pairs of clusters.
2. Average linkage minimizes the average of the distances between all observations of pairs of clusters.

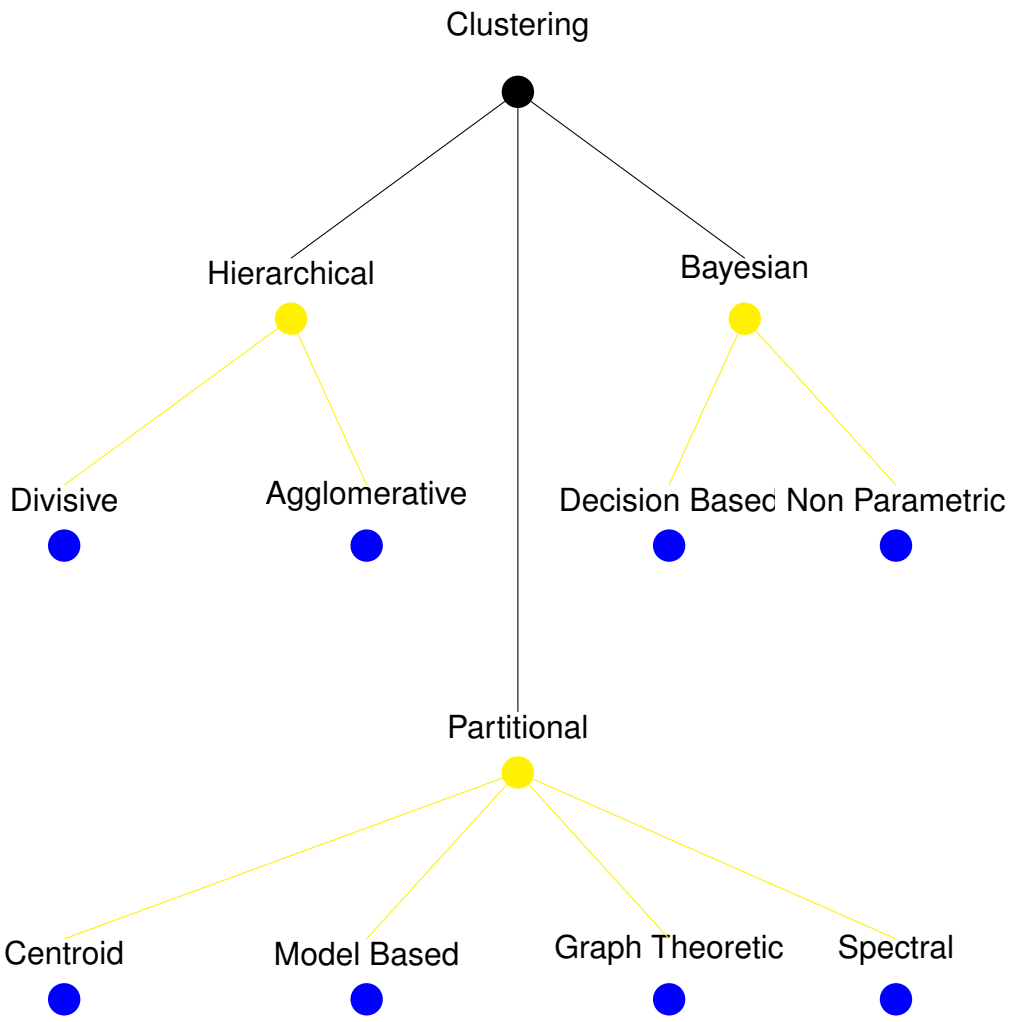

Figure 4.3: Cluster techniques

Agglomerative starts from the bottom by merging the most closed or nearest pairs of groups and then at last stops, when all different clusters are merged into one cluster. On the other hand, Dicisive works in a different way, it starts from the top with root as one cluster and start dividing it into different clusters and stops at the bottom level where each cluster is left with only a single object [17].

4.3.2 Bayesian clustering

Bayesian clustering identifies all the clusters at once but can also be used as hierarchical clustering. As, shown in Figure 4.3.1, there are two types of techniques in which Bayesian works, which are Non-parametric and Decision based.

Non-parametric clustering is a unsupervised learning approach, in which it do not consider any density function but focus on finding natural groups or clusters. Decision based clustering makes use supervised algorithm to provide unsupervised results. In this method, different targets or classes are provided to the algorithm and then apply clustering algorithm on it to produce compact clusters. In the first step, it take exactly the matching variables with same minimum and maximum values, then decision algorithm will find the smallest and the highest densities in the data set, which is used to train the model for performing clustering task. From this, the algorithm will learn the density distribution between the dimensions and can be applied on the testing data and finally produce different clusters, which are the leaves of the decision tree.

4.3.3 Partitional clustering

Partitional clustering is the most accepted category of doing clustering and it works by generating posteriori distribution on group of all data partitions. Partitional clustering curtails the given cluster measures by repeatedly re-positioning the data points between different clusters until optimal partition is achieved locally because there is no such assurance that every time a global optimal partition will be achieved. As shown in Figure 4.3.1, there are four techniques which are Centroid based or K-means, Model based, Graph theoretic and Spectral model [17].

K-means is the best known partitional based clustering approach and is also the most widely accepted and used clustering method because of its fast processing and its scaling nature for large data set. K-means splits up the data set into k disjoint clusters. Here, each group or cluster have a centroid $ce_1, ..., ce_n$ known as the cluster center and number of clusters or groups are formed based on k value, which is provided or given by the user and it should equal or less than to number of data points [17]. An useful view of clustering is the following: Given a space X, clustering could be thought of as a partitioning of this space into K parts i.e. $f : X \rightarrow \{1, ..., K\}$. Usually this partitioning is obtained by optimizing some internal criteria such as the inter-cluster distances, etc. The optimising criterion in the clustering process is the sum-of-squared-error SSE between objects in

clusters

$$SSE = (\sum_{n=1}^{k} \sum_{o \in C_n} d(y - ce_n)^2) \tag{4.1}$$

Idea behind using Model based clustering is that, observations comes from a mixture of various components. Every component is associated with a weight or associated probability in the components mixture. For computing the probability, any probability finding model can be used but components are p-variate normal distribution and thus need to have this probability model which is a multivariate normal distributions mixture. And then, the result will be a mixture of components, where each component is equal to a cluster.

Graph theoretic clustering works in two phases based on minimum spanning tree (MST). It separates the the clusters into two parts, one is separate clusters and the other one touching clusters and solve both clustering parts with two different algorithms, firstly 2-round MST are used to detect separate clusters on the basis of distance and density difference and then it constructs a graph. In the second part of the algorithm, it takes care of the touching clusters, which are the sub-groups eventually produced in previous step. It takes care of these sub-groups by comparing their cuts, respectively [19].

Spectral clustering works by using eigenvalues and eigenvectors of the used similarity matrix over the data set to reduce the dimensions and then performing clustering on less dimensions. Generalized way of doing spectral clustering is to use any standard clustering approach on applicable eigenvectors of Laplacian matrix.

4.4 Clustering Evaluation

There are various evaluation measure, which have different background and demands to perform clustering evaluation. The most fitting algorithm to be used, depends upon the evaluation measure and best result at the end for each task. So, there is no clustering evaluation measure which is check the accuracy of generate clusters. Not all the described evaluation measure will apply on clustering algorithm's.

4.4.1 Sum of Squared Error

Summing over the squared distances between the clustering data objects x and their cluster prototype y. We can measure the deviation from exact solution in terms of sum of square error, such as

$$SSE = \frac{1}{2}(\sum_{p}(f_w(x_p) - y_p)^2) \tag{4.2}$$

Whereas f is activation function that map input data points x to output object y. p is training set for clustering.

Sum of square error SSE generally refers on Euclidean distance between data points x and centroid of cluster c. So, cost function of SSE is;

$$SSE = (\sum_{n=1}^{k} \sum_{o \in C_n} d(y - ce_n)^2) \tag{4.3}$$

4.4.2 Precision and Recall

In document clustering scenario, the instances are documents and the task is to return a set of relevant documents in appropriate clusters; or equivalently, to assign each document to one of two categories, "bound" and "not bound". In this case, the "bound" documents are simply those that belong to the "bound" category.

Recall is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents, while precision is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search.

In the field of document clustering, precision is the fraction of retrieved documents

that are relevant to the features:

$$precision = \frac{t_p}{t_p + f_p} \tag{4.4}$$

In information retrieval, recall is the fraction of the relevant documents that are successfully retrieved.

$$recall = \frac{t_p}{t_p + f_n} \tag{4.5}$$

True positive t_p are number of pair common objects in cluster C_1 and C_2, false positive f_p the number of pair common is cluster c_1 but not in C_2, false negative f_n are number of common pair in cluster C_2 but not in C_1.

4.4.3 Rand Index

Rand (1971) defines an evaluation measure for a general clustering problem on basis of agreement vs. disagreement between object pairs in clustering. He states that clusters are defined as much by those points which they do not contain as by those points which they do contain.

$$RI = \frac{a+d}{a+b+c+d} \tag{4.6}$$

- $a =$ Number of pairs in same cluster C in data sample S_1 and S_2.
- $d =$ Number of pairs in different cluster C in data sample S_1 and S_2.
- $c =$ Number of pairs where one pair in one cluster C in data sample S_1 and other pair in different cluster of data sample S_2.
- $b =$ opposite of c.

Remark:

1. $RI \in [0,1]$,

2. $RI = 0 \rightarrow a = d = 0$; complete disagreement
3. $RI = 1 \rightarrow c = b = 0$; complete agreement

4.4.4 Conn Index

Conn-Index proposed by Tademir and Erzs [20] , is a validity measure to evaluate clustering of very large data sets. There is presumed C clusters in data set is represented more than one prototype, $N_p > N_c$. Cluster compactness and separation is based on intra C_{intra} and inter C_{inter} connectivity of prototypes $C \in [0,1]$. Number of prototype for one cluster is equal to or more than 2 [21].

$$C = C_{intra}(1 - C_{inter}). \tag{4.7}$$

Thereby, C_{intra} measures the compactness of the clusters and C_{inter} evaluates the separation between them. The calculation of C_{intra} is based on the cumulative adjacency matrix

$$C_{intra} = \frac{1}{N_c} \sum_{K=1}^{N_c} C_{intra}(C_k)$$

$$C_{intra}(C_K) = \frac{\sum_{i,j}^{N_p} \{CAD_j(i,j) : P_i, P_j \in C_k\}}{\sum_{i,j}^{N_p} \{CAD_j(i,j) : P_i \in C_k\}} \tag{4.8}$$

1. $C_{intra}(C_k) \in [0,1]$,
2. $C_{intra}(C_k) \uparrow =$ High compactness of C_k,
3. $C_{intra}(C_k) = 1 \rightarrow$ all $2nd$ BMU within C_k, no connection to other cluster.

For the inter-cluster connectivity C_{inter} the connectivity matrix $C = A^T + A$ is required. Their elements C_{ij} can be interpreted as the dissimilarities between the prototypes, and hence, implicitly contain information about the local data density according to the magniïňĄcation property. The inter-cluster connectivity C_{inter} is

the average of the values $C_{inter}(k)$ analogously to $C_{intra}(k)$ where

$$C_{inter} = \frac{1}{N_c} \sum_{K=1}^{N_c} C_{inter}(C_k) \tag{4.9}$$

is the maximum of the local inter-cluster connectivities $C_{inter}(k,l)$

$$C_{inter}(C_k) = \max . C_{inter}(C_k, C_l)$$

$$C_{inter}(C_k, C_l) = \begin{cases} 0 & \text{if } P_{kl} = \emptyset \\ \frac{\sum_{i,j}^{N_p} \{CONN(i,j) : P_i \in C_k, P_j \in C_l\}}{\sum_{i,j}^{N_p} \{CONN(i,j) : P_i \in P_{kl}\}} & \text{if } P_{kl} \neq \emptyset \end{cases} \tag{4.10}$$

$$P_{kl} = \{P_i | P_i \in C_k, \exists P_j : P_j \in C_l : CAD_j(i,j) > 0\} \tag{4.11}$$

1. $C_{inter}(C_k, C_l) = 0 \Rightarrow$ cluster completely separated,
2. $C_{inter}(C_k, C_l) \uparrow \Rightarrow$ High similarity of clusters,
3. $C_{inter}(C_k, C_l) > 0.5 \Rightarrow$ Prototype in C_k with connection to C_l are more similar to prototypes in C_l then to prototypes in C_k.

Remarks:

1. C_{inter} describe similarities $\rightarrow (1 - C_{inter})$.
2. C_{intra} depends on cluster size, many data points in cluster have high value.
3. C_{inter} depends on prototypes close to cluster border, independent of cluster size.

5 Research Methodology and Proposed Solution

Document clustering is modeling tool for different documents types processing and natural language processing. In this work we develop a class of dynamical systems and an associated learning meta-algorithm resulting in a framework for system identification that enjoys several theoretical and practical advantages. The following section describes the proposed model in details.

5.1 Model

Document clustering is done on basis of the similarity between the extracted features and the individual documents. Let extracted features vectors be $F = \{f_1, f_2, f_3, ...\}$ computed by $TF - IDF$. Documents in term-document matrix is $V = \{d_1, d_2, d_3, ...\}$. Make some topics on the basis of extracted features and assign each topic to documents on basis of LDA topic modeling. Then new document d_{new} will assign the assign the cluster f_c, on basis if the document d_{new} and cluster f_c have minimum distance.

5.1.1 The methodology

Our frame work is depend on these concepts:

- Construct the term-document matrix V from given files using $TFIDF$.
- Reduced the number of column of V matrix using through PCA reduction.
- Apply cosine similarity to measure distance between documents d and extracted features.
- Assign topics to each document on the basis of feature vectors using LDA.
- Apply clustering algorithm to assign each document to cluster on the basis of similarity.

