12-2018

STUDY ON THE PATTERN RECOGNITION ENHANCEMENT FOR MATRIX FACTORIZATIONS WITH AUTOMATIC RELEVANCE DETERMINATION

hau tao

STUDY ON THE PATTERN RECOGNITION ENHANCEMENT FOR MATRIX FACTORIZATIONS WITH AUTOMATIC RELEVANCE DETERMINATION

A Thesis

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Hau Quang Tao

December 2018

STUDY ON THE PATTERN RECOGNITION ENHANCEMENT FOR MATRIX FACTORIZATIONS WITH AUTOMATIC RELEVANCE DETERMINATION

A Thesis

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Hau Quang Tao

December 2018

Approved by:

Dr. Qingquan Sun, Advisor, Computer Science and Engineering

Dr. Kerstin Voigt, Committee Member

Dr. Yunfei Hou, Committee Member

ABSTRACT

Learning the parts of objects have drawn more attentions in computer science recently, and they have been playing the important role in computer applications such as object recognition, self-driving cars, and image processing, etc… However, the existing research such as traditional non-negative matrix factorization (NMF), principal component analysis (PCA), and vector quantitation (VQ) has not been discovering the ground-truth bases which are basic components representing objects. On this thesis, I am proposed to study on pattern recognition enhancement combined non-negative matrix factorization (NMF) with automatic relevance determination (ARD). The main point of this research is to propose a new technique combining the algorithm Expectation Maximization (EM) with Automatic Relevance Determination (ARD) to discover the ground truth basis of datasets, and then to compare my new proposed technique to the others such as: traditional NMF, sparseness constraint and graph embedding in pattern recognition problems to verify if my method has over performance in accuracy rate than the others. Particularly, the new technique will be tested on variety of datasets from simple to complex one, from synthetic datasets to real ones. To compare the performance, I split these datasets into 10 random partitions as the training and the testing sets called 10-fold cross validation, and then use the technique called Euclidean algorithm to classify them and test their accuracy. As the result, my proposed method has higher accuracy

than the others, and it is good to use in pattern recognition problems with missing data.

ACKNOWLEDGEMENTS

I would like to thank my family for supporting me to finish this thesis. I would like to thank to Dr. Qingquan Sun, my advisor, who contributes ideas, corrections, suggestion on this thesis as well as the committee members: Dr Kerstein Voigt and Dr. Yunfei Hou for my success.

DEDICATION

Thanks all my family members: my parents, my aunt, and my uncle, my sisters, and my younger brother for supporting the time and being patient on me to complete the thesis

TABLE OF CONTENTS

ABSTRACT iii

ACKNOWLEDGEMENTS v

LIST OF TABLES ix

LIST OF FIGURES x

CHAPTER ONE: INTRODUCTION

Pattern Recognition 1

Motivation 2

Signification Implications 3

CHAPTER TWO: AUTOMATIC RELEVANCE DETERMINATION WITH NMF

Introduction 6

NMF Concepts and Properties 7

NMF Model 7

Cost Function 8

Multiplicative Update Rules 9

Applications 9

Automatic Relevance Determination 11

Model Order Determination 12

Mathematical Model of ARD 12

CHAPTER THREE: INFORMATIVE MODEL FOR ARD USING EXPECTATION MAXIMIZATION (EM)

Motivation 18

Related Work 19

Data Models [27] 20

Gaussian Distribution 21
Poisson Distribution 21
Parameter Models [27] 22
Half Normal Distribution 22
Exponential Distribution 23
Tweedie Distribution 25
Non-Informative Model for ARD Using EM Algorithm 26
EM Mathematical Model [27] 27
Prior Assumption 27
Gaussian Log-likelihood 28
EM Algorithm Implementation 28
CHAPTER FOUR: NMF BASED ON SPARSENESS CONSTRAINTS
Sparse Coding 32
Sparseness Constraints Concepts 34
NMF with Sparseness Constraint 37
Sparseness Constraints Mathematical Model 37
CHAPTER FIVE: NMF BASED ON GRAPH EMBEDDING
Introduction 39
Graph Creation 39
Graph Embedding 41
Graph Embedding Mathemcatical Model 41
CHAPTER SIX: EXPERIMENTS AND EVALUATIONS
Datasets 43
The Fence Dataset 43

The Swimmer Dataset 44

The ORL Faces Dataset 45

The JAFFE Faces Dataset 46

The Extended Yale Faces Dataset 47

Ground Truth Bases Discovery 48

Recognition Accuracy Comparison to Unsupervised , and Supervised Learning 78

Recognition Accuracy Comparison to Sparsity Based , and Graph Embedding 84

CHAPTER SEVEN: CONCLUSION AND FUTURE WORKS

Conclusion 89

Future Works 89

REFERENCES 91

LIST OF TABLES

Table 1. Ground Truth Bases Discovery of Fence, Swimmer, ORL, Jaffe 72

Table 2. Recognition Accuracy Rate for Fence (K=16); Swimmer(K=35), ORL Faces (K=100), and Jaffe (K=100) via EM; NMF, PCA, and LDA 78

Table 3. Comparison EM to Sparsity Based, and Graph Embedding 84

LIST OF FIGURES

Figure 1. The Example of Hand Written Digit Recognition [1] 2

Figure 2. The NMF Decomposition Model 8

Figure 3. Image Reconstruction of MIT Dataset for the 19x19 Pixel Image 10

Figure 4. Basic Images: Eyes, Nose, Eyebrows, Mouth for Face 11

Figure 5. The Graphical Model for NMF where M is Number of Rows on W 14

Figure 6. The Algorithm for ARD by Multiplicative Updating Rules 16

Figure 7. The Ground Truth Bases Discovery of a Swimmer Dataset 17

Figure 8. The Sparse Coding Diagram 33

Figure 9. The Sparse Coding Network for the Image [14] 34

Figure 10. Different Sparseness Constraints Level on Vectors [5] 35

Figure 11. Changes of Sparseness Constraints Level on ORL Faces [5] 36

Figure 12. NMF with Sparseness Constraints 37

Figure 13. Adjacency Relationship Intrinsic and Penalty Graph [31] 40

Figure 14. The Sample of Fence Dataset [7] 44

Figure 15. Sample of Swimmer Dataset [8] 45

Figure 16. The Sample Images of ORL Dataset [21] 46

Figure 17. The Sample of Japanese Faces Dataset [34] 47

Figure 18. Sample Faces Images of Extended Yale Dataset [36] 47

Figure 19. Ground Truth Bases Discovery via EM Based ARD for Fence 49

Figure 20. Pattern Discovery via NMF for Fence 49

Figure 21.Pattern Discovery via PCA for Fence 50

Figure 22. Pattern Discovery via NMF with Sparseness Constraint for Fence ... 50
Figure 23. Pattern Discovery via NMF with Graph Embedding for Fence 50
Figure 24. Ground Truth Bases Discovery with EM Based ARD of Swimmer 52
Figure 25. Basic Images Discovery with NMF for Swimmer 53
Figure 26. Basic Images Discovery via NMF with GE for Swimmer 54
Figure 27. Basic Images Discovery via PCA for Swimmer 55
Figure 28. Basic Images Discovery via NMF with SC for Swimmer..... 56
Figure 29. Ground Truth Bases Discovery via EM Based ARD for ORL 57
Figure 30. Basic Images Discovery via NMF for ORL 58
Figure 31. Basic Images Discovery via NMF with GE for ORL 59
Figure 32. Basic Images Discovery via PCA for ORL..... 60
Figure 33. Basic Images Discovery via NMF with SC for ORL 61
Figure 34. Ground Truth Bases Discovery via EM Based ARD for Jaffe 62
Figure 35. Basic Images Discovery via NMF for Jaffe 63
Figure 36. Basic Images Discovery via NMF with GE for Jaffe..... 64
Figure 37. Basic Images Discovery via PCA for Jaffe 65
Figure 38. Basic Images Discovery via NMF with SC for Jaffe..... 66
Figure 39. Ground Truth Bases Discovery via EM for Extended Yale 67
Figure 40. Basic Images Discovery via NMF for Extended Yale..... 68
Figure 41. Basic Images Discovery via PCA for Extended Yale 69
Figure 42. Basic Images Discovery via NMF with SC for Extended Yale 70
Figure 43. Basic Images Discovery via NMF with GE for Extended Yale 71

Figure 44. L_2 Norm is to Discover 8 Ground Truth Bases for Fence Dataset 73

Figure 45. L_2 Norm Discovers 16 Ground Truth Bases of Swimmer Dataset 74

Figure 46. L_2 Norm is to Discover 62 Ground Truth Bases for ORL 75

Figure 47. L_2 Norm is to Discover 64 Ground Truth Bases for Jaffe Dataset 76

Figure 48. L_2 Norm is to Discover 73 Ground Truth Bases for Yale Dataset 77

Figure 49. Recognition Accuracy Comparison of EM to Others for Swimmer 80

Figure 50. Recognition Accuracy Comparison of EM to Others for ORL 81

Figure 51. Recognition Accuracy Comparison of EM to Others for Jaffe..... 82

Figure 52. Recognition Accuracy Comparison of EM to Others for Yale 83

Figure 53. Comparison of EM to Others for Swimmer Dataset..... 85

Figure 54.Comparison of EM to Others for ORL Dataset 86

Figure 55. Comparison of EM to Others for Jaffe Dataset..... 87

Figure 56. Comparison of EM to Others for Yale Dataset 88

CHAPTER ONE

INTRODUCTION

Pattern Recognition

Patten recognition and machine learning have played an important role in many modern computer applications recently such as: computer vision, image segmentation, natural language processing,1 visualization, data mining, etc… and many researchers have been discovering variety of algorithms to achieve better accuracy rate on object recognition problems. Pattern recognition is the subject which automatically discovers regularities in data by using algorithms combining with using these regularities to solve some interesting problems such as classifying objects into different categories [1]. To illustrate what the pattern recognition is, we will consider the simple, famous handwritten digit recognition problem on figure 1.1. Each digit (0,1,…,9) will have 28 x28 pixel image, so we can create a vector x consisting of 784 real numbers. Our purpose will build the adaptive model to identify correctly the digit as the output from input data x. The accuracy or performance also depends on the models, algorithms, classification methods we choose.

For more detail, we are using machine learning approach to solve recognition problems. From a large batch of input x consisting different digits is considered as training set which is used to find out the parameters for the learning models. Each digit has been put into correct catalogues beforehand.

The output running by machine learning algorithm is a function y(x) that means the model takes input x and then generate the output y(x). If the output digit y(x) is matching to the input x classified in categories, the model produces the correct identified digit. Otherwise, it is mismatching. The accuracy of the model is calculated based on the proportion of number of matching digits. When the model is trained, then it can be used to classify new digit images which are not in the training set called test set. The ability that machine can recognize the new digit image not being in the training set plays the important role in practical applications called generalization [1]

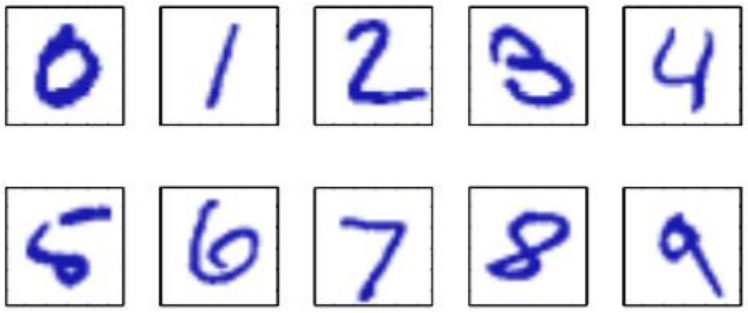

Figure 1. The Example of Hand Written Digit Recognition [1]

Motivation

Learning parts of objects is important in computer application, and it gets more attention from many researchers. However, these famous algorithms using in machine learning recently cannot recognize the part of objects called the

ground truth bases such as conventional non negative matrix factorization (NMF)[2] [3], principal component analysis (PCA)[4], sparseness constraint[5], graph embedding[6]. For example, in face recognition application, it is supposed that a human face is composed of four basic components: mouth, nose, eyes, and eyebrows that are ground-truth bases to represent a face. If an algorithm could discover correctly four above components, it can represent a face. In contrast, if an algorithm extracts components rather than four, it means that a face is composed by other parts that are not intrinsic features [7]. Indeed, PCA , sparseness constraint, and graph embedding only discovered a whole face instead of ground-truth bases while traditional NMF discovered basic components that are redundant. In practice, an algorithm fails to extract basic components leading to not recognize correctly objects, not detect motions in video, and camera processing. If it is applied in real time applications: self-driving car, face recognition, it will cause serious issues related to security and safety. Therefore, finding correctly the number of ground truth bases is significant in extracting the hidden structures of investigated data, and improving a performance.

Signification Implications

In practice, data-sets are so complicated and redundant, and we also have to deal with missing data. NMF has become the popular technique for data analysis and dimensionality reduction. However, we have to assume the

number of components and choose the appropriate values depending on specific datasets. Therefore, it causes heavily cost of computation and time consuming of doing experiments to choose the best value for the number of components [8] [9]. Tan and Févotte [10] proposed the method to automatically determine the optimal value of the number of ground truth bases. It has the advantage of fewer computations involved, but the drawback is that we have to estimate parameters in mathematical models depending on datasets. Indeed, for complicated datasets, this method is not practical because we have to choose the best values of parameters with many trials to discover the correct number of ground truth bases. In this proposal, we propose another technique that integrates Expectation Maximization (EM) algorithm [11] to determine the optimal value of the number of components. Using EM to resolve missing, hidden problems is very powerful, and efficient, and this solution is suitable to our issue. Our advantage is that it is free hyper-parameters in mathematical models, but still gets the expected result. In the audio-visual scene analysis, a speaker may face to the camera while he/she keeps silent, or a speaker turns away of a camera while he/she is speaking. Speech signals have the sparse structure and have the mixture of different sources such as voice, noise, …, music background. Applying the EM on this scene is well-suited to find audio-visual clusters, and to discriminate between speaking and silent people [12], and then we will compare our results with different algorithms: sparseness

constraint [13] [14] [15] and non-negative graph embedding [16]–[18] to see if EM algorithm has better performance compared to others.

CHAPTER TWO

AUTOMATIC RELEVANCE DETERMINATION WITH NMF

Introduction

Non-negative matrix factorization (NMF) was first introduced by Lee, Seung [2] [19] as the machine learning technique learning a parts representative of data, and then it has been used widely in dimensionality reduction, and extract the sparse and useful features from datasets[20]. For example, NMF can be applied to the face recognition to discover some basic components as known as learned base images such as: eyes, eyebrows, mouth, nose, and cheek, etc…[2] which have locally representation than comprehensively. Objects in universe are represented by non-negative physical values such as: pixels, weight, length… , so NMF is the suitable machine learning technique learning part of objects. Although principle component analysis (PCA), linear discriminant analysis (LDA) have been famous techniques in dimensionality reduction, both of them are consisting of negative and positive values which is not represented correctly the physical meaning of objects in the world. In addition, NMF decomposes the original matrix into sub matrices containing only non-negative values having meaningful representation of objects. Furthermore, NMF has the simple multiplicative iteration which has more advantages than others [21]. In this chapter, we will discuss the basic concepts of NMF such as mathematical model, cost functions, the multiplicative update rule, and some applications.

NMF Concepts and Properties

NMF Model

Supposedly, we have the M dimensional of the random non-negative vector x, and N is the number of observations denoted as x_i (i=1,2,3,...,N), Supposedly, we have the big matrix X having M-by-N dimensionality in which M is the number of rows (M dimensionality of a random vector), and N is the number of columns (the number of observation), denoted as $X \in R^{M \times N}$, NMF will separate the original matrix X in to 2 smaller sub matrices W and H in which W has M-by-K dimensionality ($W \in R^{M \times K}$, namely basic matrix consisting basic components extracted from the original data, and H is the coefficient matrix having K-by-N dimensionality ($H \in R^{K \times N}$), The combination of W and H is used to reconstruct the whole objects. In the mathematical form, we need to decompose X into W and H such that W, and H need to satisfy equation below

$$X \approx WH \; s.t. W \geq 0, H \geq 0 \; (1)$$

Where K, the unknown parameter, is the latent number as well as the number of columns on W and rows on H respectively. Normally, K usually is chosen such that $\mathrm{K} \leq \mathrm{M} \times \mathrm{N}/(\mathrm{M} + \mathrm{N})$ [7]. Figure 2 is to illustrate the NMF decomposition model

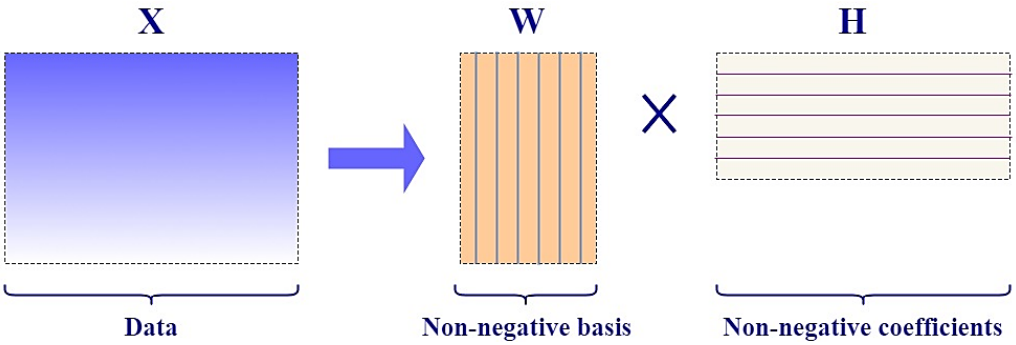

Figure 2. The NMF Decomposition Model

Cost Function

It is used to estimate the factorization X ≈ WH, and measure how performance of the approximate factorization is. Basically, const function can be calculated by distance between two non-negative matrices namely, C and D. Usually the cost function has been obtained by the simple measurement called Euclidian distance between C and D. Euclidean distance has the form below:$\|C - D\|^2 = \sum_{ij}(C_{ij} - D_{ij})^2$. In NMF, the cost function is considered as the optimization problems which minimize $\|V - WH\|^2$ with respect to W and H such that W, and H ≥ 0. Although the function above is convex either on W and H, but not both of them. Therefore, it is hard to find the global minima of this function, but finding the local minima is possible. One of the easiest way to find the local minima is to use the gradient descent. However, the convergence speed is going to be slow [22]

Multiplicative Update Rules

The alternative way to find the local minima in cost function from optimization problem above, we can use the multiplicative rules which is easy to implement and faster in convergence than gradient descent. The two equations below are the updating rules implemented on W and H at the same time using Euclidean distance:

$$H_{kj} \leftarrow H_{ij} \frac{(W^T X)_{kj}}{(W^T WH)_{kj}} \quad (2)$$

$$W_{ik} \leftarrow W_{ik} \frac{(XH^T)_{ik}}{(WHH^T)_{ik}} \quad (3)$$

Applications

NMF has been used widely in many applications by its property such as: automatically extracting hidden and sparse components from data. In this section, we will discuss more detail of NMF applications in imaging processing-facial reconstruction, and text mining-topic recovery and documentation classification. Furthermore, NMF also has applications in environment[23], biology [24], In the image processing, let assume each columns of face data matrix $X \in R^{M \times N}$ be a vector of M dimensionality in the greyscale, and N are the number of faces of each person. The entry (i, j)th means the ith pixel of the face jth . NMF can decompose the original matrix X into non-negative sub matrices W, and H such that each column of matrix W can represent images, and we usually denote W to be the basis image matrix, and then we can reconstruct the original face images by linear combination of W and coefficient matrix H ($H \geq 0$). Figure

3 is to illustrate how NMF decomposes the original matrix to basis images matrix (W) and coefficient matrix (H). Furthermore, usually, the unknown parameter k (the number of columns, and rows on W and H respectively) is much smaller than N (the number of faces) in a big dataset. Therefore, the basis matrix is decomposed into localized feature, and then with few basis images, we can reconstruct to original faces. Figure 4 is to illustrate some basic localized feature such as: mouth, nose, eyes, mustaches, lips, eyebrows.[20].

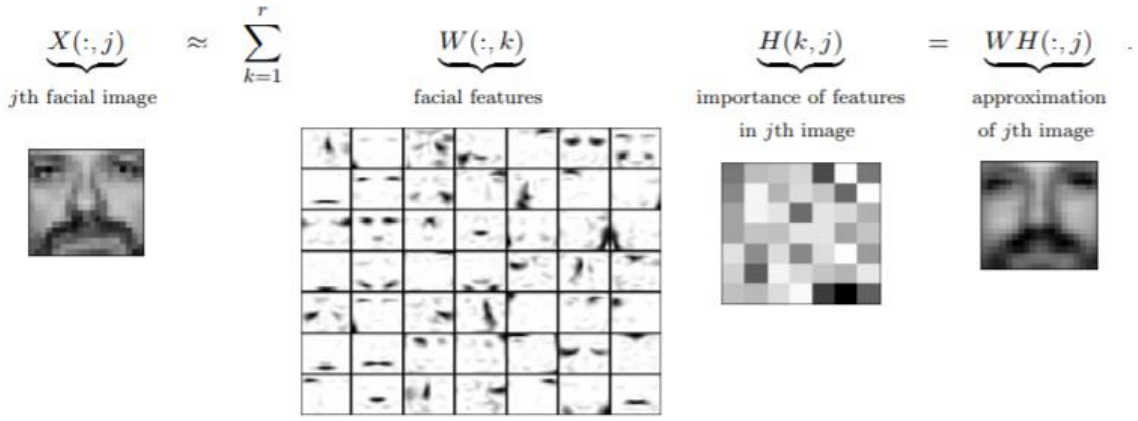

Figure 3. Image Reconstruction of MIT Dataset for the 19x19 Pixel Image

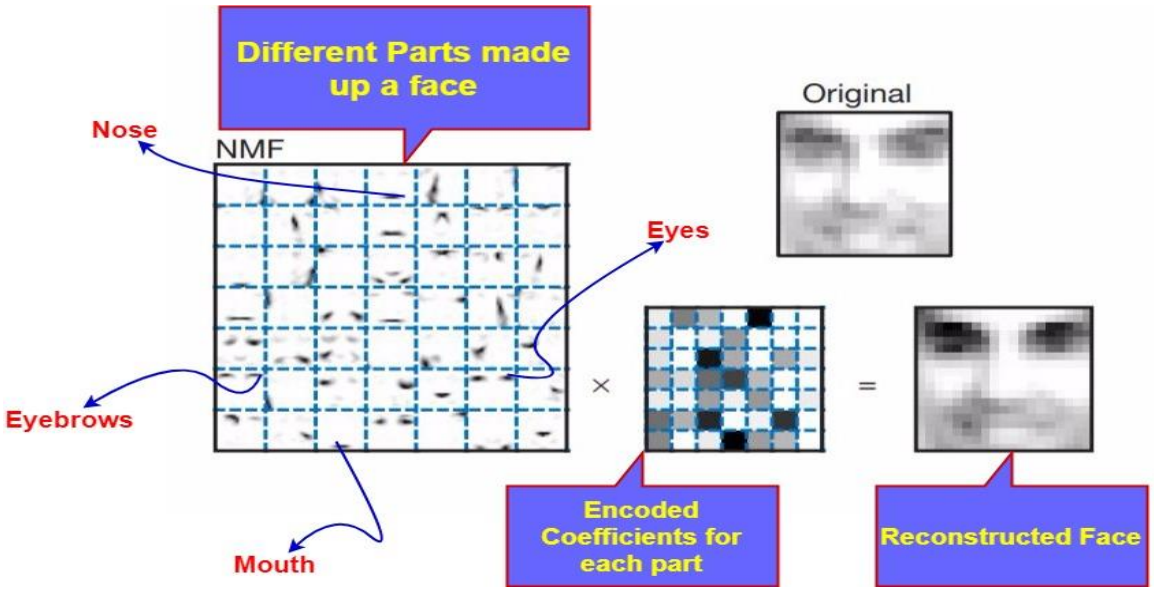

Figure 4. Basic Images: Eyes, Nose, Eyebrows, Mouth for Face

Automatic Relevance Determination

Although NMF has been used widely in machine learning such as face recognition, and text mining, speech analysis, it is assuming a given latent parameter, or the number of components K. Practically, it is not easy to choose the k components because it depends a lots on the types and size of datasets. To address this issue, