UNIVERSITY KASDI MERBAH OUARGLA
Faculty of New Information Technologies and Communication
Department of Computer Science

ACADEMIC MASTER Thesis

Domain:	Computer Science and Information Technology
Track:	Computer Science
Speciality:	Industrial

Directed by: LIMANE Abdelhadi

Topic

Non Negative Matrix Factorization for Clustering problems and its relationship to Autoencoder

Defended on: 15/10/2020

Before the jury:

Dr. Farah Dabagh	Ouargla University	President
Dr. Meriem Korichi	Ouargla University	Examiner
Prof. KHERFI Mohammed Lamine	Ouargla University	Supervisor
Dr. HEDJAM Rachid	SQU, Sultanate of Oman	Co-supervisor

Academic year: 2019/2020

UNIVERSITY KASDI MERBAH OUARGLA
Faculty of New Information Technologies and Communication
Department of Computer Science

ACADEMIC MASTER Thesis

Domain:	Computer Science and Information Technology
Track:	Computer Science
Speciality:	Industrial

Directed by: LIMANE Abdelhadi

Topic

Non Negative Matrix Factorization for Clustering problems and its relationship to Autoencoder

Defended on: 15/10/2020

Before the jury:

Dr. Farah Dabagh	Ouargla University	President
Dr. Meriem Korichi	Ouargla University	Examiner
Prof. KHERFI Mohammed Lamine	Ouargla University	Supervisor
Dr. HEDJAM Rachid	SQU, Sultanate of Oman	Co-supervisor

Academic year: 2019/2020

Gratitudes

The Prophet , may Allah bless him and grant him peace, said, *"Allah does not thank the person who does not thank people"*. Hadith

Based on this Hadith:

I thank God who gave me the power to complete this thesis.

I extend my sincere gratitude and thanks to the honorable professors: Prof. KHERFI Mohammed Lamine and Dr. HEDJAM Rachid for their guided, notes, comments and criticism. As well as their patience throughout their supervision of this thesis, despite their many preoccupations.

I also thank all professors and colleagues who provided me with assistance, whatever its nature, and to all who gave me encouragement.

I also extend my sincere thanks to all of my distinguished professors at the Department of Computer science at the University of Kassadi Murbah Ouargla, especially:Dr. H.Djediai, Dr. B.Khadra, Dr. M.Korichi and Dr. A.Nadjet.

Dedications

I dedicate this work:

To my parents: my mother and father.

To whole family, my Grandma, my uncles and aunts, my brothers and sisters.

To the university family: my professors and my colleagues during the five years at the university.

To those we love them and they love us for the sake of Allah.

A special dedication for the person who helped me a lot in my university career:
Dr. MEZATI Messaoud.

Contents

List of Tables

List of Figures

List of Algorithms

Abstract

Nonnegative matrix factorization (NMF) is a technique which tries to factorize a nonnegative matrix into a product of two nonnegative ones. It has been successfully applied to different ends including dimensionality reduction, features extraction.

In this thesis, we explore NMF from two perspectives.

Firstly, we study the relationship between NMF and clustering which is a basic operation in Machine learning. Indeed, clustering has proven to be useful for image segmentation, documents analysis, marketing and it has been shown in many works that NMF can be used to perform clustering. However, the presence of noise in the data or the overlap between the clusters cause problems for most clustering algorithms including NMF which reduces its performance. In this thesis, we propose a solution to this problem by randomly applying NMF on different subsets, we aim to achieve the partial NMF which leads to the best clustering process, this does not mean that we abandon the rest of the partial NMFs, but rather that we weight them by the Boltzman's probabilities obtained by simulated annealing. Finally, we combine them to obtain the overall NMF which is responsible for clustering the entire data. The efficiency of our method has been proven for many data sets.

Secondly, we explore the relationship between NMF and deep learning, precisely Autoencoder (AE). This latter is a very powerful type on deep neural network which has been successfully applied to clustering, which is our concern in this thesis, as well as data compression, feature extraction, data embedding, etc. By imposing non-negative constraints on an AE, it can learn features that show a captured part-based representation of the data. Furthermore, the nonnegative constraint makes deep AE easy to interpret by humans.

Key-words: Non negative matrix factorization, Clustering , Metaheuristic, Simulated annealing, Autoencoder.

Résumé

La factorisation matricielle non négative (NMF) est une technique qui factorise une matrice non négative en un produit de deux matrices non négatives. Il a été appliqué avec succès à différentes fins, y compris la réduction de dimensionnalité, l'extraction de caractéristiques.

Dans cette thèse, nous explorons NMF en deux perspectives.

Tout d'abord, nous étudions la relation entre NMF et le partitionnement de données qui est une opération de base en l'apprentissage automatique. En effet, le partitionnement de données s'est avéré utile pour la segmentation d'image, l'analyse de documents, le marketing et il a été montré dans de nombreux travaux que NMF peut être utilisé pour effectuer un partitionnement de données. Cependant, la présence de bruit dans les données ou le chevauchement entre les clusters pose des problèmes pour la plupart des algorithmes de clustering dont NMF qui réduit ses performances. Dans cette thèse, nous proposons une solution à ce problème en appliquant au hasard NMF sur différents sous-ensembles, nous visons à atteindre le NMF partiel qui conduit au meilleur processus de clustering, cela ne signifie pas que nous abandonnons le reste des NMF partiels, mais plutôt que nous les pondérons par les probabilités de Boltzman obtenue par recuit simulé. Enfin, nous les combinons pour obtenir le NMF global qui est chargé de regrouper l'ensemble des données. L'efficacité de notre méthode a été prouvée pour de nombreux ensembles de données.

Deuxièmement, nous explorons la relation entre NMF et l 'apprentissage profond, précisément Auto-encodeur (AE). Ce dernier est un type très puissant de réseau neuronal profond qui a été appliqué avec succès au partitionnement de données, ce qui est notre préoccupation dans cette thèse, ainsi que la compression de données, l'extraction de fonctionnalités, le plongement de données, etc. En imposant des contraintes non négatives à un AE, il peut apprendre des fonctionnalités qui affichent une représentation partielle capturée des données. De plus, la contrainte non négative rend l'AE profonde facile à interpréter par les humains.

Mots clés : Factorisation matricielle non-négative , Partitionnement de données , Métaheuristique, Recuit simulé, Auto-encodeur.

ملخص

التحليل اللاسلبي للمصفوفة (NMF) هي تقنية لتحليل مصفوفة غير سالبة إلى جداء مصفوفتين غير سالبيتين ، تصلح هذه التقنية للعديد من الأغراض من بينها تقليل الأبعاد واستخلاص المميزات.

في هذه الأطروحة نحاول أن نتطرق إلى التحليل اللاسلبي للمصفوفات من منظورين مختلفين.

الجزء الأول من الأطروحة يدرس العلاقة بين التحليل اللاسلبي للمصفوفات و تجميع البيانات والذي هو أحد العمليات الأساسية في التعلم الآلي. والذي بدوره يمكن أن يستخدم من أجل تقطيع الصور، تحليل المستندات و بعض المعاملات السوقية، كما أثبتت العديد من الأعمال العلمية أن التحليل اللاسلبي للمصفوفات قادر على تجميع البيانات. علاوة على ذلك، تعاني معظم خوارزميات تجميع البيانات بما في ذلك التحليل اللاسلبي للبيانات من بعض المشاكل التي تؤدي إلى تقليل أدائها مثل مشكلة حدوث ضوضاء في البيانات و كذلك التداخل بين المجموعات. في هذه الأطروحة اقترحنا حلا لهذه المشكلة من خلال تطبيق التحليل اللاسلبي للمصفوفات على مجموعات جزئية بشكل عشوائي، بحيث نهدف الحصول على أفضل مجموعة جزئية يمكن يمكن أن يؤدى عليها التجميع، في نفس الوقت لم نتخلص من باقي التحليلات الأخرى، بل قمنا بوزنها عن طريق احتمالات بولتزمان المتحصل عليها من خوارزمية التلدين المحاكي، في الأخير نقوم بدمج كل الحلول الجزئية بعد إعطائها الوزن المناسب لنصل إلى التحليل اللاسلبي الكلي المسؤول عن تجميع البيانات الكلية.

أما الجزء الثاني فهو يدرس العلاقة بين التحليل اللاسلبي للمصفوفات و التعلم العميق، وبالضبط المشفر التلقائي (AE) . هذا الأخير يعتبر أحد أقوى أنواع الشبكات العصبية والذي لديه القدرة على تجميع البيانات والذي هو محور تركيزنا هاهنا، كما يمكن أن يستخدم لضغط البيانات، استخلاص المميزات وتضمين البيانات، ...إلخ. بفرض القيد اللاسلبي على المشفر التلقائي يصبح قادرا على تعلم المميزات التي تظهر تمثلا قائما على أجزاء من البيانات الأصلية. والأكثير من ذلك أن الإنسان يصبح قادرا على فهم البنيات العميقة للمشفر التلقائي.

الكلمات المفتاحية: التحليل اللاسلبي للمصفوفة، التجميع، الأدلة العليا، التلدين المحاكي، المشفر التلقائي.

Chapter 1

Introduction

1.1 General introduction

In our vast world, a very large amount of data is generated every millisecond, with many people sharing their daily photos and videos in different social media, with millions of students having to follow their lessons in live or recorded lectures, with many patients needing to track all of their movements and even their heart rate and record their data for treatment,...,etc. These data will not be useful without processing. Machine learning includes many algorithms that provide many tasks to manipulate data and make it useful, including regression, classification, and clustering.

In this thesis, we are interested in non-negative matrix factorization (NMF) which is a method to factorize a non-negative matrix into two factors, which in turn must be non-negative matrices. The non-negativity constraint makes NMF factors more interpretable and useful for training part-based representations than other methods, including Principal component analysis (PCA), and Vector Quantization (VQ) which is a quantization technique used usually for data compression .[43] NMF was introduced in 1994 by Paatero and Tapper[58], and they called it Positive Matrix Factor (PMF), but Lee and Seung are considered the godfathers of NMF, where the NMF has become so popular after their work in 1999[43] which shows the ability of NMF to learn the parts of objects and they presented a new algorithm to solve NMF problem called multiplicative update rules (MU) in 2001 [44]. NMF has been successfully applied in many fields including text mining [57], bioinformatics[26] and pattern recognition[49].

On the other hand, Clustering is a fundamental operation in Machine learning. It aims at splitting a dataset into a number of clusters such that data from the same cluster are more similar to each other than to the data from other clusters. It is a descriptive tool that allows to analyse the distribution of data in the feature space, and can be used as a pre-processing step for other operations such as classification. Cluster analysis has been applied for a variety of ends, including marketing, biology, libraries, segmentation and text mining. Low-rank factorization techniques have shown a good capability for clustering and NMF is of them [47](we had included a whole chapter to study this point).

1.2 Problematic

The presence of noise and overlap between the clusters are one of the most important problems that reduce the performance of many clustering algorithms including NMF, and as far as we know, all the existing works apply NMF on the data as a whole, that increases the likelihood of falling into the mentioned problems. Therefore, the proposed method applies NMF on partial subsets chosen from the whole data, the partial NMF(PNMF) is weighted and combined to obtain the overall NMF that used for the whole clustering process, the weighting step means assigning a good weight to the PNMFs which leads to a good clustering and assigning a bad weights to the PNMFs which leads to a bad clustering. We get these weights by using simulated annealing algorithm. Chapter VI gives more details on the proposed method.

In addition, we try to study the relationship between NMF and autoencoders, and discuss some of the work on this topic.

1.3 Objective of the thesis

The first objective of this thesis is purely exploratory, we will explore NMF, we'll then explore the possible extending of NMF to one of various Machine Learning problems such as clustering. Finally, we will investigate its relationship to autoencoder as a deep learning method.

The second objective is to propose a new method that increases performance of NMF for clustering problems.

1.4 Organization of the thesis

This thesis is organized as follows:

Chapter II: it gives an overview on NMF, where we described the general formulation of the NMF problem, then we turned to its cost functions and its applications, we concluded this chapter with the most commonly used algorithms for the NMF problem.

Chapter III: contains an overview of clustering problem and reviews the types of its algorithms and some useful applications.

Chapter IV: elucidates the relation between NMF and clustering.

Chapter V: presents a new method that we have proposed to improve the performance of NMF for the clustering problem. Also, it report the experimental results on many data sets .

Chapter VI: the first section of this chapter describes the autoencoder and reviews its types. In the second section, we illustrate the relation between NMF and autoencoders.

Chapter 2

Non Negative Matrix Factorization

Non-negative matrix factorization(NMF) is a key tool in data analysis, and is widely used for various purposes including data compression, features extraction and clustering which will be covered in this thesis.

This chapter is an introduction to NMF where we will see how we can formulate an NMF problem and describe the different costs functions of NMF. Next, we mention for what we use NMF. Finally, we illustrate two algorithms to solve this problem.

2.1 Formulation

In fact, NMF is not a new technique, it had appeared about 30 years ago, Paatero and Tapper[58] were the first to pay attention to NMF, but Lee and Seung are truly their spiritual fathers, where NMF became famous after it was touched in their paper [43], which demonstrated the NMF's ability to learn the parts of object.

Below we will discuss NMF formulation and its cost functions.

2.1.1 NMF problem description

Suppose $X \in \mathbb{R}_+^{mn}$ is a given non-negative matrix; it has only zeros or positives elements$(i.e.X_{ij} \geq 0)$, $W \in \mathbb{R}_+^{mr}$ and $H \in \mathbb{R}_+^{rn}$ are two non negative matrices we want to find where $r \ll min(m, n)$, the non-negative matrix factorization factorizes X into a product of W and H, it can be written as:

$$X \approx WH \tag{2.1}$$

WH is called a non-negative matrix factorization of X, this product is not necessarily equal to X (that is a semi-problem of NMF called Exact NMF). Generally, in the most NMF problems we want to find an approximation .

In other mathematical expression, NMF can be defined as:

$$X = WH + E \tag{2.2}$$

Where E is the approximation matrix error that we want to optimize and is equal to 0 in the case of exact NMF.

The next lines explain the manner of applying NMF on our data.

Our data (Images, documents, ..., etc) is represented as matrix X, each element in this data corresponds a column in X, this matrix is factorized into two matrices W and H(m x r and r x n), and r is less than min(m,n), such that: r is the desired lower dimension.

We can write also each column x of X like: $x_i = Wh_i$ where i is the index of the column. In another meaning, each column x is a combination of the columns of W, and the coefficients (weights) here are the components of the column h_i.[44, 39]

2.1.2 Cost functions

To calculate the approximation error or to know which factorization is the better between two factorizations we have to define a cost function. NMF has many cost functions, this is among the strong points of NMF that makes it capable of handling many problems. In this subsection we will see three cost functions of NMF.

Frobenius norm(Euclidean distance)

The squared Euclidean distance is the most popular NMF's cost function, it appeared in many papers including Lee and Seung's paper[44]. It can be written as:

$$f(\mathbf{W}, \mathbf{H}) = \frac{1}{2}\|\mathbf{X} - \mathbf{WH}\|_F^2, \tag{2.3}$$

which is equivalent to:

$$\min_{W,H>0} \|\mathbf{X} - \mathbf{WH}\|_F^2 = \min_{W,H>0} \sum_{i,j} (\mathbf{X}_{ij} - \mathbf{WH}|_{ij})^2, \tag{2.4}$$

Kullback-Leibler divergence

Kullback-Leibler divergence is used to measure the difference between two probability distributions p and q. The expression of the Kullback-Leibler divergence is given by:

$$D_{KL} = \sum_i p(i) \log(\frac{p(i)}{q(i)}), \qquad s.t \qquad i \in probabilty\, space. \tag{2.5}$$

In addition to Euclidean distance, we can formulate NMF's cost function with (generalized) Kullback-Leibler divergence as following:

$$f(\mathbf{W}, \mathbf{H}) = D_{KL}(\mathbf{X} \| \mathbf{WH}) \tag{2.6}$$

which is equivalent to:

$$\min_{W,H>0} \sum_{i,j} (\mathbf{X}_{ij} \log \frac{\mathbf{X}_{ij}}{\mathbf{WH}|_{ij}} - \mathbf{X}_{ij} + \mathbf{WH}|_{ij}) \tag{2.7}$$

This also appeared early in Lee and Seung's paper[44, 43] .

Itakura-Saito divergence

Itakura-Saito divergence is used to measure the difference between two spectra. defined by:

$$D_{IS}(x|y) = \frac{x}{y} - log\frac{x}{y} - 1 \tag{2.8}$$

This measure had successfully applied for NMF[24] as following:

$$f(\mathbf{W}, \mathbf{H}) = D_{IS}(\mathbf{X}, \mathbf{WH}) \tag{2.9}$$

which is equivalent to:

$$\min_{W,H>0} \sum_{i,j} (\frac{\mathbf{X}_{ij}}{\mathbf{WH}|_{ij}} - \log \frac{\mathbf{X}_{ij}}{\mathbf{WH}|_{ij}} - 1) \tag{2.10}$$

Not only are these cost functions, several more are not discussed here, some of these are discussed in this book [15] including Csiszar Divergences, Bregman Divergence, Alpha-Divergences, Beta-Divergences, Gamma-Divergences, ..., etc.

After we knew NMF formulations and cost functions, the question now is how to find the two matrices W and H, this is what we will discuss in the section 2.3.

2.1.3 NMF extensions

In the standard NMF, we have only the non-negativity constraint, the researches tried to impose other constraints on the NMF, this with other parameters led to appear several extensions of the NMF, we cannot explain it here, but we will mention it, among the NMF extensions we find: Regularized NMF, Projective NMF, Semi-NMF, Convex-NMF, Cluster NMF, Kernel-NMF, Multi-layer NMF, Binary NMF, Weighted Feature Subset NMF, Robust NMF, ..., etc.

2.2 Applications

NMF is widely used in many fields, including computer vision, image processing, pattern recognition[46], text mining[7, 59], bio informatics[11]. NMF is also widely used for audio in single-channel source separation[66] and bandwidth expansion[6]. NMF algorithms were also applied to analysis financial data[16]. Finally, NMF can be viewed as a clustering problem, that what we will see in Chapter 3.

In this section we will discover two applications with examples;first is NMF in image processing where we see how NMF is applied to extract facial features and second is NMF in text mining where we illustrate an example to apply NMF for document modeling.

2.2.1 Image processing

We can use NMF in the field of Image processing, and in this subsection we illustrate how we apply it to extract facial features.

Let $X \in \mathbb{N}^{mn}$ represents the data-set of images(gray-level) of different faces, each column represents one image of face, the face is represented by m pixels. When we apply NMF on the matrix X we obtain two matrices $W \in \mathbb{R}^{mr}$ and $H \in \mathbb{R}^{rn}$.

$$X(:,j) \approx \sum_{k=1}^{r} W(:,k)H(k,j) \tag{2.11}$$

r is the number of features.
$W(:,k)$ is the k^{th} features, the feature can be eye, nose, mustache..., etc .
$H(k,j)$ is the importance of k^{th} feature in the j^{th}face. The NMF model in eq.(2.11) can interpret each face as a combination of basis images which are features and the importance of each feature in the face. The non-negative constraint gave the NMF the power to extract meaningful features. Where we cannot build basis images (features) from negative values.

In addition, Lee and seung applied three techniques PCA, VQ and NMF on the same data set, and they obtained that the NMF is the best one to extract the parts of the face.[43]

And in another problem, the faces are captured under a bad conditions as changes in expressions(smile, anger, ...) and change in light(left light on, right left on, both left on), etc. The authors compared between the NMF and PCA, the NMF results were the best.[26]

2.2.2 Text mining

Besides the image processing, NMF plays also an important role in the text mining. NMF can be applied for text mining in topic modeling [25], document clustering [76, 64], topic detection and tracking and email analysis [8, 13].

Next, we illustrate how we can apply NMF for topic modeling.

Let $X \in \mathbb{R}^{mn}$ is the matrix of our data, each column x of X represents a document, x contains the frequency of the words in the document. That means that $X_{i,j}$ is how many times the word 'i' appeared in the document 'j'. In this model, the number of the words is not important.

NMF will produce two matrices $W \in \mathbb{R}^{mr}$ and $H \in \mathbb{R}^{rn}$.

$$X(:,j) \approx \sum_{k=1}^{r} W(:,k)H(k,j) \tag{2.12}$$

r is the number of topics.

$W(:,k)$ is the k^{th} topic.
$H(k,j)$ is the importance of k^{th} topic in the j^{th}document. In other meaning, the matrix H informs us each topic is importance in each document.

The model in the eq.(2.12) can represent each document $X(:,j)$ as a combination of topics W and its weights H(importance). Simply, we can know the topic of any document; the index of the max entry in the column H(:,j) represents The topic number of the j^{th} document.

Next, we take a numerical example for more illustration. Let X is the matrix of our data, it has 4 documents, each document has set of words with different frequencies.

X=

Doc_id	D1	D2	D3	D4
Influenza	5	1	1	6
Mosque	0	4	1	0
Pizza	1	1	5	2
Covid19	6	0	1	5
Tomato	1	1	6	1
Fasting	0	4	0	0

We factorize X into W and H as following:

W =

Top_id	T1	T2	T3
Influenza	7.8	0.1	1.0
Mosque	0	1	3.8
Pizza	1.8	4.9	0.2
Covid19	7.8	0	0
Tomato	1.1	6.0	0
Fasting	0	0	4

H =

Doc_id	D1	D2	D3	D4
T1	0.7	0.0	0.1	0.7
T2	0.0	0.2	0.9	0.0
T3	0.0	1.0	0.0	0.0

From the gray cells in the matrix H (the max value in the column) we can know the topic of each document and we read it as follows:
The document D1 belongs to the topic T1.
The document D2 belongs to the topic T3.
The document D3 belongs to the topic T2.
The document D4 belongs to the topic T1.

On the other side, from the matrix W we can conclude the title of each topic,That is, by taking the largest cells in each column, the conclusion can be as follows:
max(:,T1) is 'Influenza' and 'Covid19' $\longrightarrow$ Topic T1 is virus.
max(:,T2) is 'Pizza' and 'Tomato' $\longrightarrow$ Topic T2 is food.
max(:,T3) is 'Mosque' and 'Fasting' $\longrightarrow$ Topic T3 is Islam.

2.3 Algorithms

NMF is non-convex optimization problem in both W and H, this poses difficulty in arriving at the best approximation for the factorization. More that, the factorization is not unique, this can be easily seen through $X = WDD^{-1}H$ (where D is an invertible matrix and D^{-1} is its inverse).

After its presentation by Paatero and Tapper[58] in 1994, researchers have started to create algorithms to solve NMF problem, among the algorithms we find: alternating non-negative least square(ANLS)[38], gradient descent algorithms[30], projected gradient algorithms[48], newton-type algorithms[78] and many other useful algorithms have been utilized to solve NMF problem.

Generally, these algorithm aims to find W and H with the minimum error by minimize one of the cost functions which were described in Section 2.1.2.

In this section we describe two different algorithms for NMF: alternative least square algorithm and the multiplicative update rules of Lee and Seung. Before we explain the two algorithms we have firstly define the used notations:

X: data matrix.
W: basis matrix.
H: coefficient matrix.
T: transpose of a matrix.

2.3.1 Lee and Seung's Multiplicative update rules

Lee and Seung's multiplicative update rule algorithm[44] for NMF is the most popular. In fact, Lee and Seung proposed two kinds of these rules; the first is based on Euclidean distance and the second is based on Kullback-Leibler divergence, it is easy to implement where the matrices W and H are updated through multiplications as provided in algorithm 1 and algorithm 2.

The basic steps of Lee and Seung MU rules algorithm which is based on euclidean distance are summarized in Algorithm 1 and which is based on KL divergence are summarized in Algorithm 2.

Algorithm 1: Lee and Seung's MU rules (Euclidean distance) Algorithm

```
1 W=rand(m,k);
2 H=rand(k,n);
3 for $i \leftarrow$ 1: maxiter do
4     H=H$\odot$(W$^T$X)$\oslash$(W$^T$WH+$\epsilon$);
5     W=W$\odot$(XH$^T$)$\oslash$(WHH$^T$+$\epsilon$);
```

Algorithm 2: Lee and Seung's MU rules(KL divergence) Algorithm

```
1 W=rand(m,k);
2 H=rand(k,n);
3 for $i \leftarrow$ 1: maxiter do
4     $H = H \odot (W^T(X \oslash (WH + \epsilon))) \oslash W^T ee^T$;
5     $W = W \odot ((X \oslash (WH + \epsilon))H^T) \oslash ee^T H^T$;
```

At the beginning of this algorithm, W and H are initialized with non-negative values randomly, but there are many other techniques to initialize these matrices, from the proposed techniques:

- We repeat the initialization process many times and choose the best configuration, this technique is simple but time-consuming.
- We can run a clustering algorithm and use its result to initialize our matrices.

This paper[42] addresses and compares more initialization strategies. This algorithm needs a stopping criterion, among the stopping criteria that meet the purpose:

- We predefine number of iterations.
- We fix the running time of the algorithm
- We select a threshold for the cost function and when the cost function reaches the predetermined threshold we stop it.

ϵ is added to avoid the division by zero

After Lee and Seung introduced the multiplication update rules, they have proven that the algorithm converges to the local minimum.

We must also mention that this algorithm ensures the non-negativity constraint of both W and H during the updating.

Advantages

- It can be converge to fixed point; the convergence theory is presence.
- When we initialize well both of W and H, the convergence is faster and we get good results.

Disadvantages

- Convergence is not necessarily towards a global min; I mean the convergence maybe towards local min or saddle point.
- This algorithm is slowly, at each iteration, we have many multiplications.

2.3.2 Alternating least square algorithm

Alternating least square (ALS)algorithm was the first suggested algorithm to solve the NMF problem[58]. In fact, NMF objective function is not necessarily decreased after each iteration with this algorithm.[40]

Alternately, we fix W and find H by using the least square method then we do the opposite; we fix H and find W by using LS.

Algorithm 3: Alternating least square Algorithm

```
1 W=rand(m,k);
2 for $i \leftarrow 1: maxiter$ do
3     Solve for H in Matrix equation $\mathbf{W}^T\mathbf{W}\mathbf{H} = \mathbf{W}^T\mathbf{X}$;
4     Set all negative elements in H to 0;
5     Solve for W in Matrix equation $\mathbf{H}\mathbf{H}^T\mathbf{W}^T = \mathbf{H}\mathbf{X}^T$;
6     Set all negative elements in $W$ to 0;
```

The authors ’Paatero and Tapperr’ did not use the constraint of sparseness, but it appeared after that in other papers, and they called this method Positive Matrix Factorization. It should be noted that it is fast and works well in practice.

Advantages

- It is fast and work well in practice
- We only need to initialize one matrix.

Disadvantages

- No convergence theory .
- When once an element of W or H has the value 0, it must remain 0.

Chapter 3

Clustering

Machine learning has many types of tasks; supervised learning, unsupervised learning, semi-unsupervised learning,..etc. Each type of them is divided into several other sub-problems. For example, unsupervised learning contains clustering, Association and dimension reduction,..etc.

This chapter is considered as an introduction to the clustering. We will start by defining clustering , then we give the main differences between in and classification, after that, we review the famous types of clustering and some existing algorithm and we conclude this chapter by given some useful application including image segmentation, image and document retrieval, Marketing and documents analysis.

3.1 Definition

In Cambridge dictionary [1], the cluster is:

> A group of similar things that are close together, sometimes surrounding something.

In machine learning, clustering or unsupervised classification (in some references) falls under unsupervised learning, the process of clustering divides the data into a number of clusters(groups,subsets,..), such that the elements of each cluster are similar between them and dissimilar to the elements that belong to other clusters.

[1]https://dictionary.cambridge.org/dictionary/english/cluster

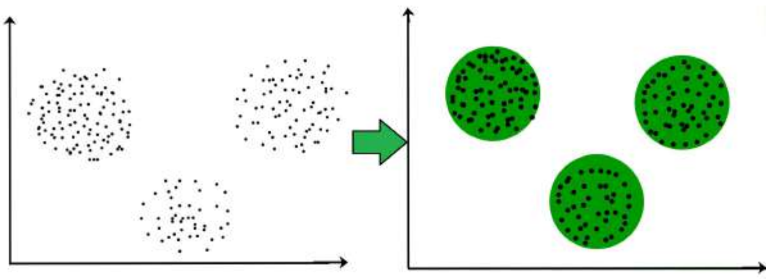


Figure 3.1: cluster data into three clusters[60]

Clustering versus classification

If you are a beginner in the field of machine learning you might think that classification and clustering are the same thing but this is not true.

The main difference depends on the nature of the data; the data is labeled in the classification and unlabeled in the clustering, for that we can consider the classification as a supervised learning and the clustering as un unsupervised learning.

The reason for resorting to the clustering is the lack of labeled data,because the labeling process is expensive and time consuming.[75]

Furthermore, the goal of the clustering according to Lior Rokach and Oded Maimon is descriptive and the gol of the classification is predictive.[51]

3.2 Algorithms

There are those who divide the clustering algorithms into partitional clustering and hierarchical clustering, And there are those who categorize it into density-based methods, model-based clustering and grid-based methods, And there are those who divide it into soft and hard clustering. This section will review some clustering algorithms.

3.2.1 Partitional Clustering

In partitional clustering, the algorithm organize data points $x_i, i = 1..N$ into K clusters $c_j, j = ...K$ by optimize a specific cost function and trying to improve the clustering quality iteratively. These types of algorithms require initially determining the number of clusters and require a set of initial.

Mathematically, Suppose $X = \{x_1, x_2, ..., x_j, ..., x_N\}$ is a given set of input, and $x_j = \{x_{j1}, x_{j2}, ..., x_{jd}\} \in \mathbb{R}^d$. The partitional clustering aims to divide X into a K-partitions (clusters), obtained clusters are $C = \{C_1, C_2, ..., C_k\}$, such that:

- $C_i \neq \emptyset, i = 1..k$; each cluster must to has an elements; the empty cluster has no significance.
- $\cup_{i=1}^{k} C_i = X$; the union of all the clusters is equivalent to the entire data X.[75]

Below, we explain K-Means and fuzzy C-Means algorithms that implement the concept of partitional clustering.

K-Means algorithm

K-Means[50] is the most popular partitional clustering algorithm. It is studied at universities and institutes for its simplicity of implementation and its ease to interpretation , and became as a traditional introduction to clustering. K-Means aims to cluster the data by minimize the sum of squared error cost function Eq.(3.1) iteratively.

$$SSE = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - c_k\|^2 \tag{3.1}$$

c_k is the centroid of the k^{th} cluster.

$$c_k = \frac{\sum\limits_{x_i \in c_k} x_i}{|C_k|} \tag{3.2}$$

Algorithm 4: K-Means algorithm

```
1 Initialize K centroids;
2 Assign each object in the data set to the nearest centroid ;
3 Recalculate the centroids based on the current partition;
4 Repeat steps 2 and 3 until there is no change for each cluster;
```

Indeed, the performance of K-Means is influenced by many factors, among them the determination of the number of clusters K, and the initialization of centroids.[61]

There are several methods for initialize the centroids, including Hartigan and Wong[27], Milligan[52], Bradley and Fayyad[10]and K-Means++[4]. In the other hand, there are other technique to estimate the optimal number of clusters K, among which Calinski-Harabasz Index[12] which maximize his equation to estimate the number K, Gap Statistic[69],Akaike Information Criterion(AIC) [77], Bayesian Information Criterion(BIC) [53], Duda and Hart[22], Silhouette Coefficient [33], Newman and Girvan[56].

As other clustering algorithms, K-Means performance can be improved by using embedding techniques. For example, the results of using UMAP with K-Means were prom promising, it improved the accuracy and the time significantly.[3]

Fuzzy C-Means algorithm

This algorithms is similar to K-Means, it has the capability to use the concepts of fuzzy clustering; one data point can belong to one cluster or more, it was developed in 1973 by J.C Dun.

Fuzzy C-Means can be formulated as an optimization problem, it minimize the following cost function:

$$J_{FCM} = \sum_{j=1}^{k} \sum_{i=1}^{n} w_{ij}^{p} \|x_i - c_k\|^2 \tag{3.3}$$

where W is the the partition matrix.
$w_{ij} \in [0, 1]$ represent the degree of membership of point i in cluster j.
p is the fuzziness exponent.

The steps of this algorithm are described as follows:

Algorithm 5: Fuzzy C-Means algorithm

```
1 initialization(W) ;
2 Calculate the centroid of each clustering ;        // $c_j = \frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} w_{ij}}$
3 Update the fuzzy partition ;        // $w_{ij} = \frac{\|x_i - c_j\|^{\frac{-2}{p-1}}}{\sum_{k=1}^{K} \|x_i - c_k\|^{-2/p-1}}$
4 repeat the steps 2 and 3 until the centroids don't change;
```

3.2.2 Hierarchical Clustering

In contracts to Partitional Clustering,Hierarchical clustering algorithms do not need to define the number of clusters. we construct the clusters in hierarchical clustering through a recursive process that divides the data with many available levels. In other words, that give us a tree of clusters, each node represent cluster which can be divided into other sub-clusters where we use dendrogram Fig(3.2) to explore clustering process of each level.

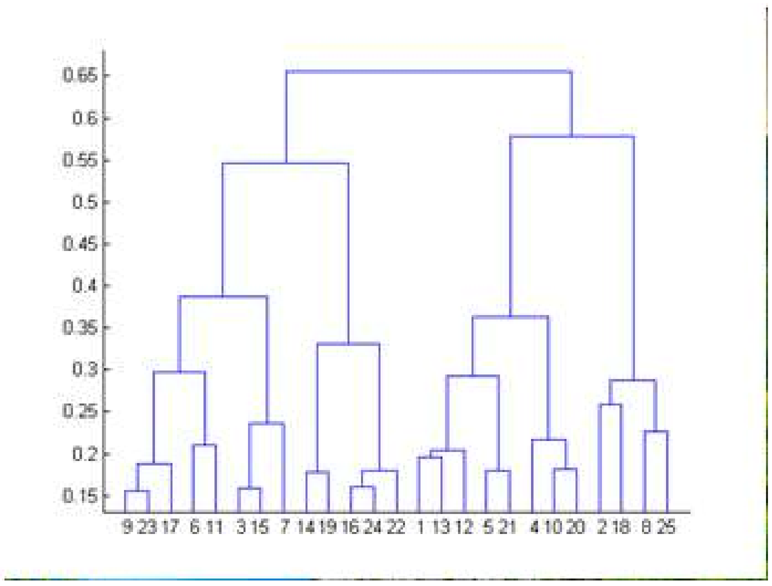


Figure 3.2: Dendrogram[71]

These methods are divided into two main types: agglomerative(bottom-up) and divisive(top-down).

Agglomerative clustering

In the begin of agglomerative clustering we represent each point in the data as a cluster which called singleton, then we merge the most similar clusters and they become one cluster at the next level, we repeat the merger until we have a single cluster containing the entire data.

The basic algorithm of agglomerative hierarchical clustering is summarized as follows:

Algorithm 6: Agglomerative clustering algorithm

```
1 Represent each point as cluster;
2 Calculate the proximity matrix;
3 while More than one cluster remaining do
4     Merge the closest two clusters;
5     Update the proximity matrix;
```

Divisive clustering

In contracts to agglomerative clustering ,we start with one cluster that contains all points, we split it into two clusters such that the split process ensure the similarity between the data points in the same cluster obtained, and the dissimilarity between the points which do not belong to the same cluster,we repeat this process until we obtain a number of clusters equal to the number of data points.

Measure of distance

Calculating the proximity matrix is one of the most important steps in hierarchical clustering, and it is repeated at every level. this is done by calculating the distance between each two clusters. When we say 'the distance', Euclidean distance comes to mind, but in reality there are many functions or formulas to represent the distance. In the following table, we tried to group some of them:

Table 3.1: Some distance metrics

Euclidean Distance	$D(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
Squared Euclidean Distance	$D(X,Y) = \sum_{i=1}^{n}(x_i - y_i)^2$
Manhattan Distance	$D(X,Y) = \sum_{i=1}^{n}\lvert x_i - y_i\rvert$
Mahalanobis Distance	$D(X,Y) = \sqrt{(X-Y)^T S^{-1}(X-Y)}$ Where S is covariance matrix
Minkowski Distance	$D(X,Y) = (\sum_{i=1}^{n}\lvert x_i - y_i\rvert^p)^{\frac{1}{p}}$
Chebyshev distance	$D(X,Y) = \max_i(\lvert x_i - y_i\rvert)$
Bray Curties distance	$D(X,Y) = \frac{\sum_{i=1}^{n}\lvert x_i - y_i\rvert}{\sum_{i=1}^{n}\lvert x_i + y_i\rvert}$
Canberra Distance	$D(X,Y) = \frac{\sum_{i=1}^{n}\lvert x_i - y_i\rvert}{\sum_{i=1}^{n}\lvert x_i\rvert + \lvert y_i\rvert}$

Linkage Criteria

We know that a cluster may contain more than one point , and that the distance function needs just two points to calculate the distance between them, the linkage criteria came to define these two points in hierarchical clustering, For example *single linkage* chooses the minimum distance, and *complete linkage* chooses the maximum distance,..,etc. some of these criteria are represented in the Fig()and in the next table:

Table 3.2: formulas of some linkage criteria

Single linkage	$D(C_l,(C_i,C_j)) = \min((C_l,C_i),(C_l,C_j))$
Complete linkage	$D(C_l,(C_i,C_j)) = \max((C_l,C_i),(C_l,C_j))$
Group average linkage	$D(C_l,(C_i,C_j)) = \frac{1}{2}((C_l,C_i)+(C_l,C_j))$
Weighted average linkage	$D(C_l,(C_i,C_j)) = \dfrac{n_i}{n_i+n_j}D(C_l,C_i) + \dfrac{n_i}{n_i+n_j}D(C_l,C_j);$ $n_i = \lvert C_i \rvert$= the number of the data points in the cluster C_i
Centroid linkage	$D(C_l,(C_i,C_j)) = \lVert m_l - m_{ij} \rVert^2$ $m_l = \frac{1}{n_l}\sum_{x\in C_l} x$
Median linkage	$D(C_l,(C_i,C_j)) = \frac{1}{2}(C_l,C_i)+\frac{1}{2}(C_l,C_j)-\frac{1}{4}(C_i,C_j)$

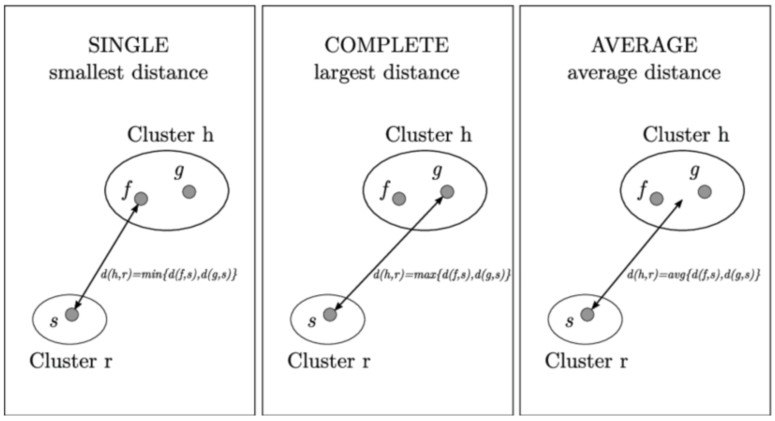


Figure 3.3: Descriptive drawing of some linkage criteria[68]

3.3 Applications

In data mining, we use the clustering for two main reasons: first we use it as a pre-processing step for the other algorithms as extract the features, secondly we use it as an independent tools to cluster the data for descriptive purpose and to focus on the importance cluster.[63]

Next, we illustrate an examples of applying clustering in Image segmentation. Also, we discuss its use in Marketing and Document analysis.

3.3.1 Image segmentation

Segmentation can be described as a process of dividing the image into multiple segments or regions. Different approaches have been used to segment images, including clustering. Image segmentation is useful for different ends including Object detectioin [21], Object recognition [1, 2], etc.

Let's see an example[62] where we use k-means in segmentation of images. We have Fig(3.4), it has three main colors: Green that represents the color of trees and weeds, blue is the color of the lake and the orange is the color of the sky.

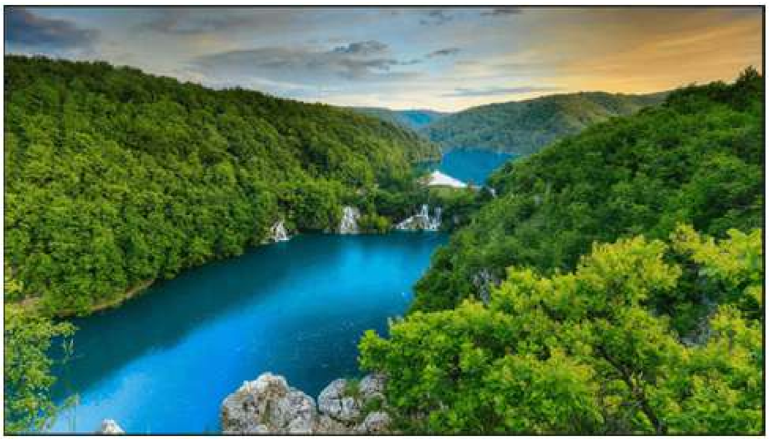


Figure 3.4: A landscape contains three main colors[62]

The k parameter in this example is 3 and each pixel in the image is a data point. After we applied K-Means algorithm on this image, we obtained Fig(3.5).

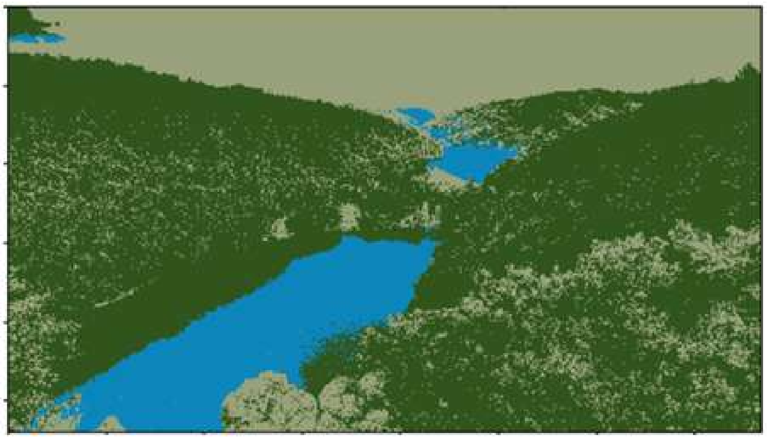


Figure 3.5: Clustering result with 3 clusters[62]

The results seem very acceptable.

3.3.2 Image and document retrieval

Document retrieval is the process consisting of retrieving documents relevant to a query formulated by the user [41]. Those documents may be textual, images in the case of image retrieval, audio and video documents, or composite ones. Document retrieval has witnessed a great advancement in the last two decades. This has been stimulated by the rapid increasing of the size of existing datasets and the huge number of documents available on the Internet [37].

In general, features are first extracted from images and documents [35, 34, 36], then used to compare them using different similarity measures [17]. In the case of large datasets, this comparison may take too much time which makes the retrieval process very slow. This is precisely where clustering can intervene. It helps organizing large image datasets and document collections, thereby allowing to make image and document retrieval quicker, by limiting the retrieval space to the images/documents belonging to the same cluster as the query and eventually neighbouring clusters.

3.3.3 Marketing

In Marketing, clustering can be used to cluster the consumers into groups, each group has a similar needs and wants, this allows businesses to introduce tailored marketing strategies highlighting the most desirable advantages and goods for each customer group. Also, it is possible to select comparable towns to test various marketing strategies by grouping cities into homogeneous clusters.[65]

Marketing generally uses any profit-enhancing tool, and clustering is one such tool.

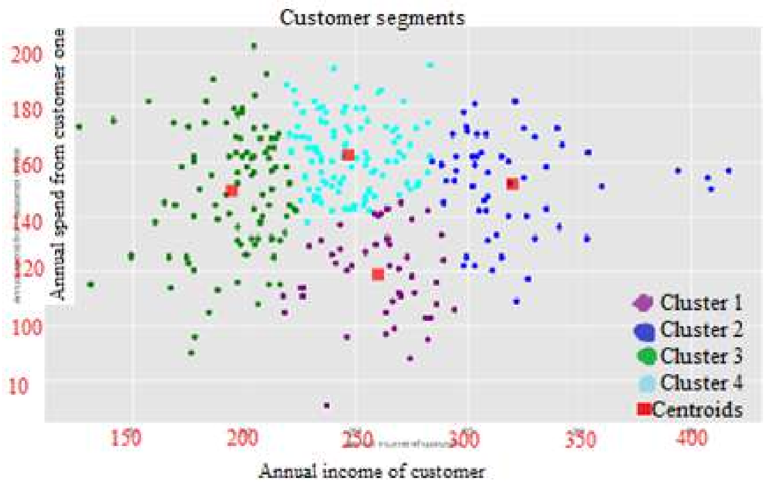


Figure 3.6: An illustrative example represents the relationship between a customer's income and spending, there are four main clusters [74]

3.3.4 Documents Analysis

In our daily lives, we need to do a document analysis for a number of reasons, and sometimes we want to do it quickly and efficiently.

The problem here is with time, and document analysis needs to understand and compare content.

But the clustering can solve this problem, it takes the text and groups it on many themes(topics), and quickly puts the similar documents in the same cluster .

Chapter 4

NMF For Clustering

In the previous chapters, we have seen that NMF can be apply for many tasks such as image processing and document analysis. Moreover, NMF can be considered as a clustering algorithm and that is the topic of this chapter.

The theoretical side shows that NMF is equivalent to K-Means algorithm with least square cost function and it is equivalent to PLSA with KL-divergence. This chapter looks at these two equations and more.

Firstly, we will give an empirical proof that NMF has capability for clustering, then we illustrate the equivalence between NMF and some clustering algorithms including K-Means, Kernel-KMeans and Probabilistic Latent Semantic Analysis(PLSA).

4.1 Empirical proof

To demonstrate the NMF has a capability to clustering we show the next example, NMF here was applied on ORL data-set.

ORL dataset: is a dataset of faces Fig(4.1), it contains 400 images 112x92 of 10 persons; each person has 10 images, where the images are different (time, lighting, facial expression, position).[73]

In Fig(4.2) we note that NMF could to compute the factors $F = f1, f2, ..., f25$, these images represent the centroids of our data-set, and that means NMF has a capability to clustering.

Figure 4.1: Sample of ORL dataset[23]

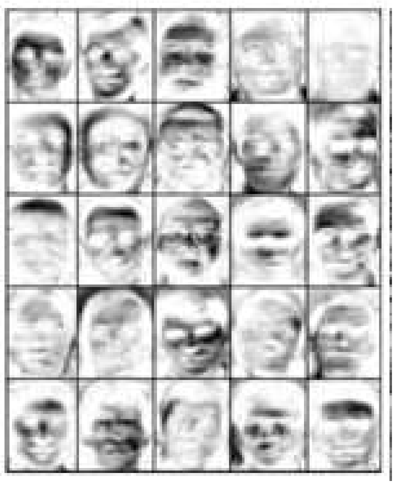


Figure 4.2: Computed NMF factors F=f1,f2,..,f25[72]

4.2 NMF and K-Means

We have talked about the K-means algorithm in previous chapter, where did we say if we have $X = \{x_1, x_2, ..., x_n\}$, K-means divides them into K clusters $C = (c_1, c_2, .., c_k)$, iteratively KMeans minimizes the next cost function:

$$SSE = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - c_k\|^2 \tag{4.1}$$

Furthermore, One of NMF cost function is the sum of squares errors as following:

$$f(\mathbf{W}, \mathbf{H}) = \|\mathbf{X} - \mathbf{WH}\|_F^2 \tag{4.2}$$

Let H is the cluster indicator, $h_{ki} = 1$ if the point x_i belongs to the cluster c_k, $h_{ki} = 0$ else. That means the k-means cost function can be written as:

$$SSE = \sum_{i=1}^{n} \sum_{k=1}^{K} h_{ki} \|x_i - c_k\|^2 = \|X - CH\|^2 \tag{4.3}$$

As we note, NMF and k-means minimize the same cost function, but with different constraints:

- KMeans: $H \in \{0, 1\}^{kn}$

- NMF: $W \geq 0, H \geq 0$

4.3 NMF and Kernel K-Means

Kernel K-Means algorithm is derived by K-Means, the main advantage of this algorithm is its ability to run on non-linear data with the assistance of kernel functions (Polynomial kernel, Gaussian kernel, sigmoid kernel).[55]

We have a special case of NMF called SNMF (Symmetric NMF), where we can write $W = HH^T$ s.t $HH^T = I$, the cost function of SNMF can be written as:

$$f(\mathbf{W}, \mathbf{H}) = \|\mathbf{X} - \mathbf{H}\mathbf{H}^T\|_F^2 \tag{4.4}$$

According to Chris Ding and its group[19], SNMF and Kernel K-Means have an equivalent cost functions.

4.4 NMF and Probabilistic Latent Semantic Analysis

There is many techniques used in topic modelling and PLSA is one of them, PLSA was developed by Th. Hofmann in 1999[28], it is based on the assumption that each document d is consist of mixing of topics, and each topic z is consist of a collection of words, also we note that the order of the words is not taken into account .Our data in PLCA can be represented by 3 sets of variables: $d_i \in D\left\{documents\right\}, w_i \in W\left\{words\right\}, z_i \in Z\left\{topics\right\}$ such that the number of topics is specified by us.

Let F is the matrix of our data, it consist of n documents and m words, and $F_{ij} = F(w_i, d_j)$ is the number of times(frequency) the word i appears in the document j (we can scale the data by $F_{ij} \leftarrow F_{ij}/T$ where T is the total number of words and $F_{ij} = p(w_i, d_j)$). PLSI maximize the next likelihood:

$$J_{PLSI} = \sum_{i=1}^{m} \sum_{j=1}^{n} F_{ij} \log P(w_i, d_j) \tag{4.5}$$

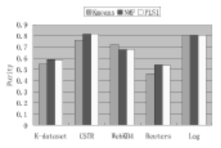
$$P(w_i, d_j) = \sum\nolimits_k p(w_i/z_k)p(z_k)p(d_j/z_k) \tag{4.6}$$

In this paper [20], the authors present theorem to prove that NMF and PLSA have similarity, where they proved that PLSI and NMF have an equivalent objective functions.

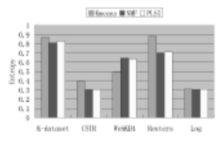
$$max J_{PLSA} \Leftrightarrow min J_{NMF} \tag{4.7}$$

In other meaning, any solution of PLSI likelihood is a solution of NMF with KL-divergence objective function .

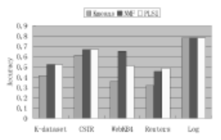
Then, they compared the performance of the two techniques for 5 data-sets : CSTR, WebKB, Log, Reuters and WebAce, and they used 4 criteria to measure the performance: accuracy, entropy, purity and adjust rand index, the results are established in Fig(4.3):



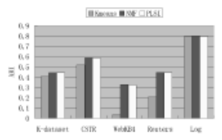
(a) Purity



(b) Entropy



(c) Aaccuarcy



(d) Adjust Rand Index

Figure 4.3: Performance comparison of NMF and PLSA

The results was very well, we observe that performance of NMF and PLSI are too close together, the difference is very small in all cases.

These papers [25, 20] show more details about the relation between NMF and PLSA.

Chapter 5

Simulated annealing ONMF for a better clustering

In section 2.3, we briefly described a number of existing algorithms for the non-negative matrix factorization such as MU of Lee and Seung, as far as we know, all those interested in NMF apply it on the data as a whole and may ignore that data sets may have some problems such as presence of noise in the data and the overlap between the cluster, most of the time, this reduces the performance of NMF for clustering as it does with the rest of clustering algorithms. In this chapter, we try to solve this problem where we present a new method to increase NMF performance for clustering with the help of simulated annealing algorithm and we describe the main steps of this algorithm, then we report the obtained results on many data sets and analyze these results.

5.1 Background

Our method is based on two algorithms; ONMF and simulated annealing, for that, before we present our method we have first to illustrate these two algorithms.

5.1.1 Orthogonal NMF

Orthogonal Non negative matrix factorization (ONMF) is an extension of the standard NMF with an additional constraint of orthogonality. This constraint is applied for one of the two factors W or H. Mathematically, ONFM can be written as:

$$X \approx WHs.t(WW^T = IorHH^T = I) \tag{5.1}$$

Were I is the identity matrix.

ONMF has a strong relation with K-Means as we showed in Sec.(4.2). Generally, ONMF had proven its worth in clustering more than standard NMF [14].

Seungjin Choi[14] has updated the standard multiplicative update rules of Lee and Seung to be able to solve ONMF with its new orthogonality constraint.

The suggested multiplicative update rules when the orthogonality constraint is imposed on W matrix are given as:

$$H \leftarrow H \odot \frac{(W^T X)}{(W^T W H)} \tag{5.2}$$

$$W \leftarrow W \odot \frac{(X H^T)}{(W H X^T W)} \tag{5.3}$$

Where $A \odot B$ is the Hadamard Product (Elementwise product) and $\frac{A}{B}$ is the element-wise division.

5.1.2 Simulated annealing

It is a Metaheuristic algorithm used to optimize the cost function, i.e., approximate to global optimum, when there are several local optima in the space of solution. This algorithm has been inspired from annealing in metallurgy. It has been used to solve the knapsack problem and the traveling salesman problem and many other problems.

The idea of simulated annealing is to generate many solutions in the system. First, it randomly chooses the solution and attempts to optimize it sync with the decrease in temperature variable (it takes large values at the beginning). in each iteration, it chooses a neighbor solution to the current one. According to its score and the temperature of the system, the solution may be accepted or not. The fig(5.1) illustrates the process of updating the current solution in the simulated annealing.

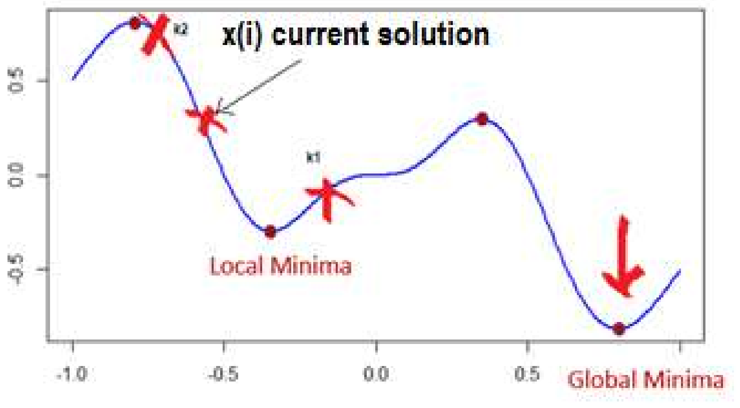


Figure 5.1: Select a new solution in simulated annealing

In Fig(5.1) the current solution selected by simulated annealing is $X(i)$, but $X(i)$ is not the optimal solution, the system moves , if $X(i+1)$ is better then $X(i)$ it is always accepted $(k1)$. Else, if $X(i+1)$ is not better then it may be accepted $(k2)$, the possibility of accepting $exp(-\Delta/t)$ is related to the current temperature of system as well as the difference between the current score and the previous one. This means that bad solutions have greater probability to accept at the beginning of simulated annealing or when the difference between scores is very small.

5.2 Relevant works

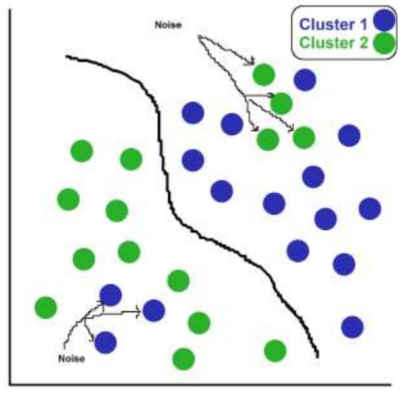
As stated in chapter 7, NMF has the ability to cluster the data, we had seen its relationship with some clustering techniques as K-Means and PLSA and as we know, NMF has many variants , some of them is doing very well for clustering. Sparse NMF (SNMF) is an NMF with a sparsity constraint, this constraint is imposed on the second matrix H, we can get that easily by adding a regularization to the cost function. Projective NMF (PNMF) is another NMF variant, which tries to get a basis vectors can learn localized representations and it has a strong relationship with K-Means clustering. We have also Non negative Spectral Clustering or NSC which aim to treat spectral clustering problem and from its advantages the ability to use kernel functions. All of these variants are reviewed in [70].

In other side, metaheuristic methods can be integrated with NMF. For example, in [32] the authors used 5 population-based algorithms, which are: Particle Swarm Optimization, Differential Evolution, Fish School Search and Fireworks Algorithm to set a good initialization for NMF and the authors of[31] explored 5 nature-inspired optimization algorithms, called: Genetic Algorithms, Particle Swarm Optimization, Differential Evolution, Fish School Search and Fireworks Algorithm to improve the multiplicative update algorithm.

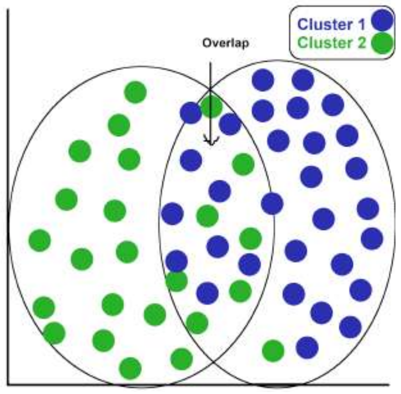
Our presented algorithm is however radically different from the works mentioned above, where we aim to improve NMF performance for clustering problem by using a metaheuristic method which is simulated annealing.

5.3 Methodology

In existing works, NMF was applied to the dataset as a whole. This can cause different problems to clustering. First and foremost, the dataset may contain noise, which will negatively affect the clustering results. The second problem is overlapping clusters, which makes separating clusters difficult.



(a) noisy data



(b) Overlapping

Figure 5.2: Some problems that reduce the clustering quality

To face those problems, we propose here to apply NMF locally, i.e. subset by subset. The First subset is chosen randomly,and in the next iteration, we have to choose a neighbor subset to the current set. The NMF is applied to each subset separately. We measure the quality (energy) of the obtained local clusterings. This quality allows us to decide which clustering to retain or discard, as well as the importance we assign to it. We only keep factorizations which have obtained high energy and reject others. This is accomplished thanks to the SA algorithm. At the end we will obtain a set of factorizations (i.e. matrices H and their corresponding W). We combine them to obtain the optimal H* which will be used to cluster the whole dataset. The same thing is done for W.

The core idea of our method can be summarized as follow:

- Each time, randomly choose a subset of our dataset. This subset may be noisy or not, but we don't know this before.
- Perform ONMF on this subset.
- Cluster this subset using the encoding matrix H generated by ONMF.
- Evaluate this clustering using Clustering validity indices. IF the score is greater than the previous one then the factorization will be retained with a weight equal to 1. Else We accept it with a Boltzmann's probability and assign it as a weight.
- At the end, the retained clusterings(the partial NMFs) are combined to get the overall clustering.

5.4 Results and discussion

In this section we show the performance of SA-ONMF on different data-sets, and compare it with Orthogonal NMF (ONMF). After that we discuss these results. By doing this, we can give the final judgment whether this algorithm is effective or not.

5.4.1 Evaluation metrics

To measure the clustering performance, we used three metrics, Davies-Bouldin index, Silhouette index and Calinski-Harabasz index as our performance measures. The common denominator between them is that they are used when the truth labels are unknown like our case. All these metrics are implemented in SKlearn[1] library.

Davies-Bouldin index

This metric describes the similarity between the clusters. The Davies-Bouldin score of the entire clustering is computed as:

$$DB = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j} R_{ij} \tag{5.4}$$

Where:

[1]https://scikit-learn.org/stable/modules/clustering.html

R_{ij}: is the similarity between cluster C_i and cluster C_j, it is defined as:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \tag{5.5}$$

s_i: is a measure of scatter within the cluster C_i, and represent the average of distance between the centroid of the cluster C_i and each point of that cluster.

d_{ij}: is a measure of separation between the centroid of C_i and C_j.

On Davies-Bouldin index, the lowest possible value is zero and the smaller Davies-Bouldin value, the better the clustering quality is.

Silhouette index

This index tells us if a data point is assigned to a suitable cluster or not. The silhouette index for a single point is defined as:

$$s = \frac{b - a}{\max(a, b)} \tag{5.6}$$

Where:

a: average distance between this point and all other points in the same cluster.

b: average distance between this points and all points in the nearest cluster.

Silhouette index takes values between -1 and 1, the values near 1 means the data points is assigned to a suitable cluster and the opposite is true for -1, 0 indicates overlap between the clusters.

If we wanted to calculate silhouette index for a complete sample then we have to calculate the mean $s(i)$ for each point i in this sample.

Calinski-Harabasz index

Calinski-Harabasz score is defined as the relationship between the within-cluster dispersion and the between-cluster dispersion. it is given as:

$$s = \frac{tr(B_k)}{tr(W_k)} \frac{n_E - k}{k - 1} \tag{5.7}$$

Where:

k: the number of cluster.

n_E: the size of entire data E. B_k: the between group dispersion matrix.

W_k: the within-cluster dispersion matrix.

tr: trace of matrix.

B_k and W_k are defined by:

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_k} (x - c_q)(x - c_q)^T \tag{5.8}$$

$$B_k = \sum_{q=1}^{k} |C_q|(c_q - c_E)(c_q - c_E)^T \tag{5.9}$$

C_q:set of points in cluster q.

c_q: the center of cluster q.

c_E:the center of entire data E.

This index takes values between 0 and $+\infty$ andthe larger the values of Calinski-Harabasz index, the better the clustering solution is.

5.4.2 Data-sets description

The SA-ONMF evaluation was performed on data sets with a different number of clusers and features. We used 03 real life data sets, Iris dataset, Breast cancer dataset, Wine dataset, and we used Blobs dataset which is unreal life data set. All these data sets are available on SKlearn [2] library.

Iris dataset:

This data set contains 150 instances of flowers distributed in 3 classes(ris-Setosa,Iris-Versicolour and Iris-Virginica), each class has 50 instances i.e. 33.33% from the total number. A flower in this database is defined by four numerical properties; The length and width of root as well as petal. Also, there are no missing values in this data-sets.

[2]https://scikit-learn.org/stable/datasets/index.html

it was created by R.A. Fisher in 1988.

Breast cancer dataset:

It is a breast cancer diagnostic data-set provided by 3 Wisconsin university researchers: Dr. William H. Wolberg, W. Nick Street and Olvi L. Mangasarian in 1995. it has 569 instances, 212 of them were classified as having malignant cancer and the remaining cases (357) have benign cancer. In this data-set , each instance is defined by 30 numerical values(radius: mean of distances from center to points on the perimeter, texture: standard deviation of gray-scale values, perimete, area, smoothness,compactness ,...,etc). This data-set has no missing values.

Wine dataset:

It has 178 samples of wine , grouped into 3 classes (class_0, class_1 and class_2). This data-set is described by 13 numerical properties: Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium,..,etc. The first class has 59 samples, the second has 71 samples and the third contains 48 samples, and there are no missing value. It was created in 1988 by R.A. Fisher. **Blob data-set:**

5.4.3 Experimental results and discussion

To know if ONMF is effective or not when we make it guided by simulated annealing, we had calculated the performance of both ONMF and SA-ONMF. The result of the two algorithms on the real-life datasets are reported in Table 6.1 where we calculated the mean and standard deviation of 1000 separate runs. In the most of cases SA-ONMFS appears that it is better than ONMF.

Table 5.1: Performance results on real-life datasets.

	Methods	**Iris**	**BreatCancer**	**Wine**
Davies	**ONMF**	1.23±0.38	2.33±0.22	1.95±0.28
	SA-ONMF	**0.68**±0.20	**1.89**±0.01	**1.92**±0.05
Silhouette	**ONMF**	0.36±0.09	0.13±0.02	0.16±0.039
	SA-ONMF	**0.53**±0.09	**0.19**±0.00	**0.17**±0.00
Calinski	**ONMF**	152.09±34.94	79.02±13.65	**36.02**±6.38
	SA-ONMF	**213**.50±48.37	**86.03**±2.24	35.97±0.96

Additionally, to demonstrate the effectiveness of SA-ONMF compared to ONMF, we investigated the relationship between the performance of each method and cluster overlap rate in each data set.

Let X is a given dataset. We computed the cluster overlap rate as:

$$OverlapRate = \frac{1}{|X|}\sum_{x \in X} error(x) \tag{5.10}$$

Where the error function is given by:

$$error(x) = \begin{cases} 1, & \text{if x and its nearest neighbor belong to the same class.} \\ 0, & \text{else.} \end{cases} \tag{5.11}$$

In other hand, we computed the gap between the performance of the two methods for each metrics as:

$$\text{PerformanceGap=Performance(SA-ONMF)-Performance(ONMF)} \tag{5.12}$$

The results are reported in Table(5.2).

Table 5.2: Performance gap between SA-ONMF and ONMF.

	Iris	**BreastCancer**	**Wine**
	Overlap rate (%)		
	5.3	4.9	4.4
	Performance gap		
Davies	0.6	0.4	0.0
Silhoutte	0.2	0.1	0.0
Calinski	61.4	7.0	-0.1

The results show that there is a positive correlation between the performance gap and the overlap rate; the higher the overlap rate, the more performance gap , and vice versa. In other words, our method becomes more useful for clustering when the data contain noise and/or cluster overlap.

In other experiment, we wanted to show if our algorithm has sensitive to the number of cluster or not, for that, we reported the results of SA-ONMF performance for different numbers of clusters. As we see in Table 6.3, our algorithm is not sensitive to the number of clusters, it always performs greater than the standard ONMF.

Table 5.3: Performance results on Blob-like datasets.

		Blob-like datasets ($f = 9$, $\sigma = 0.5$)		
	Methods	$c = 3$	$c = 5$	$c = 7$
Davies	**ONMF**	0.27±0.25	0.81±0.27	1.36±0.44
	SA-ONMF	**0.21**±0.00	**0.67**±0.01	**1.03**±0.19
Silhouette	**ONMF**	0.82±0.12	0.56±0.15	0.36±0.15
	SA-ONMF	**0.85**±0.00	**0.67**±0.01	**0.52**±0.08
Calinski	**ONMF**	14460±4278	940.80±401.10	399.30±161.20
	SA-ONMF	**15920**±853	**1116**.0±32.2	**533**.90±134.30

In our algorithm we weigh each factorization(W,H) with the appropriate weight, the weight must be between 0 and 1, but the effect of the weighting process does not appear clearly.

We did some stats to see why the weighting effect doesn't appear, and we found: The number total of matrices(iterations): 110 matrices Among them:

- 16 matrices have weight between 1 and 0.75; Although their number is large (14% of the total number)but their affect will not appear because they have a very good weight(close to 1).
- 2 matrices have weight between 0.75 and 0.5 ; good weigh and a few matrices(1.8 % of the total number) =¿ Almost no effect.
- 1 matrix has weight between 0.5 and 0.25 ; bad weight but a few matrices (less than 0.9% of the total number).
- 1 matrix has weight less than 0.25 ; very bad weight but a few matrices (less than 0.9 % of the total number).

The number of the weighted matrices is small(17 % of all matrices) and that is very logical because it is subject to simulated annealing algorithm.

we say: *"The weighting effect will only appear If we have too many matrices with very bad weights".*

"In theory, weighting is very acceptable, and it's difficult in practice to appear its affect", we added.

5.5 Implementation details

We implemented our algorithm in Python and we used Scikit-Learn as a helper library .

5.5.1 Python

we used python(v3.6.1) language to implement our code. Python has many helpful packages including numpy, matplotlib, seaborn,...,etc. To use python, we need:
Python package, it is available on: `https://www.python.org/downloads/`
Python IDE, many IDEs are available like :

- PyCharm: `https://www.jetbrains.com/pycharm/`
- Spyder: `https://www.spyder-ide.org/`

Google provide an online environment With attractive features: `https://colab.research.google.com/`

5.5.2 Scikit-learn library

It is an important library to work on machine learning, we used it for many tasks as:

- loading some ready-made datasets including Wine dataset, BreastCancer dataset, Iris dataset.
- measuring the clustering performance with many criteria(Davies-Bouldin index, Silhouette index, Calinski-Harabasz index).

The official website provides a description of modules and some examples: `https://scikit-learn.org/`

Chapter 6

NMF And AutoEncoder

NMF is really deep related with Machine learning. Previously, we had seen it capable of dealing with clustering problems. Some types of neural networks can also do clustering such as Autoencoder, So in this chapter we will investigate the relationship of NMF with autoencoders. The purpose of this chapter is not to provide a detailed overview of all the works that focus on the value of NMF with autoencoder – and we apologize for not listing many of the current works – but rather to serve as an introduction to the relationship between NMF and autoencoder. This has not been explored further , and we have not made a contribution, but merely a preliminary exploration.

We will open this chapter by giving a brief overview of the autoencoder, its components, its cots function and its types. Then, we illustrate the relationship between it and NMF.

6.1 Autoencoder

As it is known, Artificial Neural networks(ANN) are among the hot topics in machine learning. The majority of ANNs are classified as supervised algorithms, but there is ones are supervised algorithms as the autoencoder. This section is introduction to autoencoder and its types.

6.1.1 Introduction

An autoencoder is unsupervised neural network, which converts the data into compressed form and then tries to decoding this data into the original one, and aims to set its output to be equal to its input.

As shown in [Figure 6.1], An autoencoder generally consists of three main components: encoder, code and decoder.

- Encoder: It is the part responsible for compress data to latent space representation.
- Code: represents the input data in its compressed form that is fed to the decoder.
- Decoder: From its name, it looks like a reverse process of encoder.this part reconstruct the input from the latent space representation to be equal to the original input.

Stemming from the above, an autoencoder can be defined by two transitions $T1$ and $T2$:
$T1 : X \longrightarrow H$
$T2 : H \longrightarrow X$
$T1, T2 : argmin_{T1,T2} \parallel X - (T1 \cup T2)X \parallel$

The loss function of autoencoder can be defined by two terms: the first is a function L which describes the dissimilarity between input data x and output data $\hat{x}$, the second is a regularization term to prevent overfitting problem.

$$Obj = L(x, \hat{x}) + regularizer \tag{6.1}$$

The previous loss function can be optimized by many methods, the stochastic gradient descent (SGD) is one of them.[79]

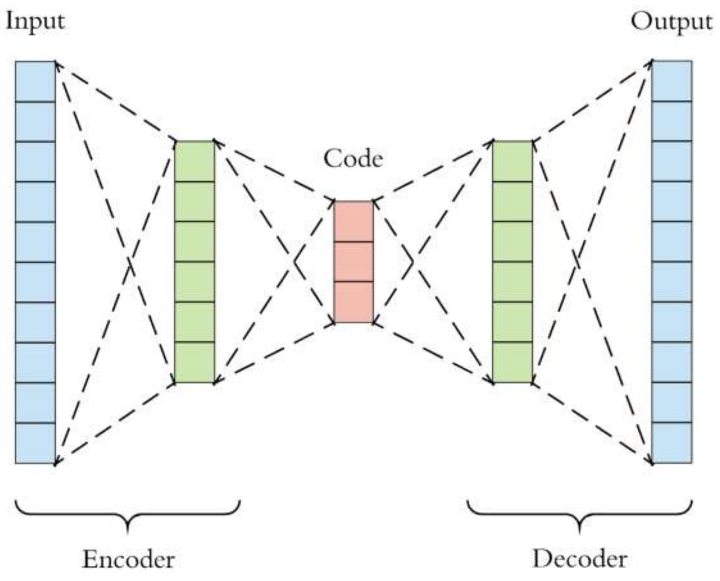


Figure 6.1: Structure of a basic autoencoder[18]

6.1.2 Types of autoencoders

The autoencoder has many types, among them: Sparse Autoencoder, Denoising Autoencoder and Variational Autoencoder.

Sparse Autoencoder

Sparse autoencoder is like a standard autoencoder with the addition of a sparsity penalty.By sparsity, we mean encouraging a few node to activate.[79] The sparsity constraint can be imposed with two different ways: L1 regularization or KL-Divergence.

The loss function with L1 regularization is defined as:

$$Obj = L(x, \hat{x}) + \lambda \sum_i \mid a_i^{(h)} \mid \tag{6.2}$$

where:
a: is the activation value.
h: is the index of the layer.
i: is the index of the observation
λ: scaling parameter

when we using KL-divergence, the loss function is given by:

$$Obj = L(x, \hat{x}) + \sum_i KL(p \| \hat{p}) \tag{6.3}$$

such than:

$$\hat{p}_j = \frac{1}{m} \sum_i [a_i^{(h)}(x)] \tag{6.4}$$

where:
m: the number of the observations.
p: is Bernoulli random variable distribution

Denoising Autoencoder

Denoising autoencoder(DAE) is a modified autoencoder to solve the problem of identity function when the output is exactly equal to the input(it is given by: $f(x) = x$).

To solve the problem of identity function, denoising autoencoder corrupts the original data. This is easily done by using a mapping function that returns the values of some nodes to zeros(E.g. 50% or 30% of the total number of nodes).

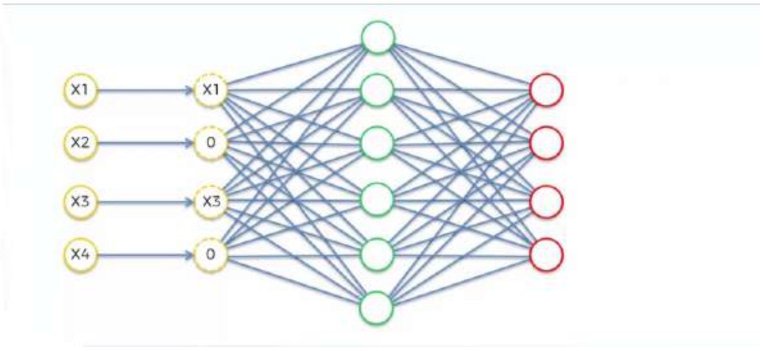


Figure 6.2: Architecture of denoising autoencoder[54]

When we calculating the loss function, we should compare the output with the original data and not with the corrupted one.

Variational autoencoder

The standard autoencoder do not produce a probability model and do not produce a deterministic model and although it encodes the data into a lower dimensional representation, but it cannot generate a new data from this lower dimension space.

The variational autoencoder came to solve these two problems, meaning it can produce a probability model which allow us to generate new data points. The meaning of all this is that VAE is considered a generative model .

6.2 Autoencoder for NMF

6.2.1 Introduction

Autoencoder can perform NMF by adding extra constraints, that was presented in many paper [5, 29, 45]. In these lines, we discuss how autoencoder can perform NMF by matching both structures of NMF and autoencoder.

Fig(6.3) provides a representation of a single-layer autoencoder to perform NMF. The input layer is fed by data point $v \in R^{m1}$. We produce the latent representation $h \in R^{r1}$ by linear or nonlinear function $f(W_1 v)$ where non-negativity constraint are respected. In other words, the output of the chosen function must be non negative. W_1 is the weights of the first layer.Furthermore, autoencoder can be contain multi-layer, which allow us to define more layer between the latent representation h and the input layer, in order for our autoencoder to become more deep. The final set of weights $W_f^{mr}$is always non negative while the autoencoder are training with activation function such that $x = W_f h$. The autoencoder is trained for $x \approx v$.[67]

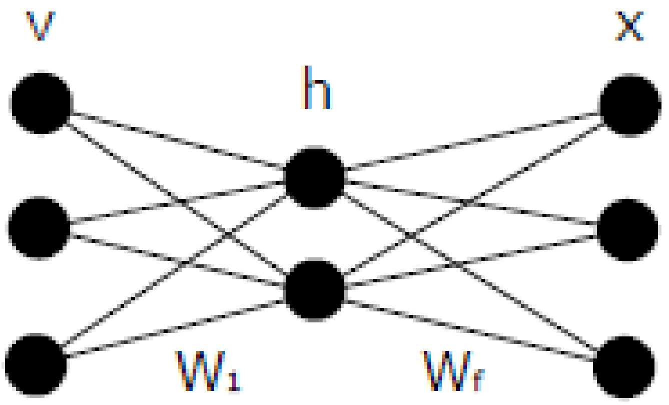


Figure 6.3: An autoencoder with one layer performs NMF

6.2.2 Some works

The authors of [67] proposed a model combines between the Variational AutoEncoder (VAE) and Non Negative matrix factorization (NMF), it called PAE-NMF. the idea behind their model is to benefit from the advantages of the both techniques AE and NMF. For example, they will have a representation more sparse and more interpretable. Among the advantages of PAE-NMF, its ability to extract meaningful features without overfitting

In [29], the authors trained a deep sparse autoencoder with non negative constraint (NCAE), they forced their autoencoder to learn part-based representation by imposing the non negativity constraint on the weights W. its loss function is given by:

$$J_{NSAE}(W,b) = J_E(W,b) + \beta J_{KL}(p\|\hat{p}) + \frac{\alpha}{2}\sum_{l=1}^{2}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}} f(w_{ij}^{(l)}) \tag{6.5}$$

Where:

$$f(w_{ij}) = \begin{cases} w_{ij}^2 & w_{ij} < 0 \\ 0 & w_{ij} \geq 0 \end{cases} \tag{6.6}$$

β and α: control the second and the third term, respectively, and $\alpha \geq 0$
s_l and s_{l+1}:are the sizes of adjacent layers.

In backpropagation, the weights W and biases b can be calculated with the help of gradient descent as follows:

$$w_{ij}^{(l)} = w_{ij}^{(l)} - \eta \frac{\partial}{\partial w_{ij}^{(l)}} J_{\text{NCAE}}(\mathbf{W}, \mathbf{b}) \tag{6.7}$$

$$b_i^{(l)} = b_i^{(l)} - \eta \frac{\partial}{\partial b_i^{(l)}} J_{\text{NCAE}}(\mathbf{W}, \mathbf{b}) \tag{6.8}$$

where $\eta > 0$ is the learning rate.

The experimental results showed that the non-negativity constraint forced NCAE to learn meaningful features (part-bast representation) and the reconstruction error of NCAE is smaller than Sparse AE's and NMF's.

In other paper[9], NMF is combined with denoising autoencoder, the model maps the data into a lower space after that optimizes the clustering objective. It could to reduce non-linear dimension, also, it improved the clustering performance.

6.2.3 future research

In this topic, there are many future research directions including:

- studying the ways of imposing the non negativity constraint on autoencoder, where some methods are suspicious as when we use $a = max(0, a)$ to make the values of nodes non negative, in fact, the negative values are removed but some informations were deleted along with it.
- Developing a novel initialization method Using autoencoder for the algorithm which need that as Lee and Seung MU, we always say: 'The better the initialization, the faster and better approximation'.

Chapter 7

Conclusion

This thesis highlighted a Non-Negative Matrix Factorization(NMF), which is a dimensionality reduction and features extraction technique. In recent years, it has become so popular because of is its ability to extract easily interpretable factors unlike the others.

In the beginning, we presented the theoretical side of NMF, where we described some applications and review some algorithms. Then, we touched upon how to consider it as a clustering tool through the strong relation between it and the other clustering methods including K-Mean and PLSI. Moreover, we investigated the relationship between NMF and autoencoder as a deep learning technique.

After presenting the theoretical side, we proposed a new method of clustering using ONMF and a simulated annealing called SA-ONMF.

The obtained results were promising, the proposed method demonstrated its performance for clustering on several datasets including BreastCancer dataset, Wine dataset, Iris dataset, Digits dataset, ..., etc, where the effect of our algorithm is directly proportional to the amount of overlap between clusters. The major limitation for SA-ONMF is much longer time consumption than ONMF.

To better understand the implications of these results, future studies should address other variants of NMF instead of ONMF, and they can propose a method to set the size of the chosen subsets, they also need to speed up this algorithm to be more efficient.

Bibliography

[1] Oussama Aiadi and Mohammed Lamine Kherfi. A new method for automatic date fruit classification. *International Journal of Computational Vision and Robotics*, 7(6):692–711, 2017.