These concepts allow us to develop efficient and tractable methods for system identification using a supervised regression approach.

5.1.2 Preprocessing

Every documents have irrelevant information in given files. Firstly we required to minimize the redundancy words and noise from each document. So,we should need some pre-processing algorithm to remove these things.

- Tokenization of text, that will break down long phrases into small pieces.
- Remove the stop words that will reduce text from $20-30\%$ of total information.
- Apply stemming will also reduced the text.
- Apply lemmatization.

It will generate the better text where we ignore noise in complete document.

Pseudo Code:

- procedure to remove stop words from document(text contents t)
- t_0 = Initial Text(t);
- $token = Tokenize(t_0)$;
- $stop = StopWrods('english')$;
- $for(token(t_0))$
- begin
- $if stop(t_0)$
- then $w := t_0$;
- end

5.1.3 Feature Extraction and Topic Modeling

Feature selection is more complex part of document clustering. At this point we select those feature that have more important as compared to less important. This method is defined for feature selection in case of financial documents, so i should ignore more common words and stick to information about financial values. So, i should focus on information about companies and shareholders values.

These paradigms have high priority as compared to other information like table of contents, information about company, etc.

$TFIDF$ is well know algorithm to extract feature from textual data. Feature extraction in document clustering is that of the use of document frequency to filter out irrelevant features. While the use of inverse document frequencies reduces $[TF - IDF]$ the importance of such words, this may not alone be sufficient to reduce the noise effects of very frequent words. Threshold value for feature extraction will be 70%, that means if words count will be more than that is ignore automatically from feature selection. Those words whose redundancy lesser than threshold value will be important for further implementation.

Topic modeling is method for probabilistic document clustering. The idea of topic modeling is to create a probabilistic generative model for the text documents in the corpus [26]. Main technique is to elaborate corpus as a function of hidden random features, the parameter is which calculate for particular document. Primary assumption in topic modeling as:

- The n documents in the corpus are assumed to have a probability of belonging to one of k topics. Thus, given document might have probability of many topics and same document have multiple topics. For a given document D_m and a list of topics $T = \{T_1, T_2, ..., T_k\}$, the propblity that given document belong to topic T_n is given by

  $$P(T_n|D_m) \tag{5.1}$$

  provides probability membership of mth document to nth topic. In non-probabilistic clustering methods, the membership of documents to clusters is deterministic in nature, and therefore the clustering is typically a clean partitioning of the document collection. The use of a soft cluster membership in terms of probabilities is an elegant solution to this dilemma [26].

- Each topic is associated with a probability vector, which quantifies the probability of the different terms in the lexicon for that topic. Let $t_1 ... t_d$ be the d terms in the lexicon. Then, for a document that belongs completely to topic T_n, the probability that the term t_l occurs in it is given by

  $$P(t_l|T_n) \tag{5.2}$$

  This value is another important feature for topic modeling algorithm.

Pseudo Code:

- Feature extraction and topic modeling
- $D = corpus(documents)$;
- $tfidf = TFIDFModel(D, dictionary)$;
- $n = Numberoftopics$;
- $train(tfidf, n, iterations)$;
- $T = Topic-Model(tfidf[corpus], n, iterations, threshold, min-probability)$;
- $t = Matrix-Similarity(corpus)$;
- return T;

Describe algorithm will generate number of possible topics for each document and return a matrix of document versus possible topic.

Topic Modeling Algorithm:

There are three basic topic modeling techniques that used

- Latent Semantic Indexing (LSI)
- Latent Dirichlet Allocation (LDA)
- Log Entropy Model

Latent Semantic Indexing:

The latent semantic space has fewer dimensions than the original space (which has as many dimensions as terms). LSI is thus a method for dimensionality reduction. A dimensionality reduction method takes a list of features that exist in a high-dimensional space and represents them in a low dimensional space, often in a two-dimensional or three-dimensional space for the purpose of visualization.

Latent semantic indexing is the application of a particular mathematical technique, called Singular Value Decomposition or SVD, to a word-by-document matrix. LSI is a least-squares method. The projection into the latent semantic space

is chosen such that the representations in the original space are changed as little as possible when measured by the sum of the squares of the differences [27].

The above set of random parameters $P(T_n|D_m)$ and $P(t_l|T_n)$ allow us to model the probability of a term t_l occurring in any document D_m. Specifically, the probability $P(t_l|D_m)$ of the term t_l occurring document D_m can be expressed in terms of afore-mentioned parameters as follows:

$$P(t_l|D_m) = \sum_{n=1}^{k} p(t_l|T_n).p(T_n|D_m) \tag{5.3}$$

Each term t_l and document D_m, we can generate a $n \times d$ matrix of probabilities in terms of these features, where n is the number of documents and d is the number of topics. For a given corpus, we also have the $n \times d$ term-document occurrence matrix X, which tells us which term actually occurs in each document, and how many times the term occurs in the document.

LSI has the cons that the number of model parameters grows linearly with the size of the documents. LSI model is not a fully generative model, because there is no accurate way to model the topical distribution of a document which is not included in the current data set.

Latent Dirichlet Allocation:

The second well known method for topic modeling is that of Latent Dirichlet Allocation. In this method, the term-topic probabilities and topic-document probabilities are modeled with a Dirichlet distribution as a prior. Thus, the LDA method is the Bayesian version of the $PLSI$ technique. It can also be shown the the $PLSI$ method is equivalent to the LDA technique, when applied with a uniform Dirichlet prior [28].

LDA is a form of unsupervised learning that views documents as bags of words. The sparse Dirichlet priors encode the intuition that documents cover only a small set of topics and that topics use only a small set of words frequently. LDA is basically works on reverse engineering. We have number of topics n and each topic will assign number of words w from bag of words.

- Assume there are n number of topics for all documents.

- Distribute these n topics across document D_m by assigning each word w a topic.
- For each word w in document D_m, assume its topic is wrong but every other word is assigned the correct topic.
- Probabilistically assign word w a topic based on topics are in document D_m and how many times word w has been assigned a particular topic across all of the corpus of documents.

$$p(D|\alpha,\beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha)\Big(\prod_{n=1}^{N_d} \sum_{Z_{dn}} p(z_{dn}|\theta_d)p(w_{dn}|z_{dn},\beta)\Big)d\theta_d. \tag{5.4}$$

The Dirichlet is a convenient distribution on the simplex - it is in the exponential family, has finite dimensional sufficient statistics, and is conjugate to the multinomial distribution.

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w.

Its main pros over the LSI method is that it is not quite as susceptible to overfitting. This is generally true of Bayesian methods which reduce the number of model parameters to be estimated, and therefore work much better for smaller data sets.

Log Entropy Model:

Entropy is a measure of the unpredictability of the state, or equivalently, of its average information content. Basic idea of information theory is that the "news value" of a communicated message depends on the degree to which the content of the message is surprising. Entropy allows us to make precise argument and perform calculation with regard to not knowing how things will turn out.

Textual data, treated as a string of characters, has fairly low entropy, is fairly predictable. In textual data we will calculate entropy value for each topic basis. The quality of the term is measured by the entropy reduction when it is removed.

$E(t)$ entropy of the term t in a collection of n documents is defined as follows:

$$E(t) = -\sum_{i=1}^{n}\sum_{j=1}^{n}(S_{ij}.\log(S_{ij}) + (1-S_{ij})\log(1-S_{ij})) \tag{5.5}$$

$S_{ij} \in (0,1)$ is the similarity between the ith and jth document in the collection, after the term t is removed, and is defined as:

$$S_{ij} = 2 - \frac{dist(i,j)}{\overline{dist}} \tag{5.6}$$

The $dist(i,j)$ is distance between term ith and jth after term t is removed and $\overline{dist}$ is average distance between documents after t term is removed. Calculation of entropy operation $E(t)$ for each term t requires $O(n^2)$ operations. This is not practical for large corpus of documents because it will required huge computations. If we want this method efficient we will make sampling method for entropy.

5.1.4 K-Means Algorithm:

It's one of the oldest and most widely used technique for clustering data sets. K-means algorithm goal is to find data in groups from unlabeled data sets, whereas number of data group will be provided using through K value. The algorithm iteratively assign a data group K to each data points based on provided extracted features from textual data. It tires to minimize the overall inter-cluster variance, and sum of squared error function.

$$V = \sum_{i=1}^{k}\sum_{x_j \in S_i}(x_j - \mu_i)^2 \tag{5.7}$$

Each centroid defines one of the clusters. In this step, each data point is assigned to its nearest centroid, based on the squared Euclidean distance. where there are k clusters $S_i, i = 1,2,...,k$, and μ_i is the centroid of all the points $x_j \in S_i$. With $k-$means, the number of clusters K should be defined prior to running the algorithm. The complexity of this algorithm is $O(t.d.k.m)$ where d is number of

documents, t is the number of terms, k is the clusters required and m is the maximum number of iteration [30].

The basic algorithm is to converge data points into the nearest of the pre-defined number of clusters until the updated cluster center is within some tolerance of old respective cluster center or until a certain number of stoping criteria has been achieved.

$K-$means algorithm use the extracted topic model from LDA. Make vectors from document topic matrix that will be generated from extracted features. At the end they will provide us with unsupervised clusters from these vectors. We will train our model on test data and save it for future documents. If new coming documents belongs to exciting corpus then no need to make new features. In case of new textual information, our model extract features and make vectors for clustering. In $K-$means then we compare these vectors or near to centroid of cluster. Automatically model assign new document to certain data group.

Pseudo Code and Algorithm:

- Initialize k number of cluster
- I number of iterations
- $Id = -1;$
- def vectors(corpus)
- for docs in corpus
- $Id = Id$ in docs;
- $V = 1 + Id;$
- return $V;$
- def cluster$(corpus, k, V)$
- $C = kmean(V, k, I);$
- return C

5.2 Experiment and Results:

To work with this model, the financial text data set EDGAR filings was used for document clustering with LDA and $K-$means. Aforementioned application was used for the purpose of financial documents clustering and structure detection. This section describes the data set used, experimental parameters and the results.

5.2.1 Data Set:

EDGAR is quite a popular for financial documents of diffident companies from U.S. stock exchange. It has a collection about 21 million documents across different groups. Each group is stored financial information in different filings, with each information about shareholder, bond relevance and un-bond relevance information in different files. Some of the files information are closely related with each other while some are highly unrelated. Some documents are structured bond that look like bond relevance but in actual it is not. These types of information will overlap on bond and not bond categories. These types of documents are noise related information that will never ignore completely. In these case we detect structure towards related category.

5.2.2 Experiment:

For the purpose of experimentation, clustering was done using up to different groups of data points. 1000 documents were taken randomly for 3 groups of filings types each and added to a folder. The folder was indexed after removing the stop-words using English stop-words and applying Porter stemming. Then the $TFIDF$ was done and generate the corpus of extracted features. After that train model for topic generation using LDA. It will generate different topics for each documents. Then train clustering model on the basis of extracted features from corpus of data. After training over model fetch other documents and cluster them using the train model. At the end we generate cluster on the basis of similarity using cosin similarity.

For clustering and topics modeling using following parameter's:

- Pre-processing remove words more than 75%.

- $TFIDF$: extract features from document that more relevance for financial details.
- LDA: topics modeling used 30 different topics for documents.
- $K-$means: $K = 4$, convergence parameter $= 0.001$, maximum iteration $= 10$, distance measure $=$ cosine
- SVC: kernal $=$ linear, $C = 1$
- Decision tree classifier: $max - depth = 5$

The performance of clustering algorithm will be check using through accuracy measures. Also compare other clustering algorithm on the same vectors.

Lastly, i apply semi-supervised classification on the basis of generated cluster. I train model on the basis of generated cluster by $K-$means. I used small amount label data that generated by clustering algorithm and apply on large amount of unlabel data.For classification used two different algorithm Support vector classification and Decision tree classification.

SVM is a discriminant technique, and, because it solves the convex optimization problem analytically, it always returns the same optimal hyperplane parameter-in contrast to perceptrons, both of which are widely used for classification in machine learning. For kernel will used that transform data from input space to feature space.

$$\Big[\frac{1}{n}\sum_{i=1}^{n}(0, 1 - y_i(w.x_i - b))\Big] \tag{5.8}$$

Whereas x_i is test data that are unlabel, y_i class label from train data and b is bias in input data sets. W is weights that apply on test data sets.

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. Leaf node represents a classification or decision. It divides data set into classes that might be pure. We tolerate some percentage of impurity for faster performance. Entropy is used internally to measure the impurity in classes.

Precision and recall will check the accuracy of different classifier to measure the

results. It will given better classification algorithm to adopt for semi-supervised classification.

5.2.3 Results:

Figure 5.2.3 show the topics that will generate using through LDA model. I provided the $TFIDF$ extracted features to topics each document. I choose 30 different topics and method will show what will be the percentage of topics that matching each documents in corpus. I test this algorithm on 1000 different random file of three different category. Each row represent different topics and there occurrence in document shows by percentage values.

```
INFO : topic #93 (0.010): 0.015*"Notes" + 0.008*"Index" + 0.005*"Credit" + 0.005*"Underlying" + 0.004*"Last" + 0.004*"Ref
INFO : topic #64 (0.010): 0.018*"underlying" + 0.011*"UBS" + 0.006*"final" + 0.006*"stock" + 0.006*"Notes" + 0.005*"o" +
INFO : topic #56 (0.010): 0.009*"Credit" + 0.008*"Last" + 0.007*"n/a" + 0.006*"borrower" + 0.006*"notes" + 0.006*"loan" +
INFO : topic #3 (0.010): 0.009*"Credit" + 0.009*"notes" + 0.008*"Since" + 0.007*"0" + 0.007*"Last" + 0.007*"loan" + 0.006
INFO : topic #12 (0.010): 0.014*"Credit" + 0.007*"years" + 0.007*"0" + 0.007*"Since" + 0.007*"Months" + 0.006*"loan" + 0.
INFO : topic diff=50.295895, rho=1.000000
INFO : -8.408 per-word bound, 339.6 perplexity estimate based on a held-out corpus of 1682 documents with 22164692 words
WARNING : scanning corpus to determine the number of features (consider setting `num_features` explicitly)
```

Figure 5.1: LDA Topics on the basis TFIDF features

Below figure 5.1.4 shows the cluster will generated on the basis of features vectors that will generate through topic modeling. Features will save in corpus for training data set for $k-$means. I used the value of $K = 4$ for clustering. Cluster 3 shows the bond relevance documents, cluster 2 shows not bond documents. In figure cluster 1 is structured bond documents. That overlapping on both bond and not bond clusters. It is hard to cluster these data point on the basis accuracy measures. Cluster 4 is noise in textual information. That is ignore after applying the reduction model like PCA. Clustering using this technique is completely heuristic approach.

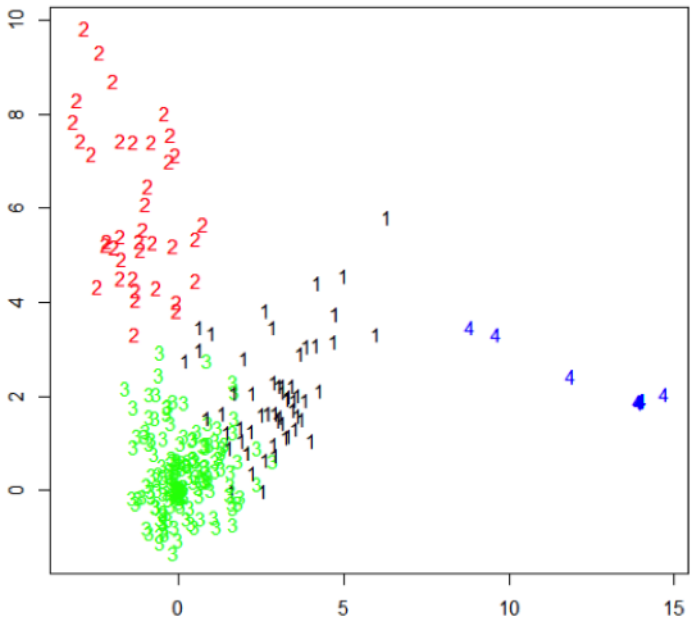

Figure 5.2: K-means clustering with 10 iterations

Semi-supervised classification is using the cluster data points that generated after $K-$means and label these documents. I used two diffident classifier's, support vector machine and decision tree classifier. I train both classifier's on the basis cluster documents and test on randomly 7000 different files. Our train model classify test data and compare the performance of both methods. I calculate precision, recall and accuracy of both model that was train on label data sets.

Lable	*Precision*	*Recall*
1	98%	98%
2	63%	18%
3	89%	96%

Table 5.1: SVM Classifier

Test accuracy for SVM is approximately 96%.

Lable	*Precision*	*Recall*
1	95%	97%
2	0	0
3	83%	82%

Table 5.2: Decision Tree Classifier

Test accuracy for Decision tree classifier is approximately 93%.

6 Conclusion

In this work, a model for financial document clustering was given in this work along with development of application based on this model. This application can be used to organise documents into sub-folders without having to know about the contents of the document. This improves the performance of information retrieval in financial documents. The accuracy of two different classification model was tested for 3 clusters of documents (bond relevant, not bond relevant, none of them). LDA and $k-$means has shown to be a good measure for clustering document and extracted features are used as the final cluster labels for $k-$means algorithm.

The extension results in an efficient and local minima-free method for learning non-linear partially observable continuous systems. This gives us a building block for prediction and reinforcement learning in complex environments.Implementation of framework to update the system on document processing. These concepts allow us to develop efficient and tractable methods for system identification using a supervised regression approach.

Bibliography

[1] EDGAR Filings https://www.sec.gov/edgar/aboutedgar.htm

[2] SUNITA SARKAR An Investigation of Computational Intelligence Techniques and Their Application PhD thesis. Assam University, 2015.

[3] Dr.S.Kannan and Vairaprakash Gurusamy. Preprocessing Techniques for Text Mining *Conference Paper*, 2014.

[4] Matthew Mayo A General Approach to Preprocessing Text Data *Framework for approaching textual data science tasks*, (17:n46), 2017.

[5] Durmaz,O.Bilge, H.S "Effect of dimensionality reduction and feature selection in text classification ". *IEEE conference ,2011, Page 21-24* , 2011.

[6] Mingming Zhou, Yabo Xu, John C. Nesbit, and Philip H. Winne *Sequential Pattern Analysis of Learning Logs* October, 2010.

[7] Kardi Teknomo's *Similaarity Measurments*, 2013

[8] Ashby, F. G. Multidimensional models of perception and cognition. Hillsdale, NJ: Erlbaum 1992

[9] Young, F. W. Hamer, R. M. Theory and applications of multidimensional scaling. *Hillsdale, NJ: Erlbaum*, 1994.

[10] Archit Jain. Discovering similarities and behavioral patterns of suspicious user profiles on social network. *Master Thesis*, TU Darmstad May 2018.

[11] Agrawal, R.; Imielinski, T.; Swami, A. Mining association rules between sets of items in large databases *Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD*, 93–207, 1993.

[12] PHILIPPE ESLING and CARLOS AGON, Time series data mining *Article in ACM Computing Surveys.*, November 2012

[13] Y. Yang, J. O. Pederson. A comparative study on feature selection in text categorization, *ACM SIGIR Conference*, 1995.

[14] C. J. van Rijsbergen Information Retrieval, Butterworths,1975

[15] Charu C. Aggarwal and ChengXiang Zhai A SURVEY OF TEXT CLUSTERING ALGORITHMS. *ICDE Conference*, (4):78–121, 2014.

[16] CHRISTOPHER ISSAL MAGNUS EBBESSON *Document Clustering : Master Thesis*. University of Gothenburg Department of Computer Science and Engineering GÃűteborg, Sweden, August, 2010.

[17] Ying Zhao and George Karypis *Evaluation of hierarchical clustering algorithms for document datasets*. Proceedings of the eleventh international conference on Information and knowledge management Pages 515-524 McLean, Virginia, USA November 04 - 09, 2002

[18] F.M. Dong, K.M. Koh, and K. LTeo. *Clustering Algorithms and Evaluations* Chapter 4. PHD Thesis, University of Stuttgart .

[19] Shubhendu Trivedi A Graph-Theoretic Clustering Algorithm based on the Regularity Lemma and Strategies to Exploit Clustering for Prediction *Master Thesis*, Worcester Polytechnic Institute 2012

[20] Kadim Tasdemir and ErzsÃľbet MerÃľnyi. *A Validity Index for Prototype-Based Clustering of Data Sets With Complex Cluster Structures*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) .

[21] Tina Geweniger, Marika Kastner, Mandy Lange, Thomas Villmann. *ModiĭňĄed Conn-Index for the evaluationof fuzzy clusterings*. ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012 .

[22] Harish, B S Guru, Devanur Shantharamu, Manjunath. (2010). *A review on classification of text documents.*.

[23] R.J.G.B. Campello . *A fuzzy extension of the Rand index and other related*

indexes for clustering and classification assessment. Pattern recognition Letters 28, 2007 .

[24] W. Dou, Y. Ren, Q. Wu, S. Ruan, Y. Chen, D. Bloyet, and J.-M. Constans . *Fuzzy kappa for the agreement measure of fuzzy classifications*. Neurocomputing, 2007.

[25] D. ZÃijhlke, T. Geweniger, U. Heimann, and T. Villmann . *Fuzzy Fleiss-Kappa for Comparison of Fuzzy Classifiers*. Proc. of European Symposium on Artificial Neural Networks (ESANN 2009), pp. 269-274, 2009.

[26] Charu C. Aggarwal ChengXiang Zhai . *A SURVEY OF TEXT CLUSTERING ALGORITHMS*.

[27] Barbara Rosario *Latent Semantic Indexing: An overview* . INFOSYS 240 Spring 2000 Final Paper .

[28] M.Girolami,AKaban. *On the Equivalance between PLSI and LDA,* . SIGIR Conference, pp. 433âĂŞ434, 2003.

[29] Steyvers, M. Griffiths, T. *(2007). Probabilistic topic models. In T. Landauer, D McNamara, S. Dennis, and W. Kintsch (eds), Latent Semantic Analysis: A Road to Meaning. Laurence Erlbaum*

[30] Strehl, A, Ghosh, J Mooney, RJ (July 2000), *Impact of similarity measures on web-page clustering,*. AAAI Workshop on AI for Web Search, pages 58-64.

Erklärung

Hiermit erkläre ich, dass ich meine Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die Arbeit noch nicht anderweitig für Prüfungszwecke vorgelegt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Mittweida, im July 2019