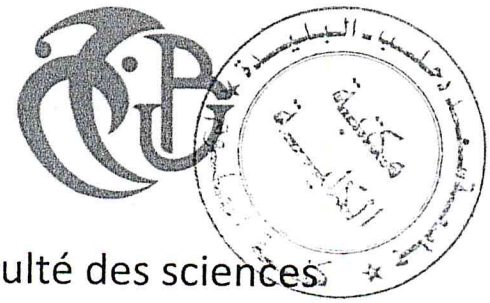
Université Saad DAHLAB de Blida



Faculté des sciences

Département d'informatique

Mémoire présenté par :

Ferdjouni zineddine

En vue d'obtenir le diplôme de Master

Sujet:

Combining link and content analysis for text clustering

Encadré par : Dr CHIKHI Nacim Fateh

2012 - 2013

Acknowledgements

First of all I thank Allah for helping me to finish this project.

I would like to express my gratitude to my supervisor, Dr. CHIKHI Nacim Fateh, who has been a great advisor and who has always helped, encouraged and oriented me during my project. This thesis could not have been finished without his efforts.

I also thank the jury members who kindly accepted to examine this thesis.

I deeply thank my mother, my father, my brothers and sisters for their love, care, support, encouragement and advice. Life would be hard without them.

I also express my sincere thanks to my friends Zaki, Younes, Yousef, Farouk, Khaled, Mustapha, Djamelle who help me to get out of stress and routine and for the great moments I have spent with them.

I especially thank my best friend Houssem Zwawi, first for being a very good friend, and second for being with me in every moment I went through and also for helping me to believe in myself when facing challenges.

I really would like to thank my classmates Wahab, Fateh, Yousef, Hamza, Salah, Nacer, Mohamed and Housin for the unforgettable moments that I had with them during my studies at university.

Lastly, I thank all the people who wished me good luck and all the persons who I have forgotten to thank.

Abstract:

In many applications huge amounts of textual data are generated continuously. The web is a typical example in which hundreds of thousands (if not millions) of articles are published every day. In order to facilitate the access to such huge document collections, researchers have developed various tools to organise them. Document clustering is one of these techniques which has recently become a very active area of research. Many document clustering algorithms have been developed such as PLSA (Probabilistic Latent Semantic Analysis) and NMF (Non-negative Matrix Factorization). These approaches however use only the textual content of documents and do not exploit other information such as the links between documents.

In this work we propose a new algorithm, the Multi-view Non-negative Matrix Factorization (MNMF), which is a hybrid algorithm for document clustering. MNMF takes into account not only the textual content of documents but also the link information. We show through experiments using real document collections the validity of the proposed approach.

Keywords:

Clustering (unsupervised classification), Text mining, Bibliometrics, Data mining, Cluster analysis, Multi-view NMF (MNFM).

ملخص

في العديد من التطبيقات تتولد كميات ضخمة من البيانات النصية بشكل مستمر. شبكة الإنترنت هي مثال نموذجي عن هذا و التي بدورها تنشر مئات الآلاف (إن لم يكن الملايين) من المقالات كل يوم و لتسهيل الوصول إلى هذه المجموعات الضخمة من الوثائق، طور الباحثون أدوات مختلفة لتنظيمها و تجميعها و من بين هذه التقنيات التي أصبحت مؤخرا محل الكثير من البحوث PLSA وNMF.

لكن هذه الأساليب تعتمد فقط على المحتوى النصي للوثائق وليس على استغلال المعلومات الأخرى مثل الروابط بين الوثائق.

في هذا العمل نقترح خوارزمية جديدة و التي تعتبر خوارزمية هجينة لتجميع الوثائق (MNMF)

و التي تأخذ بعين الاعتبار ليس فقط المحتوى النصي من الوثائق ولكن أيضا معلومات الارتباط. وتبين لنا من خلال التجارب باستخدام مجموعات الوثائق الحقيقية صلاحية النهج المقترح.

Contents

Introduction

In many applications huge amounts of textual data are generated continuously. The web is a typical example in which hundreds of thousands (if not millions) of articles are published every day. This is mainly due to the web 2.0 which allows any internet user to publish comments, reviews, articles, etc. on the web. Science is another example where a lot documents are generated through the conferences, books and journals that are published each year.

In order to facilitate the access to such huge document collections, researchers have developed various tools to organise them. Document clustering is one of these techniques which has recently become a very active area of research. Many document clustering algorithms have been developed such as PLSA (Probabilistic Latent Semantic Analysis) and NMF (Non-negative Matrix Factorization). These approaches however use only the textual content of documents and do not exploit other information such as the links between documents.

In this work we propose a new algorithm, the Multi-view Non-negative Matrix Factorization (MNMF), which is a hybrid algorithm for document clustering. MNMF takes into account not only the textual content of documents but also the link information.

We show through experiments using real document collections the validity of the proposed approach in the sense that MNMF, which combines link and content information, gives better clustering results than other approaches that are based only on textual content.

This thesis is organised as follows:

Chapter 1 is an introduction to the field of data clustering. Chapter 2 is an overview on document clustering where existing approaches are reviewed. In Chapter 3, we present our new algorithm MNMF. Experimental results using two collections of documents are presented in Chapter 4.

1

Introduction to clustering

1.1 Introduction:

1.1.1 What is Data Mining?

Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data sets. These tools can include statistical models, mathematical algorithms, and machine learning methods (algorithms that improve the performance automatically through experience, such as neural networks or decision trees). Consequently, data mining consists of more than collecting and managing data, it also includes analysis and prediction [Adriaans and Zantinge 96].

Data mining can be performed on data represented in quantitative, textual, or multimedia forms. Data mining applications can use a variety of techniques to examine the data. They include association (patterns where one event is connected to another event, such as purchasing a pen and purchasing paper), sequence or path analysis (patterns where one event leads to another event, such as the birth of a child and purchasing diapers), classification (identification of new patterns, such as coincidences between duct tape purchases and plastic sheeting purchases), clustering (finding and visually documenting groups of previously unknown facts, such as geographic location), and forecasting (discovering patterns from which one can make reasonable predictions regarding future activities, such as the prediction that people who join an athletic club may take exercise classes).

A number of advances in technology and business processes have contributed to a growing interest in data mining. Some of these changes include the growth of computer networks, which can be used to connect databases; the development of performance search-related techniques such as neural networks and advanced algorithms; the spread of the client/server computing model, allowing users to access centralized data resources from the desktop; and an increased ability to combine data from disparate sources into a single searchable source [Adriaans and Zantinge 96].

Data mining has become increasingly common in both the public and private sectors. Organizations use data mining as a tool to survey customer information, reduce fraud

and waste, and assist in medical research. However, the proliferation of data mining has raised some implementation and oversight issues as well. These include concerns about the quality of the data being analysed, the interoperability of the databases and software between agencies, and potential infringements on privacy.

1.1.2 Data mining uses:

Data mining is used for different purposes and in many applications. For example:

1.1.2.1 Data exploration:

The main use of data mining is to explore what classical analysis methods can't explore because of the important amount of data (satellite data, geometric, scientific simulation and Multidimensional data).

1.1.2.2 Improving productivity:

Data mining can play an important role to improve productivity by allowing good decisions to be taken by exploring the historical data to predict future events.

1.1.2.3 Exploiting the increase of computing power:

Recent advances in technology have allowed the development of computers that:

- Support great volumes of data.
- Execute efficiently the exploration process.
- Manipulate heterogeneous data.

1.1.3 Different types of data:

Data Mining can be applied on different types of data including:

- Relational data and transactional data.
- Spatial and temporal data, spatial-temporal observations.
- Text.
- Images, video.
- Mixtures of data.
- Sequence data.

1.1.4 Different types of attributes:

There are three types of data attributes:

- **Numerical:** The domain is ordered and can be represented using real numbers (e.g., age, income).
- **Nominal or categorical:** The domain is a finite set without any natural ordering (e.g., occupation, marital status, race).
- **Ordinal:** The domain is ordered, but absolute differences between values is unknown (e.g. Preference scale, severity of an injury).

1.1.5 Data mining Model:

A data mining model is a description of a specific aspect of a dataset. It produces output values for an assigned set of input values for example:

- Linear regression model.
- Classification model.
- Clustering.

A data mining model can be described in two levels:

Functional level: describes the model in terms of its intended usage.
Examples: Classification, clustering

Representational level: specific representation of the model.
Examples: classification tree, nearest neighbour method.

1.2 Knowledge Discovery from Data:

1.2.1 Introduction:

KDD is the organized process of identifying valid, novel, useful, and understandable patterns from large and complex data sets. *Data Mining* (DM) is the core of the KDD process, involving the inferring of algorithms that explore the data, develop a model

and discover previously unknown patterns. The model is used for analysis, prediction and to understand phenomena from the data [Maimon and Rokach 05].

The accessibility and abundance of data today make knowledge discovery and Data Mining a matter of considerable importance and necessity. Given the recent growth of the field, it is not surprising that a wide variety of methods is now available to researchers and practitioners.

1.2.2 KDD Process:

The knowledge discovery process is iterative and interactive, consisting of nine steps (Figure 1.1).

Note that this process is iterative at each step, meaning that moving back to previous steps may be required. The process has many "artistic" aspects in the sense that one cannot present one formula or make a complete taxonomy for the right choices for each step and application type. Thus it is required to understand the process and the different needs and possibilities in each step [Maimon and Rokach 00].

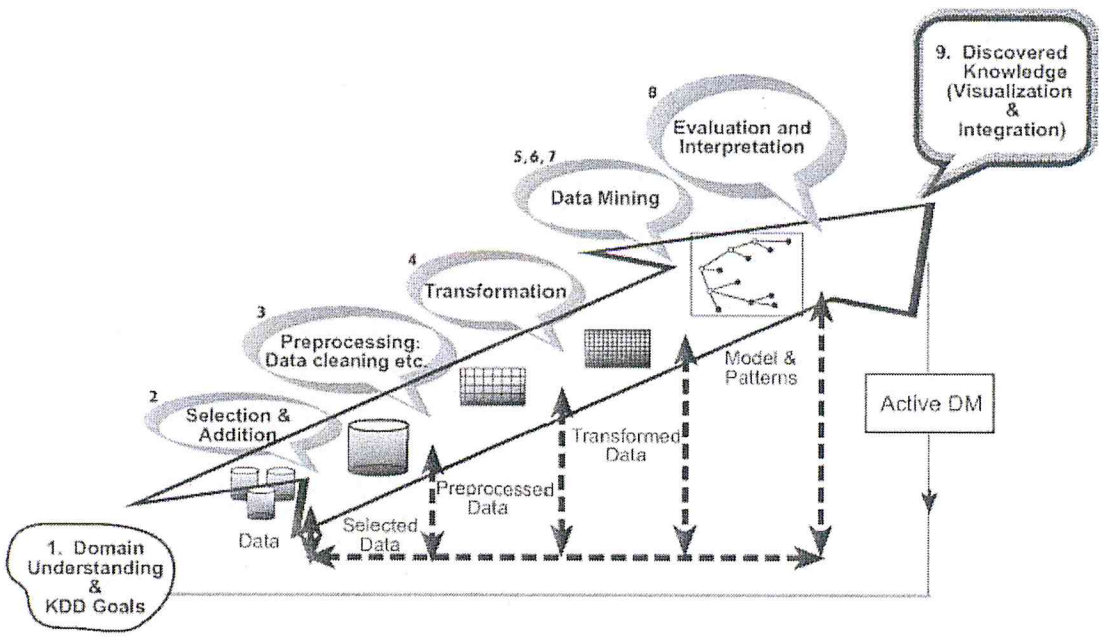


Figure 1.1 - The Process of Knowledge Discovery.

1.2.2.1 Domain understanding:

This is the initial preparatory step. It prepares the scene for understanding what should be done with many decisions (about transformation, algorithms, representation, etc.). The people who are in charge of a KDD project need to understand and define the goals of the end-user and the environment in which the knowledge discovery process will take place (including relevant prior knowledge). [Maimon and Rokach 00].

1.2.2.2 Data selection:

Once the goals defined, the data that will be used for knowledge discovery have to be determined. This includes finding out what data is available, obtaining additional necessary data, and then integrating all the data into one dataset, including the attributes that will be considered for the process. This step is very important because the data mining learns and discovers from the available data. This is the evidence base for constructing the models. If some important attributes are missing, then the entire study may fail. [Maimon and Rokach 00].

1.2.2.3 Pre-processing and cleaning:

In this stage, data reliability is enhanced. This includes data cleaning, such as handling missing values and removal of noise or outliers. This step makes possible the reduction of the sample set to be analysed. Two tasks can here be affected:

- **Reduction of the number of data objects:**

In the reduction of the number of data objects, data can be generalized attending to the defined domain's hierarchies or attributes with continuous values that can be transformed into discrete values attending to the defined classes.

- **Reduction of the number of attributes:**

The reduction of the number of attributes attempts to verify if any of the selected attributes can later be omitted.

1.2.2.4 Data transformation:

In this stage, the generation of better data for the data mining is prepared and developed. Methods here include dimensionality reduction (such as feature selection and extraction and record sampling), and attribute transformation (such as discretization of numerical attributes and functional transformation). This step which is usually very project-specific can be crucial for the success of the entire KDD project.

1.2.2.5 Data mining task selection:

This stage includes choosing which type of data mining task to use, for example, classification, or clustering. This mostly depends on the KDD goals, and also on the previous steps. There are two major goals in data mining: prediction and description. Prediction is often referred to as supervised data mining, while descriptive data mining includes the unsupervised and visualization aspects of data mining. Most data mining techniques are based on inductive learning, where a model is constructed explicitly or implicitly by generalizing from a sufficient number of training examples.

The underlying assumption of the inductive approach is that the trained model is applicable to future cases. The strategy also takes into account the level of meta-learning for the particular set of available data. [Maimon and Rokach 00].

1.2.2.6 Data mining algorithm selection:

This stage includes selecting the specific method to be used for searching patterns which can be classification or clustering. The difference between the two is that classification is assigns objects into pre-defined classes whereas in clustering the classes are to be defined. For example, in considering precision versus understandability, the former is better with neural networks, while the latter is better with decision trees. For each strategy of meta-learning there are several possibilities of how it can be accomplished. [Maimon and Rokach 00].

1.2.2.7 Use of the data mining algorithm:

Finally the implementation of the data mining algorithm is completed. In this step, we might need to apply the algorithm several times until a satisfying result is obtained, for instance by tuning the algorithm's control parameters, such as the minimum number of instances in a single leaf of a decision tree.

1.2.2.8 Model evaluation:

This stage includes the evaluation and the interpretation of the mined patterns (rules, reliability etc.), with respect to the goals defined in the first step.

1.2.2.9 Using the discovered knowledge:

In this stage, the discovered knowledge is incorporated into another system for further action. The knowledge becomes active in the sense that changes are made to the system and their effects are measured. Actually the success of this step determines the effectiveness of the entire KDD process.

There are many challenges in this step, such as losing the "laboratory conditions" under which the model has been developed. Data structures may change (certain attributes become unavailable), and the data domain may be modified (such as, an attribute may have a value that was not assumed before) [Maimon and Rokach 00].

1.3 Unsupervised classification (Clustering):

1.3.1 Definition:

Clustering is a multivariate statistical technique to automatically subdivide a set of objects into groups. The purpose is to make each group (or cluster) as homogeneous as possible in the sense that all objects in it have similar properties, while objects in different clusters should be as dissimilar as possible. In order to increase the efficiency in database systems, the number of disk accesses is to be minimized. In clustering, the objects of similar properties are placed in one class of objects and a single access to the disk makes the entire class available [Nicholas O. Andrews and Edward A. Fox 07].

In order to elaborate the concept a little bit, let's take the example of a library system. In a library, books dealing with a large variety of topics are available and are usually kept in form of clusters. The books that have some kind of similarity among them are placed in one cluster. For example, books on "databases" are kept in one shelf and books on "operating systems" are kept in another cupboard, and so on. To further reduce the complexity, the books that cover the same topics are placed in the same shelf. And then the shelf and the cupboards are labeled with a relative name. Now when a user wants a particular book on a specific topic, he or she would only have to go to that particular shelf and check for the book rather than checking in the entire library.

1.3.3 Definitions:

In the following, we give definitions of some frequent terms used in clustering:

1. 3.3.1 Cluster:

Is a group of instances of data (or objects), which have some common characteristics.

1.3.3.2 Distance between two clusters:

The distance between two clusters involves some or all elements of the two clusters. The clustering method determines how the distance should be computed [Bouguettaya 96].

1.3.3.3 Similarity:

A similarity measure SIM (O_i,O_j) can be used to represent the similarity between two objects i and j. Typical similarity generates values of '0' for objects exhibiting no agreement among their attributes, and '1' when perfect agreement is detected. Intermediate values are obtained for cases of partial agreement [Bouguettaya 96].

1.3.3.4 Threshold:

The lowest possible input value of similarity required to join two objects in one cluster.

1.3.3.5 Similarity Matrix:

Similarities between objects calculated by the function SIM (O_i, Oj) can be represented in the form of a matrix called a similarity matrix. [A.K. JAIN et al 99]

1.3.3.6 Dissimilarity Coefficient:

The dissimilarity coefficient of two clusters is defined to be the distance between them. The smaller its value, the more similar the two clusters are.

1.3.3.7 Cluster Seed:

The first object of a cluster is defined as the initiator of that cluster i.e. every incoming object's similarity is compared with the initiator. The initiator is called the cluster seed [A.K. JAIN et al 99].

1.3.4 Types of clustering methods:

There are many clustering methods available, and each of them may give a different grouping of a dataset. The choice of a particular method will depend on the type of the desired output, the known performance of the method with particular types of data, the hardware and software facilities available and the size of the dataset. In general, clustering methods may be divided into two categories based on the cluster structure which they produce. The non-hierarchical methods divide a dataset of N objects into M clusters, with or without overlap. These methods can be divided into partitioning methods, in which the classes are mutually exclusive, and the less common clumping method, in which overlap is allowed. Each object is a member of the cluster with which it is most similar; however the threshold of similarity has to be defined. Hierarchical methods produce a set of nested clusters in which each pair of objects or clusters is progressively nested in a larger cluster until only one cluster remains. Hierarchical methods can be further divided into agglomerative and divisive methods. In agglomerative methods, the hierarchy is built up in a series of N-1

agglomerations, or fusion, of pairs of objects, beginning with the un-clustered dataset. The less common divisive methods begin with all objects in a single cluster and at each of the N-1 steps divide some cluster into two smaller clusters, until each object resides in its own cluster.

1.3.4.1 Nonhierarchical methods:

1.3.4.1.1 Partitioning Methods:

Partitioning methods generally result in a set of M clusters where each object belongs to one cluster. Each cluster may be represented by a centroid or a cluster representative; this is some sort of summary describing the objects contained in a cluster. The precise form of this description will depend on the type of the objects to be clustered. In the case where real-valued data is available, the arithmetic mean of the attribute vectors for all objects within a cluster provides an appropriate representative; alternative types of centroid may be required in other cases, e.g., a cluster of documents can be represented by a list of keywords that occur in some minimum number of documents within a cluster. If the number of clusters is large, the centroids can be further clustered to produce a hierarchy [Bing Liu 07].

The single pass algorithm is a very simple partitioning algorithm which proceeds as follows:

1. Make the first object the centroid for the first cluster.
2. For the next object, calculate the similarity, S, with each existing cluster centroid, using some similarity coefficient.
3. If the highest calculated S is greater than some specified threshold value, add the object to the corresponding cluster and re-determine the centroid; otherwise, use the object to initiate a new cluster. If any objects remain to be clustered, return to step 2.

As its name implies, this method requires only one pass through the dataset; the time requirements are typically of order $O\ (N \log N)$ for order $O\ (\log N)$ clusters. This makes it a very efficient clustering method for a serial processor.

1.3.4.1.2 Hierarchical methods:

a. Hierarchical Agglomerative methods (bottom-up):

Hierarchical agglomerative clustering methods are the most commonly used. The construction of a hierarchical agglomerative classification can be achieved by the following general algorithm:

1. Find the 2 closest objects and merge them into a cluster;
2. Find and merge the next two closest points, where a point is either an individual object or a cluster of objects;
3. If more than one cluster remains, return to step 2.

Individual methods are characterized by the definition used for the identification of the closest pair of points, and by the means used to describe the new cluster when two clusters are merged.[Bing liu 07]

b. Divisive clustering (top-down):

This approach starts by assigning all data points to one cluster, the root. This root is then divided into a set of child clusters. Each child cluster is recursively divided further until only singleton clusters of individual data points remain, i.e., each cluster with only a single point.

1.3.5 Aggregation methods

1.3.5.1. Single Link Method (SLINK):

The single link method is probably the best known of the hierarchical methods. It operates by joining, at each step, the two most similar objects, which are not yet in the same cluster. The name single link thus refers to the joining of pairs of clusters by the single shortest link between them [Guillaume Cleuziou 04].

The single link algorithm is giving below:

1. Begin with the disjoint clustering having level L (0) = 0 and sequence number m = 0.
2. Find the least dissimilar pair of clusters in the current clustering, say pair (r), (s), according to

   d [(r), (s)] = min d [(i), (j)]

   where the minimum is over all pairs of clusters in the current clustering.
3. Increment the sequence number: m = m +1. Merge clusters (r) and (s) into a single cluster to form the next clustering m. Set the level of this clustering to

   L(m) = d [(r), (s)]
4. Update the proximity matrix, d, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted (r, s) and old cluster (k) is defined in this way:

   d[(k), (r, s)] = min d[(k),(r)], d[(k),(s)]
5. If all objects are in one cluster, stop.

   Else, go to step 2.

1.3.5.2 Complete Link Method (CLINK):

The complete link method is similar to the single link method except that it uses the least similar pair between two clusters to determine the inter-cluster similarity (so that every cluster member is more like the furthest member of its own cluster than the furthest item in any other cluster). This method is characterized by small, tightly bound clusters [Guillaume Cleuziou 04].

1.3.5.3 Group Average Method:

The group average method relies on the average value of the pairwise within a cluster, rather than the maximum or minimum similarity as with the single link or the complete link. Since all objects in a cluster contribute to the inter-cluster similarity, each object is on average more like every other member of its own cluster then the objects in any other cluster [Guillaume Cleuziou 04].

1.4 Evaluation of Cluster Quality:

For clustering, there are many measures of cluster "goodness" or quality. One type of measures allows comparing different sets of clusters without reference to external knowledge and is called an internal quality measure; an example is the "overall similarity" based on the pairwise similarity of objects in a cluster. The other type of measures helps to evaluate the clustering by comparing the groups produced by the clustering algorithms to known classes. This type of measures is called an external quality measure.

1.4.1 Internal measures:

This type of evaluation doesn't use external knowledge; it uses just input data as reference. Some measures are described below:

1.4.1.1 Inter-cluster:

This type of measures is based on a distance between clusters; to calculate this distance some methods can be used such us the simple link (SLINK), complete link (CLINK) or group average link. If the distance between clusters is high then the clustering algorithm can be considered as a good clustering.

1.4.1.2 Intra-cluster:

This type of measures is based on the distance between each cluster's centroid and its members. The objective function which is the sum of the internal distances is then calculated. The lower the value of the objective the better is the clustering.

1.4.1.3 Overall Similarity:

In the absence of any external information, such as class labels, the cohesiveness of clusters can be used as a measure of cluster similarity. One method for computing the cluster cohesiveness is to use the weighted similarity of the internal cluster similarity

$$\frac{1}{|s|^2}\sum_{\substack{d\in s\\ d'\in s}} cosin(d', d)$$

1.4.2 External measures:

This type of evaluation uses external knowledge as a measure of quality. We describe below some of these measures.

1.4.2.1 Entropy:

Let's consider that *CS* is a clustering solution. For each cluster, the class distribution of the data is calculated first, i.e., for cluster *j* compute p_{ij}, the "probability" that a member of cluster *j* belongs to class *i*. Then using this class distribution, the entropy of each cluster *j* is calculated using the standard formula:

$$E_j = \sum_i P_{ij} \log(P_{ij})$$

where the sum is taken over all classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster:

$$E_{cs} = \sum_{j=1}^{m} \frac{n_j * E_j}{n}$$

where n_j is the size of cluster *j*, *m* is the number of clusters, and *n* is the total number of data points.

1.4.2.2 F-measure:

It's a measure that combines the precision and recall ideas from information retrieval. Each cluster is treated as if it was the result of a query and each class as if it was the desired set of documents for a query. Then calculate the recall and precision of that cluster for each given class. For an entire hierarchical clustering the F measure of any class is the maximum value it attains at any node in the tree. An overall value for the F-measure is computed by taking the weighted average of all values for the F-measure as given by the following:

$$F = \sum_{i} {}^{n_i}/_{n} \max\{F(i,j)\}$$

where the max is taken over all clusters at all levels, and n is the number of documents.

1.5Applications:

Data clustering has many applications in different field such us computer science, biology…etc. Some of these applications include:

1.5.1 Similarity searching in Medical Image Database:

This is a major application of the clustering technique. In order to detect many diseases like Tumor, The scanned pictures or the x-rays are compared with the existing ones and the dissimilarities are recognized.

We have clusters of images of different parts of the body. For example, the images of the CT scan of brain are kept in one cluster. To further arrange things, the images in which the right side of the brain is damaged are kept in one cluster. The hierarchical clustering is used. The stored images have already been analyzed and a record is associated with each image. In this form a large database of images is maintained using the hierarchical clustering.

Now when a new query image comes, it is firstly recognized that what particular cluster this image belongs, and then by similarity matching with a healthy image of

that specific cluster the main damaged portion or the diseased portion is recognized. Then the image is sent to that specific cluster and matched with all the images in that particular cluster. Now the image, with which the query image has the most similarities, is retrieved and the record associated to that image is also associated to the query image. This means that now the disease of the query image has been detected.

Using this technique and some really precise methods for the pattern matching, diseases like really fine tumor can also be detected.

So by using clustering an enormous amount of time in finding the exact match from the database is reduced.

1.5.2 Business and marketing:

Clustering is used by market researchers to partition the general population of costumers into groups and to understand the relationship between these groups to better position products and to add new products.

1.5.3 Word wide web:

Clustering is used to group search results, files and web sites to make the research faster and to reorder the big volume of data.

1.5.4 Database segmentation:

The main purpose of database segmentation is to decrease the data ensemble after using some treatments; this technique can be useful in image segmentation which is the division of an image into heterogeneous areas.

1.6 Conclusion:

In this chapter, some basic concepts of clustering were introduced by first providing the definition of Data Mining, KDD and clustering. We also gave the definition of some related terms and an example to elaborate the clustering concept. We then

presented different approaches to data clustering and also discussed some existing algorithms in particular the partitioning and the hierarchical methods. At the end of the chapter, we listed some applications of data clustering.

2

State of the art on document clustering

2.1 Text clustering:

2.1.1 Introduction:

Document or text clustering is a subset of the large field of data mining. The clustering aims to discover naturel groupings of data. In the field of artificial intelligent, it is known as unsupervised machine learning.

Text clustering has many challenges among them is how to represent textual data which are unstructured and how to determine the features of a document that are considered to be discriminatory.

2.1.2 Text preparation:

2.1.2.1 Bag of words:

To operate a statistical analysis of textual data, we have to find a representation for the corpus (it's a set of documents written in natural language by an expert) to build a model. There are many techniques in text mining but the best is to associate each document to a vector; this hypothesis is known as the bag of words representation. Before applying any text mining technique, a pre-processing step involving the following operations is generally performed:

a. Filtering signs :

The punctuation between words is generally ignored. The problem is more complicated for dates and statistics; their appearance sometimes can't be neglected but the condition is to normalise them correctly, for example 11-09-2012, 11 September 2012 and 2012 9/11 have to be together on a same index.
[Soumen Chakrabatri 03]

b. Stop-words filtering:

It's used to remove linking words and text articulation (articles, conjunctions, prepositions) because their discriminating power for a document to another is week. Removing these words also helps to decrease the total amplitude of the band frequency and so the difficulty of storing and computing. Stop-words filtering is often based on a predefined list. [Soumen Chakrabatri 03]

c. Filtering hapax:

The "bag of words" model is related to the problem of high dimensionality; the vocabulary length is always superior to the number of documents. In this vocabulary, most of the words have a minimum number of occurrences. A common pre-processing operation consists in ignoring words that have just one occurrence in the corpus (called hapax) in order to minimize the size of the vocabulary. This will enhance processing speed make date less noisy. [Soumen Chakrabatri 03]

d. Stemming and lemmatization:

In some situations, it may be desirable to regroup some forms. For example the form of a word in singular and plural designate fundamentally the same concept, so they have to be detected like similar. The simple stemming technique makes simply a call to a heuristic to eliminate some suffixes ("s" letter at the end in case of singular/plural). Lemmatization is a more advanced linguistic technique which allows having stamps for each form (infinitive for verbs and masculine singular for nouns), however the result for those methods are not perfectly satisfying as there is the problem of homonyms. [Soumen Chakrabatri 03]

e. Grouping units:

The basic unit has been chosen is "the word" we talk in this case on the model of "unigram" to say that the dependence between occurrence is not considers directly it is also possible to work on units that the unit is bigger "bigrams" the length of vocabulary must be in the worse case n_{w^2} .this treatment allowed to extract more information related to the original text another way to regroup words is research terms in which they are include in a same "syntagram" for example (Eiffel is often associate to tour) . [Soumen Chakrabatri 03]

In the Vector Space Model (VSM), an entity such as a document is represented by a vector or point in a high-dimensional space. The dimensions constituting the vector space usually represent the set of all different words that can be found throughout a document collection, i.e., the vocabulary, lexicon or thesaurus (it's a set of term, based in a hierarchical structuration it contains a set of constructor, key words and relation) [Magnus Roselle 09].

2.1.3.2 Similarity of Documents:

The Term-Document Matrix can use the following notational conventions: Matrices are denoted by bold capital letters, A. Vectors are denoted by bold lowercase letters, b. Scalars are represented by lowercase italic letters, c. if we have a large collection of documents, and hence a large number of document vectors, it is convenient to organize the vectors into a matrix. The row vectors of the matrix correspond to terms (almost of time terms are words) and the column vectors correspond to documents (web pages, for example). This kind of matrix is called a term-document matrix.
In mathematics, a bag (also called a multi-set) is like a set, except that duplicates are allowed. For example, {a,a,b,c,c,c} is a bag containing a, b, and c. Order does not matter in bags and sets; the bags {a,a,b,c,c,c} and {c,a,c,b,a,c} are equivalent. We can represent the bag {a, a, b, c, c, c} with the vector x = (2, 1, 3) by stipulating that the first element of x is the frequency of a in the bag, the second element is the frequency of b in the bag, and the third element is the frequency of c. A set of bags can be represented as a matrix X, in which each column x_j corresponds to a bag, each row x_i corresponds to a unique member, and an element x_{ij} is the frequency of the i^{th} member in the j^{th} bag.
In a term-document matrix, a document vector represents the corresponding document as a bag of words. In information retrieval, the bag of words hypothesis is the ability of estimation of the relevance of documents to a query by representing the documents and the query as bags of words. That is, the frequencies of words in a document tend to indicate the relevance of the document to a query.

2.1.3.3 Similarity of relations: The Pair-Pattern Matrix:

In a pair-pattern matrix, row vectors correspond to pairs of words, such as mason: stone and carpenter: wood, and column vectors correspond to the patterns in which the pair's co- occur, such as "X works with Y". [Lin and Pantel 01] introduced the pair-pattern matrix for the purpose of measuring the semantic similarity of patterns; that is, the similarity of column vectors. Given a pattern such as "X solves Y", their algorithm was able to find similar patterns, such as "Y is solved by X", "Y is resolved in X", and "X resolves Y".

[Lin and Pantel 01] proposed the extended distributional hypothesis; those patterns that co-occur with similar pairs tend to have similar meanings. The patterns "X solves Y" and "Y is solved by X" tend to co-occur with similar X: Y pairs, which suggests that these patterns have similar meanings. Pattern similarity can be used to infer that one sentence is a paraphrase of another [Lin and Pantel 01].

[Turney et al 03] introduced the use of the pair-pattern matrix for measuring the semantic similarity of relations between word pairs; that is, the similarity of row vectors.

For example, the pairs mason: stone, carpenter: wood, potter: clay and glassblower: glass share the semantic relation artisan: material. In each case, the first member of the pair is an artisan who makes artifacts from the material that is the second member of the pair.

The pairs tend to co-occur in similar patterns, such as "the X used the Y to" and "the X shaped the Y into".

The latent relation hypothesis is that pairs of words that co-occur in similar patterns tend to have similar semantic relations [Turney 08]. Word pairs with similar row vectors in a pair-pattern matrix tend to have similar semantic relations. This is the inverse of the extended distributional hypothesis that patterns with similar column vectors in the pair-pattern matrix tend to have similar meanings.

2.1.3.4 Weighting:

The weighting of terms is a statically measure the principle is: the appearance of words in texts in naturel language signifies the importance of these words in that the only objective is the representation of the content of these words.
It is possible to calculate not just the frequency of term in the corpus also the number of document contain this term and o for that there are different methods [Magnus Roselle 09].

2.1.3.4.1 Weighting formulas:

a. Term Frequency (TF):

- A term which appear several time in a document is more important that a term which appear just one time.
- w_{ij}= the number of occurrences of a term t_i in a document d_j
- TF_{ij} =Frequency of Term t_i in a document d_j

$$TF_{ij} = \frac{w_{ij}}{|d_j|}$$

b. Inverse Document Frequency:

- A term which appear in few document is a best discriminant that a term which appear in all documents.
- df_i = the number of document having a term t_i.
- d =the number of document of the corpus.

$$IDF_i = log \frac{d}{df_i}$$

c. TF-IDF:

- That defines term frequency x inverse document frequency.
- Allows measuring the importance of a term in a document relative to the ensemble of documents.

- tf_{ij} = frequency of a term i in a document j.
- df_i = number of document contains the term.
- N=d = corpus number of document.

$$w_{i,j} = tf_{i,j} \times \log(\frac{n}{df_i})$$

d. Vector TF-IDF:

The basic idea is to represent documents in vectors and to measure the proximity between documents, this angle represent a semantic distance.
The principle is to code each element of the bag of words with a number called TF-IDF to give a mathematical aspect to texts documents.

The TF-IDF has two fundamentally limits:

- The first is that long documents have more strong length because they have a lot of words so the frequency of term is high.
- The second is that the dependence of term frequency is too important .if a word appear two times in a document d_j that doesn't mean that it has two times plus the importance just on a document d_K in which it appear just one time.

2.2 content based approaches: text mining

2.2.1 Definition:

is a knowledge -intensive process in which user interact with documents using analysis tools ,text mining purpose is to extract useful information from data source ,however in text mining data source are document and interesting patterns .
text mining derive much of its inspiration from the seminal research on data mining, so we find that text mining and data mining prove that many architectural levels are

the same ,both type of system rely on pre-processing, patterns discovery and presentation of elements such as visualization tool to enhance the browsing of answer set.

2.2.2Texte Mining Process:

Different text process are represented in Figure 2.1 [Grobelnik, M. and Mladenic, D 04]

text	Text pre-processing	Text transformation	Attribute selection	Data mining/pattern discovery	Interpretation/ evaluation
input	1	2	3	4	5

Figure 2.1- text mining process.

2.2.2.1text:

a. Document Clustering :

Large volume of textually data (Billions of documents) must be handled in an efficient manner so the solution is to use a document clustering and the most popular clustering document are K-means and Agglomerative hierarchical clustering.

b. Text characteristics:

There are a lot of characteristics of text:

- Several input mode: Texts are intended for different forms and different languages.
- Ambiguity: there are Word ambiguity and Sentence ambiguity.
- Noisy data: there are Erroneous data and misleading (intentionally) data
- Unstructured text: there is Chat room, normal speech
- High dimensionality (There is Tens of thousands of words (attributes) and only a very small percentage is used in a typical document).

2.2.2.2 Text pre-processing:

In this part there are some operations like:

- Text cleans up: like removing ads from web pages and normalise text.
- Tokenisation: Splitting up a string of characters into a set of tokens.
- Part of speech tagging: it is the process of marking up the words in a text with their corresponding parts of speech.
    - Rule based: Depends on grammatical rules.
    - Statistically based: Relies on different word order probabilities and that Needs a manually tagged corpus for machine learning.
- Semantic Structures: here there are two methods:
    - Full parsing: Produces a parse tree for a sentence.
    - Chunking with partial parsing: Produces syntactic constructs like Noun Phrases and Verb Groups for a sentence.

2.2.2.3 text transformation(attribute generation):

To represent documents there are two ways:

- Text Representation: Text document is represented by the words (features) it contains and their occurrences. And there are two main approaches of document representation "Bag of words". And Vector Space.
- Feature Selection: is to select just a subset of the features to represent a document

2.2.2.4 attribute selection:

To select attribute some technic are used:

- Further reduction of dimensionality: Learners have difficulty addressing tasks with high dimensionality.
- Irrelevant features: Not all features help

e.g., the existence of a noun in a news article is unlikely to help classify it as "politics" or "sport".

2.2.2.5 data mining / pattern discovery:

This is the important part and it is used to:

- Merges text mining process with the traditional Data Mining process.
- Use the Classical Data Mining technics on the structured database that resulted from the previous stages.
- This is a purely application-dependent stage.

2.2.2.6 Interpretation and evaluation:

In this part two operations can be used

Terminate: Results well-suited for application at hand.

Iterate: Results not satisfactory but significant.

2.2.3 Text mining Tools:

There is a plethora of software tools to help with the basic processes of text mining. language models and concordances; several different corpora (large collections, particular languages, etc.); dictionaries, lexical, and morphological resources; software modules for handling XML and SGML documents; and other relevant resources such as courses, mailing lists, people, and societies. It classifies software as freely downloadable and commercially available, with several intermediate categories.

One particular framework and development environment for text mining, called General Architecture for Text Engineering or GATE [Cunningham 02], aims to help users to develop, evaluate and deploy systems for what the authors term" language engineering." It provides support not just for standard text mining applications such as information extraction, but also for tasks such as building and annotating corpora, and evaluating the applications.

At the lowest level, GATE supports a variety of formats including XML, RTF, HTML, and SGML, email and plain text, converting them into a single unified model that also supports annotation.

There are three storage mechanisms: a relational database, a serialized Java object, and an XML based internal format; documents can be re-exported into their original

format with or without annotations. Text encoding is based on Unicode to provide support for multilingual data processing, so that systems developed with GATE can be ported to new languages with no additional overhead apart from the development of the resources needed for the specific language.

GATE includes a tokenizer and a sentence splitter. It incorporates a part of speech tagger and a gazetteer that includes lists of cities, organizations, days of the week, etc. It has a semantic tagger that applies hand-crafted rules written in a language in which patterns can be described and annotations created as a result. Patterns can be specified by giving a particular text string, or annotations that have previously been created by modules such as the tokenizer, gazetteer, or document format analysis. It also includes semantic modules that recognize relations between entities and detect co-reference. It contains tools for creating new language resources, and for evaluating the performance of text mining systems developed with GATE.

One application of GATE is a system for entity extraction of names that is capable of processing texts from widely different domains and genres. This has been used to perform recognition and tracking tasks of named, nominal and pronominal entities in several types of text. GATE has also been used to produce formal annotations about important events in a text commentary that accompanies football video program material.

2.3 Link-Based Document Clustering:

Link-based document analysis applies co-citation as a similarity measure for clustering. The typical goal is to discover subsets of large document collections that correspond to individual fields of study

Conceptually, documents form clusters if they share links between them, Collections of linked documents (e.g. through citations or hyperlinks) can be modelled as directed graphs, as in Figure 2.2. A graph from one document to another indicates a link from the first to the second. In a matrix formulation, a binary adjacency matrix is formed corresponding to the document link graph. Assume that adjacency matrix

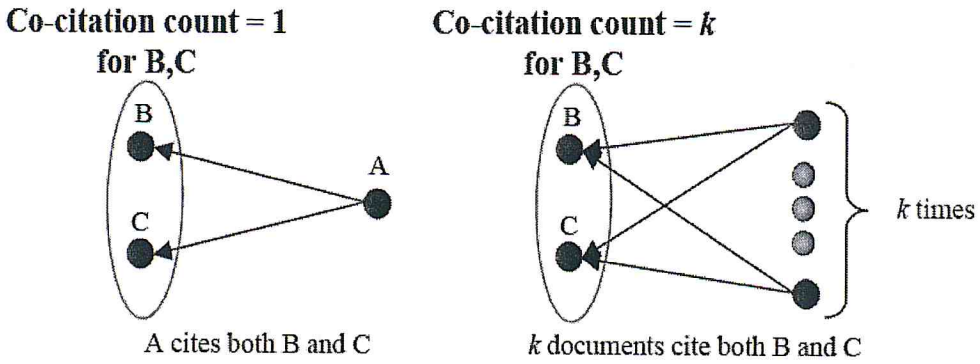


Figure 2.3 - Co-citation document similarity.

In terms of the document citation adjacency matrix A, co-citation count is a scalar quantity computed for pairs of matrix columns (cited documents).

For columns j and k, co-citation count $c_{j,k}$ is then

$$c_{j,k} = \sum_i a_{i,j}\ a_{i,k} = a_j\ a_k = A^T A$$

Here a_j and a_k are column vectors of A, i indexes rows, A^T is the transpose of A, note that the product $a_{i,j}\ a_{j,k}$ represents single co-citation occurrences, which the summation counts.

The co-citation count $c_{j,j}$ of a document with itself is simply a citation count, i.e. the number of times the document has been cited.

2.3.1 Clustering Algorithms for Citation Analysis:

This part, can include a discuss about how *k*-means and EM clustering can be applied to citation analysis. [Frizo JANSSENS 07].

2.3.1.1 Clustering by k-Means:

The well-known *k*-means algorithm starts with *k* random mean vectors and then, in turns, assigns each instance to the cluster with nearest mean vector and re-calculates

distinct values, it is governed by a cluster-specific distribution $\theta_i\ (x_k)$ references are drawn without replacement as there can be at most one link between each pair of papers. The distribution of n random variables with $|V|$ values, drawn without replacement, is governed by the multi-hyper geometric distribution.

The multi-hyper-geometric distribution is the generalization of the hyper geometric distribution for non-binary variables. Unfortunately, it is computationally infeasible because calculation of probabilities requires summation over a huge trellis and even a lookup-table is impractically large. Since the number of links in a paper is much smaller than the number of papers in V, it can be approximated by the multinomial distribution. This corresponds to drawing citations with replacement. The likelihood in the multinomial citation model is given in Equation 1. The "$n!$" term reflects that there are $n!$ Ways of drawing any given set of n citations in distinct orderings.

$$P_\theta\ (x_j|c_i\ ) = \prod_{xk\epsilon V} p(n)n!\ \theta_i\ (x_k)^{x_{jk}} \qquad (1)$$

Again, $x_j = x_j^{in}$ for co-citation and $x_j = x_j^{out}$ for bibliographic coupling.

The E and M steps for the multinomial model are given in Equations 2, 4, and 5 (posterior and maximum likelihood estimator for the multinomial distribution are well-known). As we see in Equation 3, it is not necessary to know $P\ (n)$ if only the posterior $P_\theta\ (x_j|c_i\ )$ is of interest. We can apply Laplace smoothing by adding one to all frequency counts.

$$\underline{\text{E step}}:\ P_\theta\ (c_i|x_j) = \frac{\pi_i P_\theta\ (x_j|c_i)}{\sum_k \pi_k p_\theta (x_j|c_k)} = \frac{\pi_i \prod_{xl\epsilon V} P(n)n!\theta_i(x_l)^{x_{jl}}}{\sum_k \pi_k \prod_{xl\epsilon V} p(n)n!\ \theta_k(xl)^{x_{jl}}} \quad (2)$$

$$= \frac{\pi_i \prod_{xl\epsilon V} \theta_i(xl)^{x_{jl}}}{\sum_k \pi_k \prod_{xl\epsilon V} \theta_k(xl)^{x_{jl}}} \qquad (3)$$

$$\underline{\text{M step}}:\ \theta_i\ (x_k) = \frac{\sum_{xl\epsilon X} x_{lk} p(c_i|x_l,\theta)}{\sum_{j\epsilon V} \sum_{xl\epsilon X} x_{lj} p(c_i|x_l,\theta)} \quad (4)$$

$$\pi_i = \frac{1}{|x|} \sum_{xk\epsilon X} P_\theta(c_i|x_k) \quad (5)$$

The multinomial distribution is also frequently used as a model for text. In the multinomial text model, words are drawn with replacement according to a cluster-specific distribution $\theta_i\ (x_k)$ the likelihood of a document $x_j = x_j^{txt}$ in cluster c_i can be characterized analogously to Equation 1; the E and M steps for the multinomial text model follow Equations 3 and 5, respectively (with $x = x^{txt}$) [Frizo JANSSENS 07].

2.4 Combining links and contents based clustering:

In this part a study which explains the need of bibliometrics and contents each other to enhance clustering for scientific topics will be discussed.

In a study by Glenisson, Glanzel, and Persson, full-text analysis and traditional bibliometric methods were serially combined to improve the efficiency of the individual methods. This methodology was applied to a special issue of Scientometrics. The study was based on 19 selected papers that were assigned to five categories. The outcomes have shown that such hybrid methodology can be applied to both research evaluation and information retrieval. The bibliometric part of the pilot study was restricted to simple statistical functions obtained from the papers reference lists, particularly the mean reference age and the share of references to serial literature. Because of the limited number of papers underlying the study, it has to be considered a pilot study that was further extended and confirmed by Glenisson et al. relevant results of this manuscript are discussed by the next. The number of papers under study was increased to the complete publication year 2003, in the journal Scientometrics, comprising 85 research articles and notes. This data set covered a broader and more heterogeneous spectrum of bibliometrics and related research.

In Figure 2.5 documents are clustered under consideration with a hierarchical method and compare these results with expert category assignments as well as with a bibliometric analysis. For interpretation purposes, top-scoring terms from each cluster are presented in term networks [Frizo JANSSENS 07].

Abbreviation	Description	Share (%)
A	advances in scientometrics	31.8
E	Empirical papers/case studies	34.1
M	Mathematical models	2.4
P	Political issues	17.6
S	Sociological approaches	3.5
I	Informatics/Webometrics	10.6

Figure 2.4 - Category of scientometrics papers and their distribution over categories.

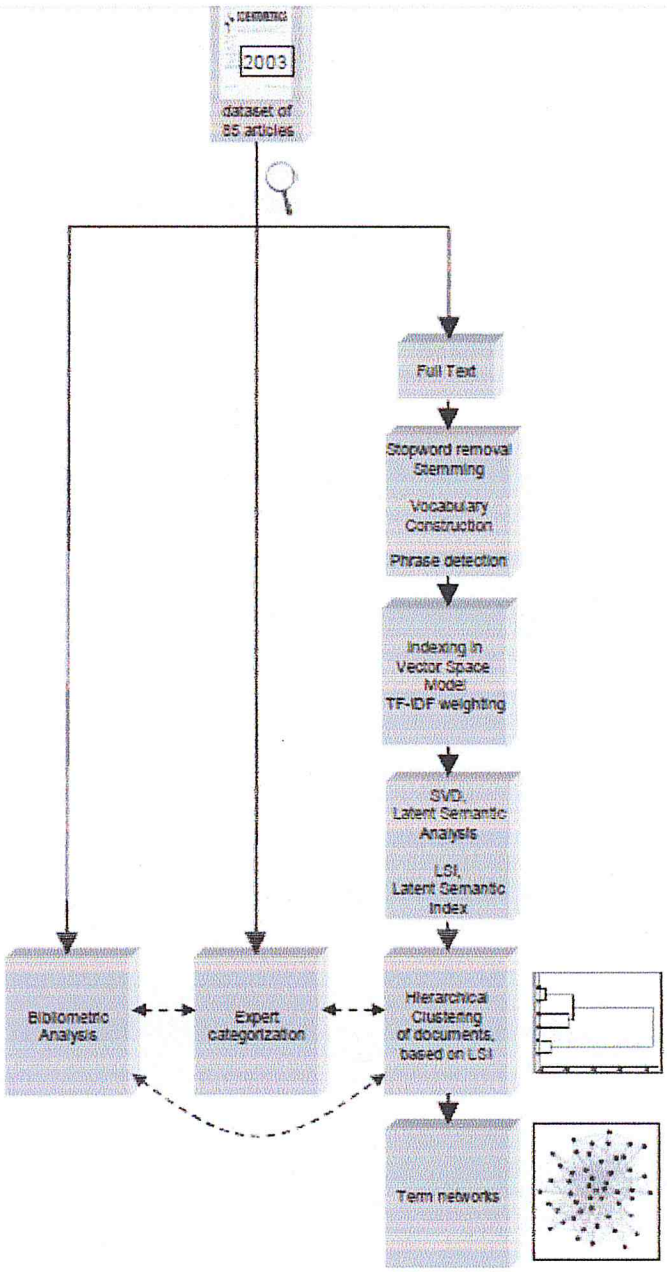


Figure 2.5- Overview of the analysis of a set of 85 articles and notes published in Scientometrics in 2003.

2.4.1 Serial combination of text-based clustering and bibliometrics:

The statistical analysis of the full texts provided a relational chart of the structure represented by the documents under study, the mean reference age (MRA) and the share of serials in all references can be used to characterize fields and sub disciplines

in the sciences and social sciences. In what follows checking whether these indicators can be used to characterize the six clusters found by the statistical full-text analysis (using (Latent Semantic Indexing (LSI)) which is a mathematical technic based on the truncated Singular Value Decomposition SVD (LSI uses the truncated SVD to approximate a term-document matrix A with a matrix A_K of lower rank k)) of a matrix which analyses the term-by document matrix A in order to find the major associative patterns of word usage in the document collection and to get rid of the 'noise variability' in It) To reduce the rank of the term-document matrix into (6).

First combining the bibliometric approach with the full-text analysis by means of aggregating both results: Figure 2.4 shows the relation between mean reference age and share of serials with the cluster results as overlay. Clusters are named by the title of their medoids (i.e., representative elements). In the example, Cluster 2 (indicated by its medoid Changing trends in publishing behaviour), is characterized by a medium MRA. The two special issues (Triple Helix Conference and S&T Indicators Conference) are indicated by ellipses.

These issues form surprisingly homogeneous groups, although, in general, there is not much correspondence between text-based cluster membership and common bibliometric characteristics. Papers with similar content might thus have different bibliometric characteristics depending on target readership and field of application. Therefore it deemed as an interesting option to integrate these two disparate information sources earlier in the segmentation process. [Frizo JANSSENS 07].

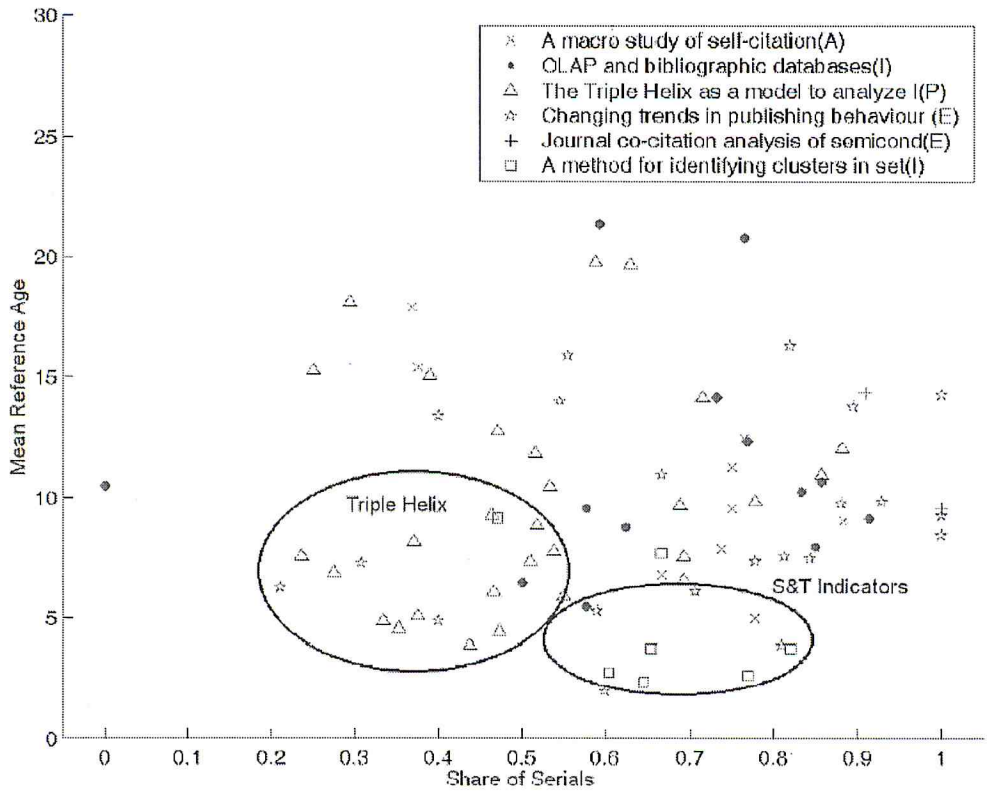


Figure 2.6 - Plot of mean reference age vs. share of serials for the documents in different text based clusters.

2.4.2 Integrating text and bibliometric information:

This part aims at devising a methodology for deeply combining text mining and bibliometrics by integrating text-based and bibliometric information early in the mapping process. More specifically, multiple information sources are incorporated before the clustering algorithm is applied.

Integrating textual content and a citation present in data sets containing bioinformatics and LIS (Library and Information Science) publications.

Then investigate how clustering performances of linear combinations, of Fisher inverse chi-square method, and of other integration systems. [Frizo JANSSENS 07].

For each data source, such as a normalized term-document matrix "A" or a normalized cited references-by-document matrix "B", square distance matrices D_t and D_{bc} can be constructed as follows:

$$D_t = O_N - A^T.A$$

$$D_{bc} = O_N - B^T.B$$

With N the number of documents and O_N a square matrix of dimensionality N with all ones. bc Refers to bibliographic coupling.

2.4.2.1 Weighted linear combination of distance matrices:

The distance matrices D_t and D_{bc} can be combined into an integrated distance Matrix D_i by a weighted linear combination as follows:

$$D_i = \alpha.D_t + (1-\alpha)\,D_{bc}$$

The resulting D_i can then be used in clustering algorithms.

2.4.2.2 Fisher's inverse chi-square method:

As a plain linear combination might not be the best solution for integrating textual and bibliometric information, a methodology based on Fisher's inverse chi-square method was developed. Fisher's inverse chi-square is an omnibus statistic from statistical meta-analysis to combine p-values from multiple sources.

In contrast to the weighted linear combination procedure, this method can handle distances stemming from different metrics with different distributional characteristics and avoids domination of any specific information source.

"Glenisson" has proposed this method as a means to integrate distances stemming from both text and gene expression data. The method is described in more detail and the rescaling of distances is improved by calculating p-values with respect to randomized data sets. This randomization is a necessary condition for having valid p-values.

The next schema (Figure 2.7) explains more:

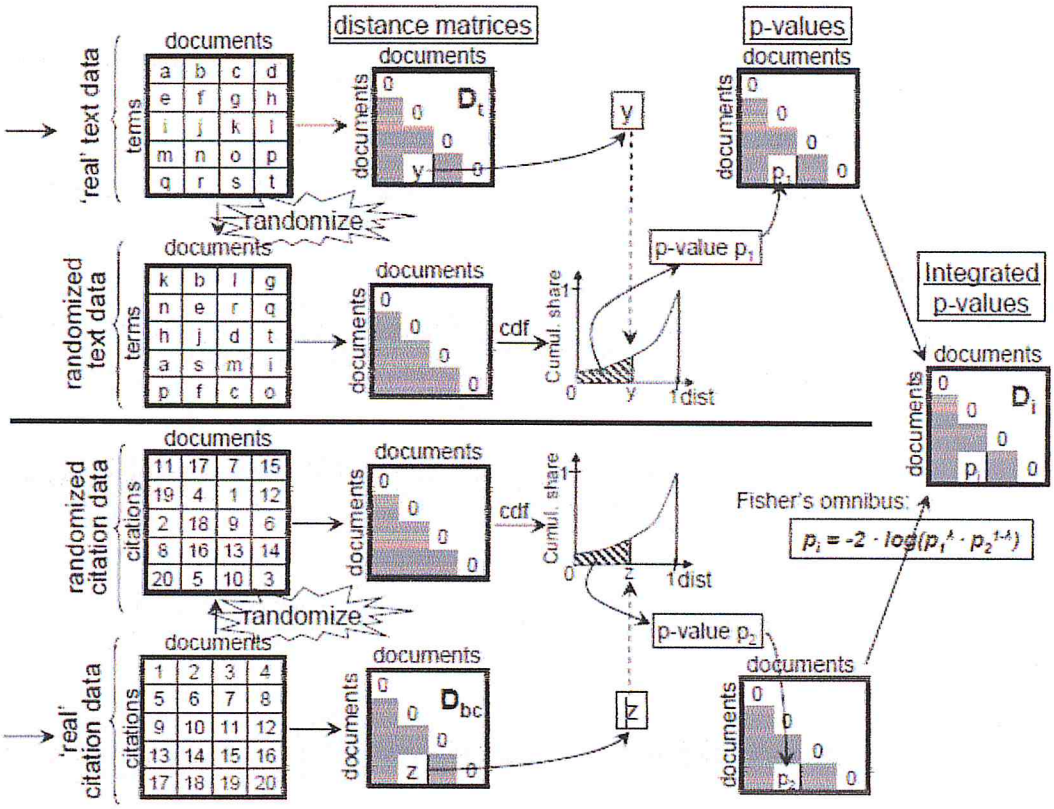


Figure 2.7 - Distance integration by using Fisher's inverse chi-square method.

2.4.2.3 Integrated Random Indexing:

Random Indexing (RI) can also be used to index citations in bibliographies besides words in texts. Furthermore, the method can even be modified to obtain an integrated random index containing both textual and citation information. As an aside, a weighted linear combination could also be used to integrate mutual similarities stemming from two distinct random indices, one based on text and one based on citations it would necessitate storing two random indices in memory.

The RI with textual information from the LIS data set provided good clustering performance, and that a RI with citation information even did slightly better than bibliographic coupling (probably due to the incorporation of context). Since these two experiments provided promising results, namely that the bag-of-concepts approach of RI seemed to work on the LIS data set, RI can be modified for data integration. To the best of the knowledge, RI has not been used for data integration

before, context vectors can also be constructed for all citations in the data set, and an integrated bag-of-concepts representation of a document can then be built by adding all (weighted) context vectors of all terms and of all cited references occurring in the document. If the integrated Hybrid analysis combining text mining and bibliometrics RI would be constructed without weighting the relative contributions of words and citations, again one of the data sources could dominate the other, especially because a term-document matrix A is usually much denser than the matrix B which indexes cited references in each document. Therefore, the following weight α was proposed to boost the contribution of citation context vectors to the final bag-of concepts representation of a document, relative to word context vectors. It is based on the relative sparseness of both A and B.

$$\alpha = \frac{\frac{\sum_i \sum_j A_{i,j}}{t.d}}{\frac{\sum_i \sum_j B_{i,j}}{r.d}} = \frac{r \sum_i \sum_j A_{i,j}}{t \sum_i \sum_j A_{i,j}}$$

With d the number of documents, t the term dimension of A, and r the reference dimension of B [Frizo JANSSENS 07].

Conclusion:

In this chapter contain we study existing approaches for text clustering which are divided into three families content based approaches, link base approaches and hybrid approaches which use the two link and content information after that we have seen techniques to combine link and content which are simple so it needs to be improved.

3

Multi-view NMF

3.1 Introduction:

Document clustering has been widely used as a fundamental and effective tool for efficient document organization, summarization and retrieval of large number of documents, among clustering method K-means algorithm has been the most used.

A recent algorithm developed for document clustering is the non-negative matrix factorisation (NMF). Basically, the NMF aims to decompose a non-negative matrix into a product of two non-negative matrices which is a good approximation of the original matrix.

The initial work on NMF by [Lee and Seung 1999 – 2001] has shown that the NMF factors contain coherent parts of the original data. Later works by [Xu et al. 2003], [Pauca et al. 2004] showed the usefulness of NMF for document clustering. In a more recent study, [Ding et al. 2005] show the equivalence between K-means and NMF.

In this chapter, we first study the NMF algorithm then we present a new approach to document clustering. The proposed algorithm, multi-view non-negative matrix factorisation (MNMF), takes as input a term-document matrix and an adjacency matrix and performs a joint factorization of these two matrices. We also discuss some aspects regarding our approach such as the initialization step of MNMF.

3.2 Matrix properties:

We first give some properties that will be used in the rest of this chapter.

3.2.1 Matrix trace:

The trace of a square matrix M is defined as:

$$\mathrm{Tr}\,(\mathrm{M}) = \sum_i m_{ii}$$

That means the sum of its diagonal elements.

The following are some properties of matrix trace:

$$Tr(A+B) = Tr(A) + Tr(B)$$

$$Tr(\alpha A) = \alpha\, Tr(A)$$

$$Tr(AB) = Tr(BA)$$

$$Tr(A) = Tr(A^T)$$

$$Tr(AB^T) = Tr(A^TB) = Tr(B^TA) = Tr(BA^T)$$

3.2.2 Lagrange multipliers:

Lagrange multipliers are a method that helps to determine stationary points (max or min) of a function with several variables; this method is useful to solve optimisation problems under constraints.

Example: this example shows how to use Lagrange multipliers.
The goal is to determine the radius R and the height H of a cylinder which minimize the surface S. The problem can be written as follows:

$$S(R, H) = 2\pi R(R+H) \qquad \text{the function to minimize.}$$

$$g(R, H) = \pi R^2 (H-V) \qquad \text{the constraint.}$$

We have here only one constraint so there is one Lagrange multiplier; the Lagrangian function is thus:

$$L_{exp}(R, H, \lambda) = S(R, H) + \lambda g(R, H)$$

Then the problem to minimize becomes:

$$L_{exp}(R, H, \lambda) = 2\pi RH + 2\pi R^2 + \lambda(\pi R^2 (H-V))$$

So the goal of Lagrange multipliers is to solve an equation with constraints by transforming it to an equation with no constraint or with a simple one.

3.2.3 Karush Kuhn Tucker conditions:

The KKT conditions were originally named after "Harold Kuhn" and "Albert Tucker" who first published the conditions in 1951. Researchers have later discovered that the necessary conditions for this problem had been stated by "William Karush" in his master's thesis in 1939 [Kuhn-tucker conditions 04].

Example: this example shows where and when to use the KKT conditions.

Suppose we have a function f, which we wish to maximise, together with some constraints, $g_i \leq c_i$, which must be satisfied:

Max $f(x_1, x_2) = 4\,x_1 + 3\,x_2$

subject to $g(x_1, x_2) = 2\,x_1 + x_2 \leq 10$ and $x_1 , x_2 \geq 0$

First, we write the Lagrangian function:

$$L_{KKT} = 4\,x_1 + 3\,x_2 + \lambda\,(10 - 2\,x1 - x2\,)$$

The Kuhn-Tucker conditions, which are necessary for a point to be a maximum are:

$\frac{\partial L}{\partial x_i} \leq 0$	$x_i \geq 0$	$x_i \frac{\partial L}{\partial x_i} = 0$
$g(x) \leq c_i$	$\lambda_j \geq 0$	$\lambda_j\,(c - g(x)) = 0$

By adding the necessary KKT conditions for maxima we obtain:

$L_{KKT_{x1}} = 4 - 2\lambda \leq 0$	$x_1 \geq 0$	$x_1\ (4 - 2\lambda) = 0$
$L_{KKT_{x2}} = 3 - \lambda \leq 0$	$x_2 \geq 0$	$x_2\ (3 - \lambda\ ) = 0$
$2\,x_1 + x_2 - 10 \leq 0$	$\lambda \geq 0$	$\lambda\ (2\,x_1 + x_2 - 10\ )$

After solving this set of inequalities and equations, points which may be maxima will be found.

3.3 Non-negative matrix factorization (NMF):

3.3.1 Definition

The NMF algorithm is used to find a factorized $n\times k$ W matrix and $k\times m$ H matrix where W = $[w_{ik}]$ and H = $[h_{kj}]$ for the given $n\times m$ V matrix where V= $[v_{ij}]$ such that: V≈WH. This factorization is done by minimizing the following objective function F:

$$F_{NMF} = ||V - WH||_F^2 \quad (1)$$

Where $||.||_F$ denotes the *Frobenius norm.* and W and H are positive. [Lee and Seung 1999 – 2001]

3.3.2 NMF algorithms

The main steps of the NMF algorithm are given on figure 3.1. Matrices H and W are first initialized and then updated in an iterative way.

NMF Algorithm

Input: matrix V

Output: matrices W and H (such that V≈WH)

Step 1: Algorithm initialisation

- initialise W
- initialise H

Step 2: Iterative update

while (not converged)

- update W
- update H

end while

Figure 3.1- NMF algorithm steps.

There exist different algorithms to solve the NMF problem; they differ in the way they update matrices H and W. The three most used techniques include: multiplicative update rules algorithms, gradient descent algorithms, and alternating least squares (ALS) algorithms.

3.3.2.1Multiplicative updates rules:

In this version, matrices W and H are updated using the following multiplicative update rules:

$$W \leftarrow W \frac{VH^T}{WHH^T}$$

$$H \leftarrow H \frac{VW}{HWW^T}$$

Update rules derivation detail:

This algorithm achieves a local minimum. The objective function F_{NMF} in equation (1) can be rewritten as:

$$F_{NMF} = Tr((V-WH)(V-WH)^T)$$
$$= Tr(VV^T) - 2\,Tr(VWH^T) + (WHH^T W^T)$$

Where X^T represents the transpose of X.

By adding the Lagrange multipliers $\alpha = [\alpha_{ik}]$ and $\beta = [\beta_{kj}]$ for the constraints $w_{ij} \geq 0$ and $h_{ij} \geq 0$ respectively, we obtain the following Lagrangian function:

$$L_{NMF} = Tr(VV^T) - 2\,Tr(VWH^T) + (WHH^T W^T) + Tr(\alpha W^T) + Tr(\beta H^T)$$

The partial derivatives of L_{NMF} with respect to W and H are:

$$\frac{\partial L_{NMF}}{\partial W} = -2\ VH^T + 2\ WHH^T + \alpha$$

$$\frac{\partial L_{NMF}}{\partial H} = -2\ VW + 2\ HWW^T + \beta$$

After using the KKT conditions $\alpha_{ij} w_{ij} = 0$ and $\beta_{ij} h_{ij} = 0$, we obtain:

$$-(VH^T)_{ij}\ w_{ij} + (WHH^T)_{ij} w_{ij} = 0$$
$$-(VW)_{ij}\ H_{ij} + (HWW^T)_{ij} H_{ij} = 0$$

These equations lead to the following update rules:

$$w_{ij} \leftarrow w_{ij} \frac{VH^T}{WHH^T + 10^{-9}}$$

$$h_{ij} \leftarrow h_{ij} \frac{VW}{HWW^T + 10^{-9}}$$

Note: adding 10^{-9} avoids division by 0.

3.3.2.2 Gradient descent algorithms:

Another family of NMF algorithms is based on gradient descent algorithms. Algorithms of this class repeatedly apply update rules of the form shown below:

$$H \leftarrow H - \varepsilon H \frac{\partial F}{\partial H}$$

$$W \leftarrow W - \varepsilon W \frac{\partial F}{\partial W}$$

The step-size parameters εH and εW vary depending on the algorithm; the partial derivation is the same as in the multiplicative update rules algorithm (3.1.2) [Michael W. Berry 06].

These algorithms always take a step in the direction of negative gradient. The trick comes on choosing the value of εH and εW: some algorithms initialize this step-values by 1, then multiply them in each subsequent iterations by 0.5 [Hoyer 04].

This is simple but not ideal because there is no restriction that keeps the elements of the updated matrices W and H from becoming negative. A common technique employed by many gradient descent algorithms is a simple projection step [Shahnaz et al. 06]; [Hoyer 04];[Chu et al. 04]; [Pauca et al. 06]. That is after each update rule the update matrices are projected to the non-negative value by setting the entire negative element to the nearest non-negative value, 0.

3.3.2.3 Alternating Least Squares (ALS):

ALS algorithms represent another solution to the NMF problem. In these algorithms a least squares step is followed by another least squares step in an alternative way; ALS algorithms exploit the fact that, while the optimisation problem is not convex in both W and H, given one matrix the other can be found with a simple least squares computation. At each iteration, the ALS NMF algorithm updates W and H as follows:

(LS) Solve for **H** in matrix equation $\mathbf{W}^T\mathbf{WH} = \mathbf{W}^T\mathbf{A}$.

(NONNEG) Set all negative elements in **H** to 0.

(LS) Solve for **W** in matrix equation $\mathbf{HH}^T\mathbf{W}^T = \mathbf{HA}^T$.

(NONNEG) Set all negative elements in **W** to 0.

In the above steps, a simple method was included for insuring non-negativity: the projection step, which sets all negative elements resulting from the least squares

computation to 0. This simple technique also has a few added benefits; it allows iterating some additional flexibility which is not available in other algorithms, especially those of the multiplicative update class. One drawback of the multiplicative algorithms is that once an element in W or H becomes 0, it must remain 0. This locking of 0 elements is restrictive, meaning that once the algorithm starts heading down a path towards a fixed point, even if it is a poor fixed point, it must continue in that vein. The ALS algorithms are more flexible, allowing the iterative process to escape from a poor path [Michael W. Berry 06].

3.3.2.4 Illustrative example:

We give in this section a toy example to illustrate the NMF principle and how clustering is performed using this algorithm.

Let V a matrix representing the weights and the heights of 6 persons which we want to cluster into two groups (k=2), and W, H two matrices initialised with random values.

V =

100	190
75	165
110	190
95	185
60	155
65	160

W =

0.6268	0.6702
0.5503	0.5328
0.9379	0.2240
0.7457	0.4358
0.0490	1.1853
0.3315	0.8017

H =

88.4827	181.8255
59.5292	116.8685

Applying NMF on V until convergence, we obtain the following factorization:

H =	W =
103.3677 170.3584 44.8176 127.5644	0.7142 0.5431 0.4511 0.6828 0.9716 0.1955 0.7190 0.4856 0.0646 1.1359 0.2532 0.9097

Where w_{ik} represents the membership degree of person i in cluster k, and h_{kj} represents the degree of importance of feature j in cluster k. From matrix W, we can get the membership of each person by assigning each one to the cluster in which the membership degree is maximal. Using this principle, the first cluster contains persons 1, 3 and 4, and the second one contains persons 2, 5 and 6.

3.4 Multi-view NMF (MNMF):

NMF has been shown to be a useful technique in a wide range of applications, such as image processing, text mining .... The basic NMF is however not adapted to the analysis of data represented using different information sources. For example, documents contain not only words but also other information such as links (citations or hyperlinks), author information, etc.

In this work, we propose a new document clustering algorithm which extends the classical NMF. Our approach, Multi-view NMF (MNMF), takes into account simultaneously link and content information to enhance clustering results.

3.4.1 Algorithm description:

MNMF performs a joint-factorization of the word-document matrix and the adjacency matrix. Basically, given two matrices C and L, MNMF finds three matrices W, H and R such that:

$$C \approx WH \text{ and } L \approx WR$$

To achieve this decomposition, MNMF minimizes the following objective function:

$$F_{MNMF} = \alpha \left(||C - WH||_F^2\right) + (1-\alpha)\left(||L - WR||_F^2\right) \quad (2)$$

where $||.||_F$ denotes the *Frobenius norm*, and $\alpha \in [0,1]$ is a real which controls the importance given to links and contents:

- if $\alpha > 0.5$ the algorithm gives more importance to textual content.
- if $\alpha < 0.5$ the algorithm gives more importance to links.
- if $\alpha = 0.5$ the algorithm treats equally link and content information.

The main steps of our algorithm are given below:

Multi-view NMF Algorithm (MNMF)
Input: matrices L $\in \mathbb{R}^{n \times m}$ and C $\in \mathbb{R}^{n \times s}$, number of clusters k **Output:** matrices W $\in \mathbb{R}^{n \times k}$, H $\in \mathbb{R}^{k \times m}$ and R $\in \mathbb{R}^{k \times s}$ **Step 1:** Algorithm initialisation - initialise W - initialise H - initialise R

Step 2: Iterative update

while (not converged)

- update W
- update H
- update R

end while

3.4.2 Multiplicative update rules:

We describe here how matrices W, H and R are updated at each step of the algorithm.

Equation (2) can be written as:

$$F_{lc} = \alpha \, Tr((C-WH)(C-WH)^T) + (1-\alpha) \, Tr((L-WR)(L-WR)^T)$$
$$= \alpha \, Tr(CC^T) - 2\alpha \, Tr(CWH^T) + \alpha(WHH^T W^T) + (1-\alpha) \, Tr(L L^T)$$
$$-2(1-\alpha) \, Tr(L WR^T) + (1-\alpha)(WR R^T W^T)$$

The second step of derivation uses the matrix trace properties; X^T represents the transpose matrix of X.

By adding the Lagrange multipliers α_{ij}, β_{ij} and γ_{ij} for the constraints $w_{ij} \geq 0$, $h_{ij} \geq 0$ and $r_{ij} \geq 0$ respectively, we obtain the following Lagrangian function:

$$L_{MNMF} = \alpha \, Tr(CC^T) - 2\alpha \, Tr(CWH^T) + \alpha(WHH^T W^T) + (1-\alpha) \, Tr$$
$$(L L^T) - 2(1-\alpha) \, Tr(L WR^T) + (1-\alpha)(WR R^T W^T) + Tr(\alpha W^T) + Tr$$
$$(\beta H^T) + Tr(R^T)$$

The partial derivatives of L_{MNMF} with respect to W, H and R are:

$$\frac{\partial L_{MNMF}}{\partial W} = -2\,\alpha\,CH^{T} + 2\,\alpha\,WHH^{T} - 2\,(1-\alpha)\,LR^{T} + 2(1-\alpha)WRR^{T} + \alpha$$

$$\frac{\partial L_{MNMF}}{\partial H} = -2\,\alpha\,C\,W + 2\,\alpha\,HWW^{T} + \beta$$

$$\frac{\partial L_{MNMF}}{\partial R} = -2\,(1-\alpha)\,L\,W + 2\,(1-\alpha)\,R\,WW^{T} + \gamma$$

Using the KKT conditions $\alpha_{ij} w_{ij} = 0$, $\beta_{ij} h_{ij} = 0$ and $\gamma_{ij} R_{ij} = 0$, we obtain:

$$-\alpha(CH^{T})_{ij}\,w_{ij} + (1-\alpha)\,(WHH^{T})_{ij} w_{ij} + -\alpha(L\,R^{T})_{ij}\,w_{ij} + (1-\alpha)\,(WRR^{T})_{ij} w_{ij} = 0$$

$$-\alpha(CW)_{ij}\,H_{ij} + \alpha(HWW^{T})_{ij} H_{ij} = 0$$

$$-(1-\alpha)\,(LW)_{ij}\,R_{ij} + (1-\alpha)\,(RWW^{T})_{ij} R_{ij} = 0$$

These equations lead to the following update rules:

$$w_{ij} = w_{ij}\,\frac{\alpha(CH^{T}) + \alpha(L\,R^{T})}{(1-\alpha)(WHH^{T}) + (1-\alpha)\,(WRR^{T})}$$

$$H_{ij} = H_{ij}\,\frac{(CW)}{(HWW^{T})}$$

$$R_{ij} = R_{ij} \frac{(LW)}{(RWW^T)}$$

3.4.3 Algorithm initialisation:

3.4.3.1 NMF sensitivity to initialisation

NMF is an iterative algorithm whose final result depends on the "quality" of the initialisation. We give below an example to show this sensitivity to the initial state. The example highlights the fact that given a matrix V and two different initialisations of W and H, the algorithm gives very different results: the first initialisation leads to (according to matrix W) the following membership vector [1 2 1 1 2 2] (which is correct), while with the second initialisation leads to the vector [2 2 1 2 2 2] (which is not correct).

	Initial W and H	Final W and H
V =	W =	W =
100 190 75 165 110 190 95 185 60 155 65 160	0.6268 0.6702 0.5503 0.5328 0.9379 0.2240 0.7457 0.4358 0.0490 1.1853 0.3315 0.8017	0.7142 0.5431 0.4511 0.6828 0.9716 0.1955 0.7190 0.4856 0.0646 1.1359 0.2532 0.9097
	H =	H =
	88.4827 181.8255 59.5292 116.8685	103.3677 170.3584 44.8176 127.5644

	The initial W and H	Final W and H
V= 100 190 75 165 110 190 95 185 60 155 65 160	W = 0.0944 0.9642 0.2698 0.7280 0.8406 0.5768 0.5344 0.7075 0.5655 0.4704 0.1879 0.7326	W = 0.4748 0.7691 0.2508 0.7360 0.8247 0.5872 0.5278 0.7057 0.1156 0.7397 0.1077 0.7758
	H = 102.3391 84.8546 49.1167 201.1047	H = 83.3414 91.7282 72.8385 193.2467

3.4.3.2 MNMF initialisation strategies

Likewise NMF, initialisation is an important step in the MNMF algorithm. We propose here to use three different techniques for this step.

a. Random initialization:

It's the simplest technique in which matrices W, H, and R are filled with random positive values [JOHN FREDERIC 08]:

W = random (n, k);
H = random (k, m);
R = random (k, s).

b. Acol initialization:

The second technique, random Acol initialization, was introduced by [Langville et al]. In this strategy, each column of S is initialized by averaging p randomly chosen columns of A. This will help to maintain any sparsity in A which would be lost with a random initialization with dense or long vectors. This method is also very computationally inexpensive and easy to implement [JOHN FREDERIC 08].

Acol Initialization Algorithm

Given: $A \in \mathbb{R}^{n \times m}$ with $A \geq 0$

1. Find p columns of A.
2. for each column of S do

    2a. Average p randomly chosen columns out of the matrix A.

3. End loop.

c. Toric k-means based initialisation:

It's an algorithm with based on the same principle of spherical K-means. The main difference is that toric K-means takes as input two data matrices whereas spherical K-means takes only one matrix. The toric K-means algorithm is described below:

Toric k-means steps

Centres Initialization

Input:

dataL n× n matrix

dataWD n× m matrix

k centres.

Initial centres

chose randomly k $centres_1$ from dataL

chose randomly k $centres_2$ from dataWD

while (nbr iteration < 50)

Calculate distance

S_1 = similarity between $center_1$ and dataL matrix

S_2 = similarity between $center_2$ and dataWD matrix

Membership

Associate each document D_i to a specific cluster j

Calculate new centres

$Ncenter_1$ is the arithmetic mean between dataL cluster k documents.

$Ncenter_2$ is the arithmetic mean between dataWD cluster k documents.

End while

3.4.4 Algorithm convergence criteria:

MNMF is an iterative algorithm which stops when a convergence condition is verified. Different convergence criteria can be used; we study three of them:

3.4.4.1Fixed number of iterations:

The simplest convergence condition is to stop the algorithm when a fixed number of iterations (for example 100) have been reached.

3.4.4.2 Membership vector stability:

This criterion is based on the stability of the membership vector (vector containing the cluster index of each object). If this vector does not change or changes slightly between two successive iterations, the algorithm stops.

3.4.4.3 Objective function stability:

This strategy is based on the stability of the objective function (Eq. 2). If the difference between two successive values of the objective function is below a threshold (for example 10^{-3}), the algorithm stops.

3.4.5 Data normalization:

Before applying the MNMF algorithm, the word-document matrix (C) is normalized using the TF-IDF weighting as follows:

- Calculate TF (Term Frequency):

$$TF_{ij} = \frac{W_{ij}}{d_j}$$

Where W_{ij} denotes the number of occurrence of a term t_i in a document d_j.

- Calculate IDF (Inverse Document Frequency):

$$\mathrm{IDF}_i = \log\left(\frac{d}{d_{f_i}}\right)$$

Where d denotes the number of documents in the corpus, and d_{f_i} the number of documents containing term t_i.

The two weights are then multiplied to obtain the TF-IDF value of each word occurrence.

3.5 Conclusion:

In this chapter, we have studied the NMF algorithm and three different approaches to such decomposition: multiplicative update rules, gradient descent algorithm and alternating least squares. After that we proposed a new algorithm (Multi-view Non-negative Matrix Factorization) for documents clustering. MNMF takes as input two different views of documents (in our case links and contents) to achieve better clustering results. We then described each step of the MNMF algorithm starting by the initialization which influences the quality of the final results. We described three different strategies to initialize the algorithm: random, Acol random and spherical k-means. We also discussed different convergence criteria that can be used to decide when to stop the algorithm: fixed number of iterations, stability of membership vector, and stability of the objective function.
In the next chapter we give some experimental results by applying MNMF on real datasets.

4

Experiments

4.1Experimental environment:

In this section, we present the adopted methodology to evaluate the proposed algorithm including used datasets and clustering evaluation measure.

4.1.1 Evaluation measures:

For the evaluation of our algorithm, many clustering criteria can be used. In our experiments we have used the F-measure and the Normalized Mutual Information (NMI) which is the most used measures for clustering evaluation.

4.1.1.1 F-measure:

It is an external measure which corresponds to the harmonic mean between precision and recall. It is computed by the following formula [A. Strehl 02]:

$$F - measure = \frac{2 * precision * recall}{precision + recall}$$

4.1.1.2 NMI:

The normalized mutual information between two clusters A and B is defined as [A. Strehl 02]:

$$NMI(A,B) = \frac{H(A) + H(B) - H(A,B)}{\sqrt{H(A).H(B)}}$$

Where H (A) and H (B) are respectively the entropy of A and B, H (A, B) is the joint entropy of A and B.

4.1.2 Datasets:

To evaluate our approach and compare it with other approaches, we used two datasets of scientific papers. The first one is a subset of the Cora collection, which is a set of more than 30,000 papers in the computer science field [A. McCallum et al 00]. Our subset contains 2700 documents where each one belongs to one of the following categories: Neural networks, genetic algorithms, learning theory, rule learning, probabilistic learning methods, and case based reasoning. The second

dataset contains a collection of 3000 papers extracted from the Citeseer database. The documents are classified into one of the following topics: Agents, databases, information retrieval, machine learning, and human computer interaction. Statistics on the two datasets are presented in Figure 4.1.

Dataset	Document	Categories	Average words per document	Average links per document	Document having in links	Document having out links
Cora	2708	7	62	2	1565	2222
Citeseer	2994	5	32	1.43	1760	2099

Table 4.1- statistics about Cora and Citeseer

4.2 Experimental results:

In order to evaluate the MNMF algorithm, we carried out three different experimentations. In the first one, we study the effect of normalisation on clustering quality; the second one investigates the importance of initialization in our algorithm; and the third one analyses the convergence speed of MNMF.

4.2.1 Normalization effect:

In this part, we compare the three algorithms: MNMF without normalisation, MNMF with TF-IDF normalization of the word-document matric, and toric K-means.

The three figures below show the F-measure values obtained when combining in links and content, out links and content, and in & out links and content.

4.2.1.1 F-measure values using Cora:

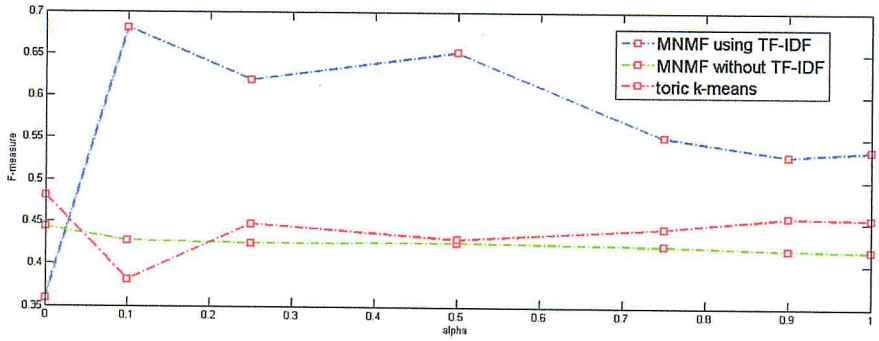


Figure 4.1 - F-measure values using in links and content with Cora

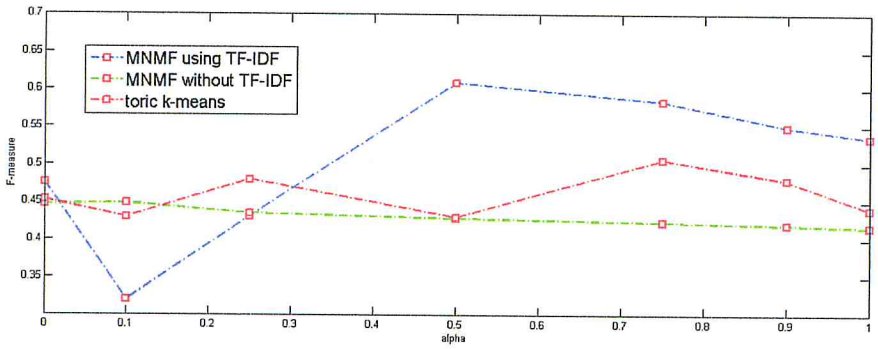


Figure 4.2 - F-measure values using out links and content with Cora

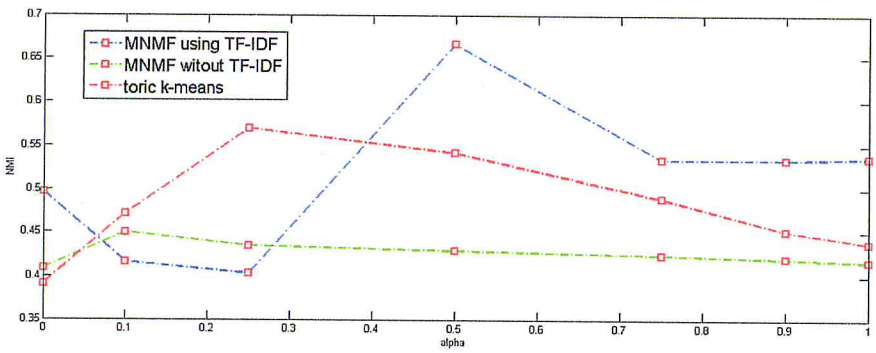


Figure 4.3 - F-measure values using in & out links and content with Cora

F-measure values obtained using in links shows that MNMF with normalisation gives better results than MNMF with no normalization. Figures also show that MNMF is better than toric K-means for almost all values of α and particularly when $\alpha \geq 0.5$.

4.2.1.2 NMI values using Cora:

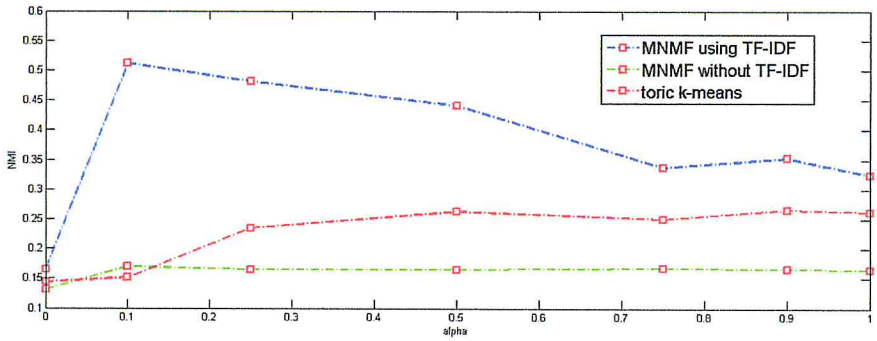


Figure 4.4 - NMI values using in links and content with Cora

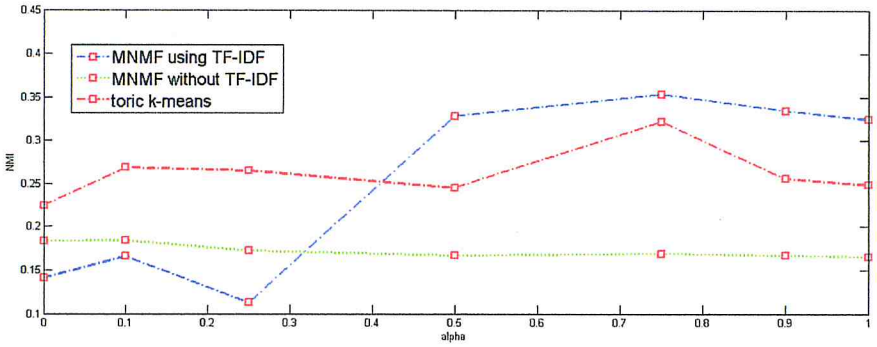


Figure 4.5 - NMI values using out links and content with Cora

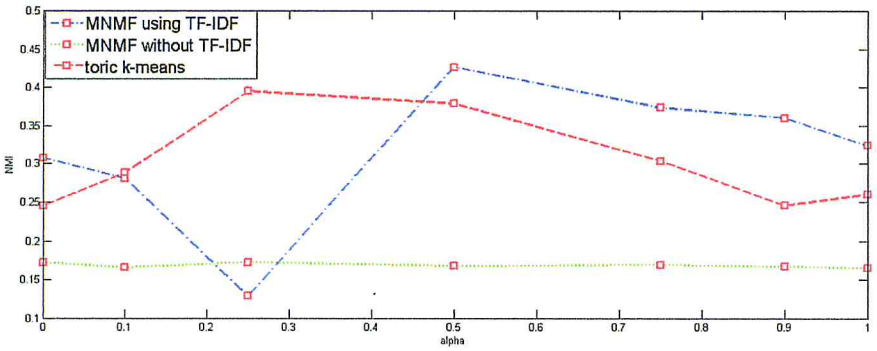


Figure 4.6 - NMI values using in & out links and content with Cora

NMI evaluation shows that MNMF with normalisation is again better than MNMF without it. Figures also show that the best results are obtained when combining in links and content information with MNMF; in this case MNMF is by far better than toric K-means. When combining out links or in & out links with content, MNMF gets the best results for $\alpha \geq 0.5$.

4.2.1.3 F-measure values using Citeseer:

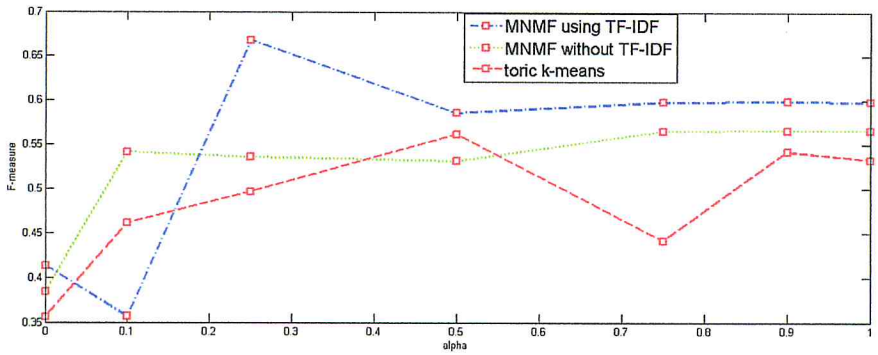


Figure 4.7 - F-measure values using in links and content with Citeseer

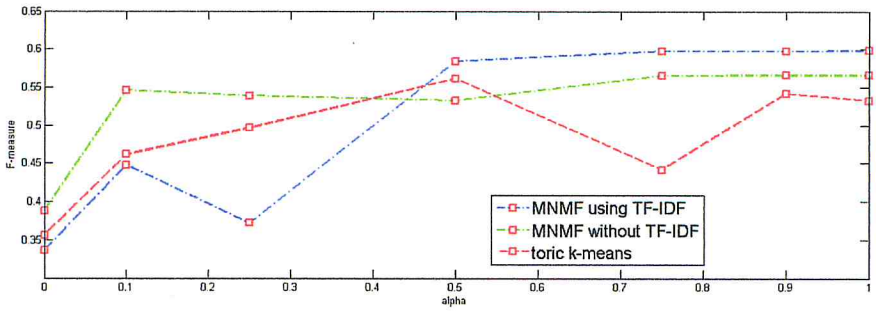


Figure 4.8 - F-measure values using out links and content with Citeseer

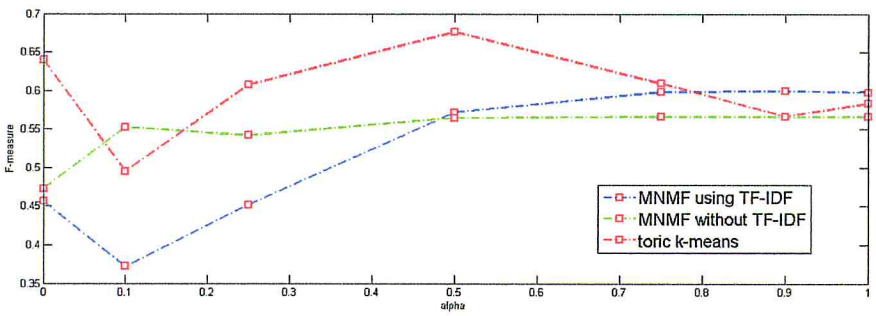


Figure 4.9 - F-measure values using in & out links and content with Citeseer

From the figures above, we observe that normalization is often beneficial to MNMF and particularly when $\alpha \geq 0.5$. In this latter case, MNMF turns out to be better than toric K-means except when combining both in an out links with content information where the superiority is achieved for $\alpha \geq 0.8$.

4.2.1.4 NMI values using Citeseer:

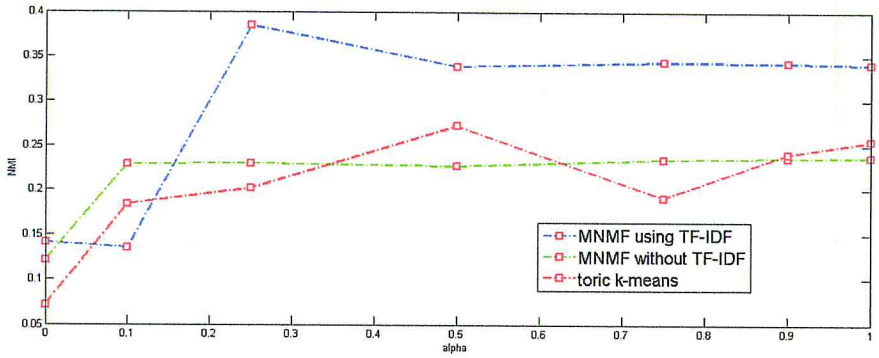


Figure 4.10- NMI values using in links and content with Citeseer

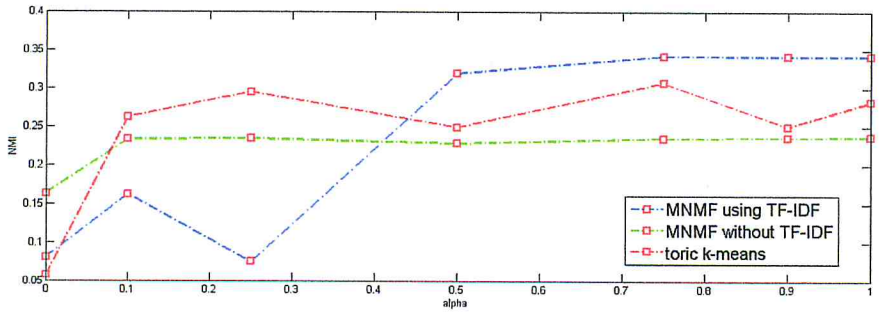


Figure 4.11 - NMI values using out links and content with Citeseer

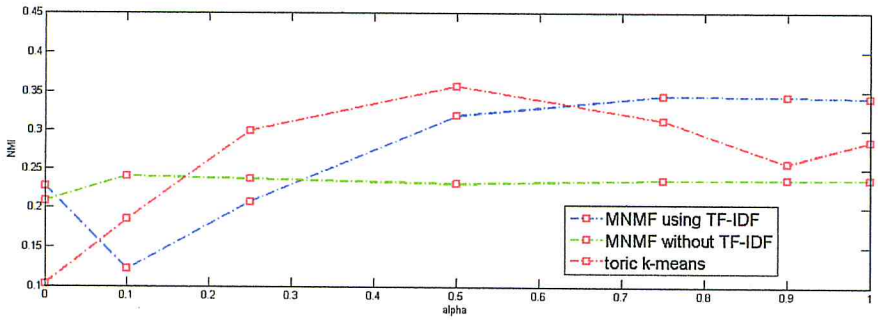


Figure 4.12 - NMI values using in & out links and content with Citeseer

According to NMI values, MNMF with normalization is always better than the other two algorithms especially for $\alpha \geq 0.6$. The best results are again obtained when combining in links and content information using MNMF with normalization.

4.2.2 Initialisation:

In this part, we study the effect of initialisation on MNMF. MNMF is iterative algorithm whose final results "quality" depends much on the "quality" of the initialisation. We therefore compare two different MNMF initialisation strategies: random initialisation and toric K-means based initialisation.

We report below the experimental results with Cora and Citeseer. Since we found that the NMI and the F-measure results were similar, we report only the NMI results.

Note that reported NMI values are those obtained after 150 iterations of the MNMF algorithm.

4.2.2.1 NMI values using Cora:

The three figures below show the NMI values with respect to combination factor (alpha) using different information sources (in links, out links, in-out links combined with content) of the Cora collection and using two intialisation techniques for MNMF.

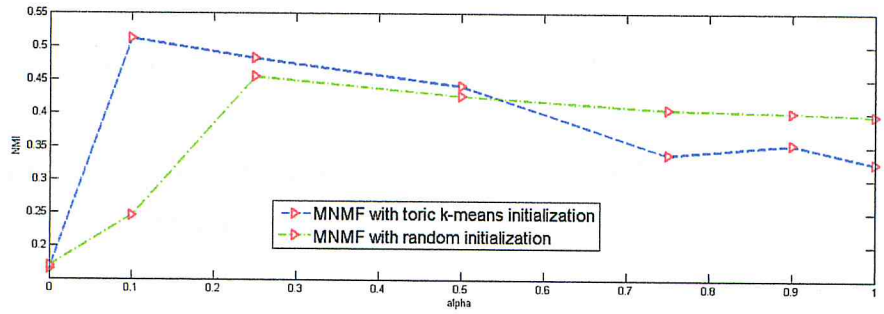


Figure 4.13 - NMI w.r.t. to α using in links and two initialisation strategies

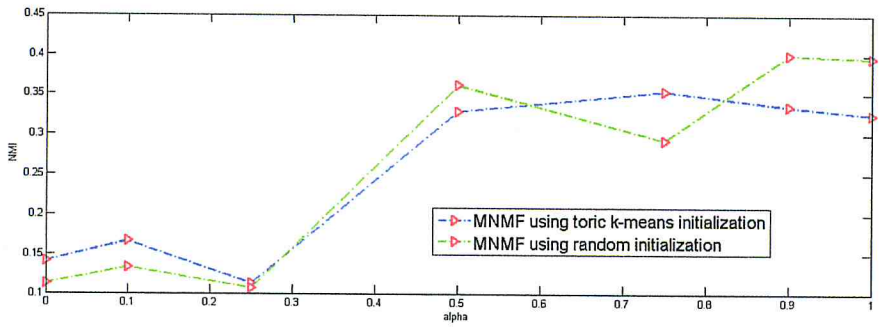


Figure 4.14 - NMI w.r.t. to α using out links and two initialisation strategies

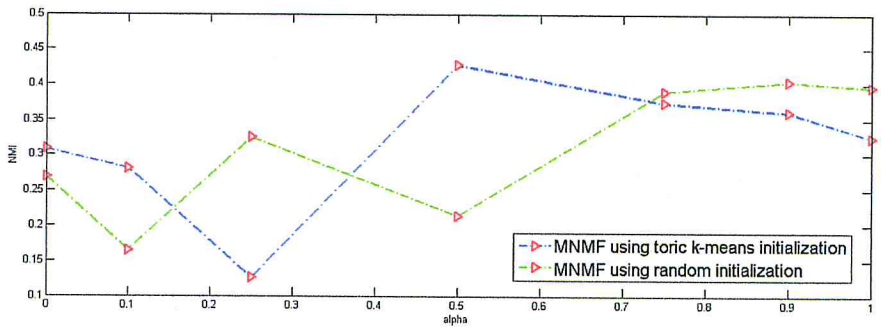


Figure 4.15 - NMI w.r.t. to α using in & out links and two initialisation strategies

4.2.2.2 NMI values using Citeseer:

The three figures below show the NMI values with respect to combination factor (α) using different information sources (in links, out links, in-out links combined with content) of the Citeseer collection and using two intialisation techniques for MNMF.

4.2.3 Algorithm convergence:

In this part, we study how initialization affects convergence speed by evaluating the MNMF objective function with respect to the number of iterations. Let's recall that the MNMF objective function is given by:

$$F_{MNMF} = \alpha\ (||C - WH||_F^2) + (1 - \alpha)\ (||L - WR||_F^2)$$

We run MNMF for 50 iterations using two different initialisation techniques: random initialisation and toric k-means based initialisation.

Figures below show the obtained results when combining both in and out link information with content using different values for the combination parameter α.

4.2.3.1 Convergence analysis using Cora:

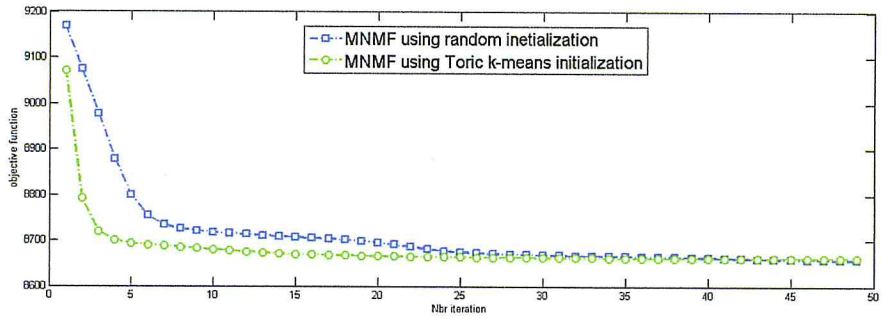


Figure 4.19 - MNMF objective function w.r.t the number of iterations using Cora ($\alpha = 0.25$)

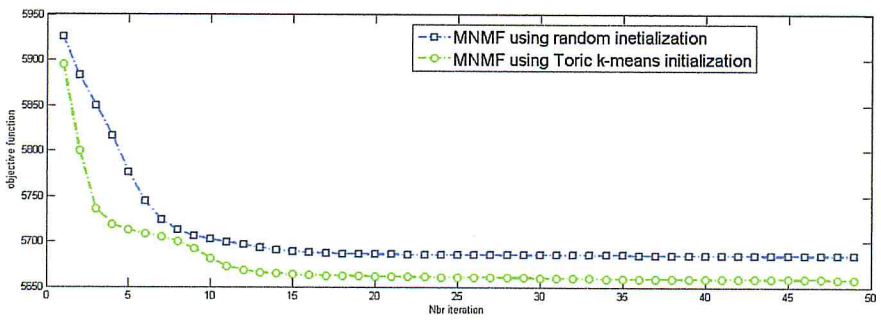


Figure 4.20 - MNMF objective function w.r.t the number of iterations using Cora ($\alpha = 0.5$)

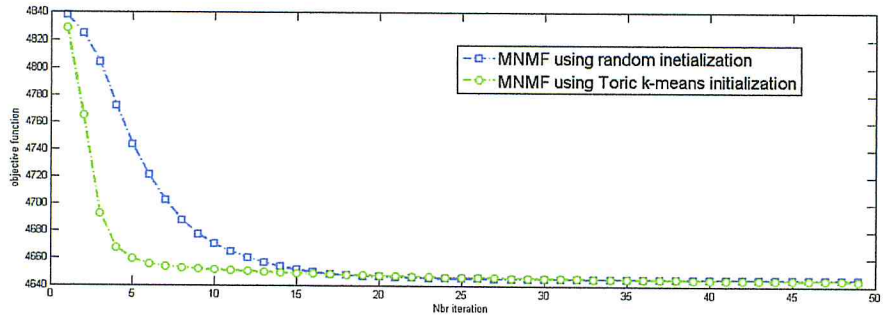


Figure 4.21 - MNMF objective function w.r.t the number of iterations using Cora ($\alpha = 0.75$)

4.2.3.2 Convergence analysis using Citeseer:

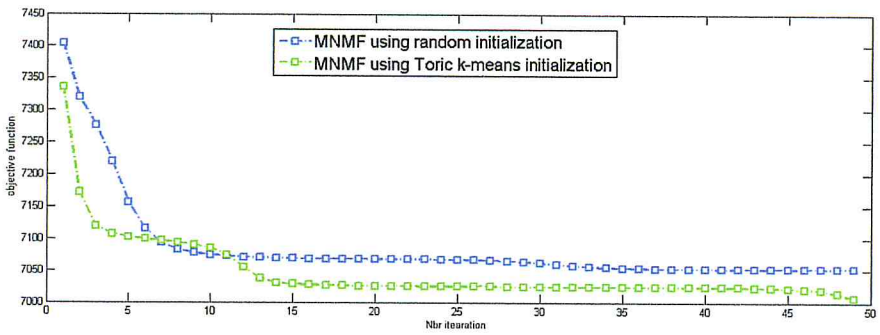


Figure 4.22 - MNMF objective function w.r.t the number of iterations using Citeseer ($\alpha = 0.25$)

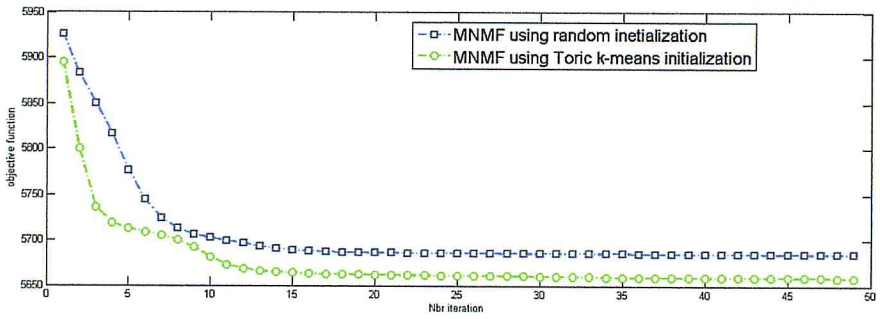


Figure 4.23 - MNMF objective function w.r.t the number of iterations using Citeseer ($\alpha = 0.5$)

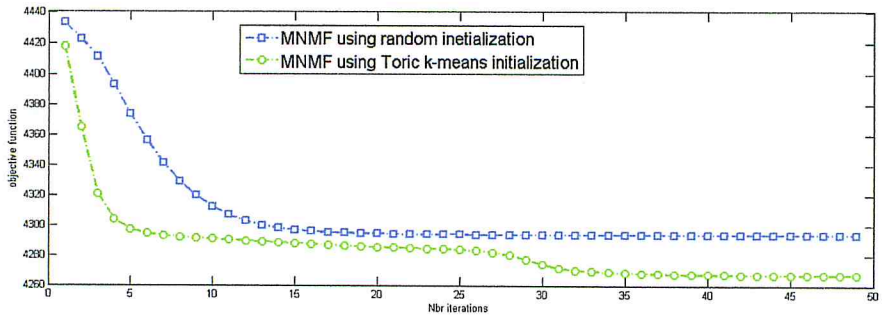


Figure 4.24 - MNMF objective function w.r.t the number of iterations using Citeseer ($\alpha = 0.75$)

The above figures show clearly that initialisation has a strong impact on MNMF convergence. Using the toric K-means based initialisation, MNMF converges (i.e. achieves a low value of the objective function) faster (i.e. in a smaller number of iterations) than with a random initialisation; this has an advantage in the sense that time complexity of the algorithm is reduced.

From the Citeseer results, we also observe that the toric k-means based initialisation not only speeds up MNMF convergence but also leads to better (i.e. smaller) values of the objective function.

Conclusion

In this thesis, we talked about the problem of document clustering. Recently, a variety of techniques has been proposed to solve this problem such as PLSA, K-means, etc. However, most of these techniques use only text information in the clustering process.

In this work we proposed MNMF (Multi-view Non-negative Matrix Factorization), a new hybrid algorithm which combines link and content information for document clustering. The proposed algorithm extends the well-known Non-negative matrix factorization (NMF) technique which has proven to be very effective for document clustering.

Using two document collections, we conducted various experiments to evaluate different aspects of our algorithm. We studied the effect of word-document matrix normalisation and the impact of initialisation on clustering performance. Experiments results show clearly that MNMF is better than an approach based solely on text information.

Bibliography

[Pieter Adriaans and Dolf Zantinge 96],"*Data Mining* (New York: Addison Wesley"), 1996.

[Oded Maimon and Lior Rokach 00] "Introduction to knowledge discovery in databases", 2000.

[Athman Bouguettaya 96] "On Line Clustering", *IEEE Transaction on Knowledge and Data Engineering* Volume 8, No. 2", April 1996.

[Bing Liu 98] "Web and Data Mining", 1998.

[Laurent candilier 06] "Contextualisation, vésualisation et évaluation en apprentissage non supervisé", 2006.

[Loïs RIGOUSTE 06] "Méthodes probabilistes pour l'analyse exploratoire de données textuelles", 2006

[Frizo JANSSENS 07] "Clustering of scientific field by integrating text mining and bibliometrics", Mai 2007.

[Alexander Strehl 02] "Relationship-based Clustering and Cluster Ensembles for High dimensional Data Mining", Mai 2002.

[Guillaume Cleuziou 04] "Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information", décembre 2004.

[Isabel Drost et al. 05] "Discovering Communities in Linked Data by Multi-View Clustering", 2005.

[Ronen Feldman, James Sanger 07], "Advanced Approaches in Analysing Unstructured Data", 2007.

[Anouar Mellakh 09] "Reconnaissance des visages en conditions dégradées", April 7, 2009.

[Michael W. Berry et al 06] "Algorithms and applications for approximate non-negative matrix Factorisation" 29 November 2006.

[Xuansheng Wang et al 12] "An effective initialization for orthogonal non-negative matrix factorization" Journal of Computational Mathematics, 2012.

[Inès MEGANEM 12] "Méthodes de Séparation Aveugle de Sources pour l'imagerie hyper spectrale. Application à la télédétection urbaine et à l'astrophysique" december 05, 2012.

[John Frederic 08] "Examination of Initialization Techniques for Non-negative Matrix Factorization" *Mathematics Theses.* Paper 63, 2008.

[Brian Wallace 04] Kuhn-Tucker conditions September 23, 2004.

[Daniel D. Lee, H. Sebastian Seung 01] "Algorithms for Non-negative Matrix Factorization", 2001.

[Chris Ding, Tao Li, Wei Peng 08] On the equivalence between Non-negative Matrix Factorization and Probabilistic Latent Semantic Indexing January 18, 2008.

[Da Kuang, Chris Ding, Haesun Park 12] "Symmetric Nonnegative Matrix Factorization for Graph Clustering" SIAM International Conference on Data Mining, Anaheim, CA, April 26, 2012.

[Nacim Fateh Chikhi et al. 08] "Combining Link and Content Information for Scientific Topics Discovery". In IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Dayton, Ohio (USA), IEEE Computer Society, pages 211-214, 2008.

[Nacim Fateh Chikhi 10] "Calcul de centralité et identification de structures de communautés dans les graphes de documents".

[Young-Min Kim 10] "Document Clustering in a Learned Concept Space" December 16, 2010.

[Nicholas O. Andrews and Edward A. Fox 07] "Recent Developments in Document Clustering" October 2007.

[A.K. JAIN et al 99] "Data clustering" 1999.

[Bing Liu 07] "Web data mining" 2007.

[Guillaume Cleuziou 04] "Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information", December 2004.

[Magnus Roselle 09] "Text clustering exploration" 2009.

[Lin and Pantel 01] "Discovery of inference rules for question answering" 2001.

[Turney et al 03] "From frequency to meaning: vector space models of semantics" 2003.

[Turney 08] "The latent relation mapping engine: algorithm and experiments" 2008
[Grobelnik, M. and Mladenic, D 04] "Text mining tutorial" 2004.

[Pauca et al. 2004] "Décomposition de spectrogrammes musicaux par des modèles de synthèse spectrale". 2004

[Ding et al. 2005] "Une nouvelle mesure pour l'évaluation des méthodes d'extraction des thématiques" 2005

[Michael W. Berry 06] "Algorithms and applications for approximate nonnegative matrix Factorization" 2006.

[A. McCallum et al] "Automating the construction of internet portals with machine learning". Information Retrieval Journal, 3 (200), 127–163, 2000.