[2] Oussama Aiadi, Mohammed Lamine Kherfi, and Belal Khaldi. Automatic date fruit recognition using outlier detection techniques and gaussian mixture models. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, 18(1):52–75, 2019.

[3] Mebarka Allaoui, Mohammed Lamine Kherfi, and Abdelhakim Cheriet. Considerably improving clustering algorithms using umap dimensionality reduction technique: A comparative study. In *International Conference on Image and Signal Processing*, pages 317–325. Springer, 2020.

[4] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[5] Babajide O Ayinde, Ehsan Hosseini-Asl, and Jacek M Zurada. Visualizing and understanding nonnegativity constrained sparse autoencoder in deep learning. In *International Conference on Artificial Intelligence and Soft Computing*, pages 3–14. Springer, 2016.

[6] Dhananjay Bansal, Bhiksha Raj, and Paris Smaragdis. Bandwidth expansion of narrowband speech using non-negative matrix factorization. In *Ninth European Conference on Speech Communication and Technology*, 2005.

[7] Paresh Chandra Barman, Nadeem Iqbal, and Soo-Young Lee. Non-negative matrix factorization based text mining: feature extraction and classification. In *International Conference on Neural Information Processing*, pages 703–712. Springer, 2006.

[8] Michael W Berry and Murray Browne. Email surveillance using non-negative matrix factorization. *Computational & Mathematical Organization Theory*, 11(3):249–264, 2005.

[9] Satwik Bhattamishra. Deep probabilistic nmf using denoising autoencoders. *International Journal of Machine Learning and Computing*, 8(1):49–53, 2018.

[10] Paul S Bradley and Usama M Fayyad. Refining initial points for k-means clustering. In *ICML*, volume 98, pages 91–99. Citeseer, 1998.

[11] Jean-Philippe Brunet, Pablo Tamayo, Todd R Golub, and Jill P Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the national academy of sciences*, 101(12):4164–4169, 2004.

[12] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[13] Bin Cao, Dou Shen, Jian-Tao Sun, Xuanhui Wang, Qiang Yang, and Zheng Chen. Detect and track latent factors with online nonnegative matrix factorization. In *IJCAI*, volume 7, pages 2689–2694, 2007.

[14] Seungjin Choi. Algorithms for orthogonal nonnegative matrix factorization. In *2008 ieee international joint conference on neural networks (ieee world congress on computational intelligence)*, pages 1828–1832. IEEE, 2008.

[15] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[16] Ruairí de Fréin, Konstantinos Drakakis, Scott Rickard, and Andrzej Cichocki. Analysis of financial data using non-negative matrix factorization. In *International Mathematical Forum*, volume 3, pages 1853–1870. Journals of Hikari Ltd, 2008.

[17] Farah Debbagh, Mohammed Lamine Kherfi, and Mohamed Chaouki Babahenini. A semantic relatedness-based solution for reducing missing problem in tbir. *International Journal of Signal and Imaging Systems Engineering*, 10(3):146–156, 2017.

[18] Arden Dertat. Applied deep learning - part 3: Autoencoders. `https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798`. Accessed: 2020-09-22.

[19] Chris Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 606–610. SIAM, 2005.

[20] Chris Ding, Tao Li, and Wei Peng. Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method. In *AAAI*, volume 42, pages 137–43, 2006.

[21] Khaoula Drid, Mebarka Allaoui, and Mohammed Lamine Kherfi. Object detector combination for increasing accuracy and detecting more overlapping objects. In *International Conference on Image and Signal Processing*, pages 290–296. Springer, 2020.

[22] Richard O Duda, Peter E Hart, et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[23] Mohammed Ehsan. Face recognition rate using different classifier methods based on pca - scientific figure on researchgate. `https://www.researchgate.net/figure/The-ORL-database-for-training-and-testing_fig1_318126880`. Accessed: 2020-09-21.

[24] Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830, 2009.

[25] Eric Gaussier and Cyril Goutte. Relation between plsa and nmf and implications. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 601–602, 2005.

[26] David Guillamet and Jordi Vitria. Non-negative matrix factorization for face recognition. In *Catalonian Conference on Artificial Intelligence*, pages 336–344. Springer, 2002.

[27] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[28] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

[29] Ehsan Hosseini-Asl, Jacek M Zurada, and Olfa Nasraoui. Deep learning of part-based representation of data using sparse autoencoders with nonnegativ-

ity constraints. *IEEE transactions on neural networks and learning systems*, 27(12):2486–2498, 2015.

[30] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(Nov):1457–1469, 2004.

[31] Andreas Janecek and Ying Tan. Iterative improvement of the multiplicative update nmf algorithm using nature-inspired optimization. In *2011 Seventh International Conference on Natural Computation*, volume 3, pages 1668–1672. IEEE, 2011.

[32] Andreas Janecek and Ying Tan. Using population based algorithms for initializing nonnegative matrix factorization. In *International Conference in Swarm Intelligence*, pages 307–316. Springer, 2011.

[33] Leonard Kaufmann and Peter J Rousseeuw. Finding groups in data: an introduction to cluster analysis. *New York: Jonh Wiley*, 1990.

[34] Belal Khaldi, Oussama Aiadi, and Mohammed Lamine Kherfi. Combining colour and grey-level co-occurrence matrix features: a comparative study. *IET Image Processing*, 13(9):1401–1410, 2019.

[35] Belal Khaldi, Oussama Aiadi, and Kherfi Mohammed Lamine. Image representation using complete multi-texton histogram. *Multimedia Tools and Applications*, pages 1–19, 2020.

[36] Belal Khaldi and Mohammed Lamine Kherfi. Modified integrative color intensity co-occurrence matrix for texture image representation. *Journal of Electronic Imaging*, 25(5):053007, 2016.

[37] Mohammed Lamine Kherfi. Review of human-computer interaction issues in image retrieval. *Advances in Human-Computer Interaction, I-Tech Education and Publishing KG, Vienna, Austria*, pages 215–240, 2008.

[38] Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 2007.

[39] Jingu Kim, Yunlong He, and Haesun Park. Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. *Journal of Global Optimization*, 58(2):285–319, 2014.

[40] Jingu Kim and Haesun Park. Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM Journal on Scientific Computing*, 33(6):3261–3281, 2011.

[41] Meriem Korichi, Mohamed Lamine Kherfi, Mohamed Batouche, and Khadra Bouanane. Extended bayesian generalization model for understanding user's intention in semantics based images retrieval. *Multimedia Tools and Applications*, 77(23):31115–31138, 2018.

[42] Amy N Langville, Carl D Meyer, Russell Albright, James Cox, and David Duling. Initializations for the nonnegative matrix factorization. In *Proceedings of the twelfth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 23–26. Citeseer, 2006.

[43] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[44] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[45] Andre Lemme, René Felix Reinhart, and Jochen Jakob Steil. Online learning and generalization of parts-based image representations by non-negative sparse autoencoders. *Neural Networks*, 33:194–203, 2012.

[46] Stan Z Li, Xin Wen Hou, Hong Jiang Zhang, and Qian Sheng Cheng. Learning spatially localized, parts-based representation. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.

[47] Tao Li and Chris HQ Ding. Nonnegative matrix factorizations for clustering: A survey., 2013.

[48] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[49] Weixiang Liu, Nanning Zheng, and Qubo You. Nonnegative matrix factorization and its applications in pattern recognition. *Chinese Science Bulletin*, 51(1):7–18, 2006.

[50] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[51] Oded Maimon and Lior Rokach. Data mining and knowledge discovery handbook. 2005.

[52] Glenn W Milligan. A monte carlo study of thirty internal criterion measures for cluster analysis. *Psychometrika*, 46(2):187–199, 1981.

[53] Richard Mojena. Hierarchical grouping methods and stopping rules: an evaluation. *The Computer Journal*, 20(4):359–363, 1977.

[54] Dominic Monn. enoising autoencoders explained. `https://towardsdatascience.com/denoising-autoencoders-explained-dbb82467fc2`. Accessed: 2020-09-22.

[55] Azad Naik. kernel k-means clustering algorithm. `https://sites.google.com/site/dataclusteringalgorithms/kernel-k-means-clustering-algorithm`. Accessed: 2020-02-12.

[56] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[57] Finn Årup Nielsen. Clustering of scientific citations in wikipedia. *arXiv preprint arXiv:0805.1154*, 2008.

[58] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, June 1994.

[59] V Paul Pauca, Farial Shahnaz, Michael W Berry, and Robert J Plemmons. Text mining using non-negative matrix factorizations. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 452–456. SIAM, 2004.

[60] Surya Priy. Clustering in machine learning. `https://www.geeksforgeeks.org/clustering-in-machine-learning/#:~:text=Clustering%20is%20the%20task%20of,data%20points%20in%20other%20groups.` Accessed: 2020-09-21.

[61] Chandan K Reddy and Bhanukiran Vinzamuri. A survey of partitional and hierarchical clustering algorithms. In *Data Clustering*, pages 87–110. Chapman and Hall/CRC, 2018.

[62] Abdou Rockikz. How to use k-means clustering for image segmentation using opencv in python. `https://www.thepythoncode.com/article/kmeans-for-image-segmentation-opencv-python`. Accessed: 2019-12-28.

[63] R Roseline, G Jenitha, and Henri Amirhtaraj. Analysis and application of clustering techniques in data mining. *International Journal of Computing Algorithm*, 3:910–912, 2014.

[64] Farial Shahnaz, Michael W Berry, V Paul Pauca, and Robert J Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386, 2006.

[65] Cori Sheperis. Using cluster analysis for market research. `https://aytm.com/blog/using-cluster-analysis-for-market-research/`. Accessed: 2020-05-03.

[66] Paris Smaragdis. Convolutive speech bases and their application to supervised speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):1–12, 2006.

[67] Steven Squires, Adam Prügel Bennett, and Mahesan Niranjan. A variational autoencoder for probabilistic non-negative matrix factorisation. *arXiv preprint arXiv:1906.05912*, 2019.

[68] Charlotte Van Steen. Localisation and characterisation of corrosion damage in reinforced concrete by means of acoustic emission and x-ray computed tomography - scientific figure on researchgate. `https://www.researchgate.net/figure/Different-linkage-methods-for-hierarchical-clustering_fig5_329208978`. Accessed: 2020-09-21.

[69] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[70] Ali Caner Türkmen. A review of nonnegative matrix factorization methods for clustering. *arXiv preprint arXiv:1507.03194*, 2015.

[71] Unknown. dendrogram. `https://www.mathworks.com/help/stats/dendrogram.html`. Accessed: 2020-09-21.

[72] Unknown. Orl images. `http://nimfa.biolab.si/nimfa.examples.orl_images.html`. Accessed: 2020-09-21.

[73] unknown. Orl images. `http://nimfa.biolab.si/nimfa.examples.orl_images.html`. Accessed: 2020-05-09.

[74] Sowmya Vivek. Clustering algorithms for customer segmentation. `https://towardsdatascience.com/clustering-algorithms-for-customer-segmentation-af637c6830ac`. Accessed: 2020-09-21.

[75] Rui Xu and Don Wunsch. *Clustering*, volume 10. John Wiley & Sons, 2008.

[76] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273, 2003.

[77] Ka Yee Yeung, Chris Fraley, Alejandro Murua, Adrian E. Raftery, and Walter L. Ruzzo. Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10):977–987, 2001.

[78] Rafal Zdunek and Andrzej Cichocki. Nonnegative matrix factorization with quadratic programming. *Neurocomputing*, 71(10-12):2309–2320, 2008.

[79] Syoya Zhou. What happens in sparse autoencoder. `https://medium.com/@syoya/what-happens-in-sparse-autencoder-b9a5a69da5c6`. Accessed: 2020-08-11.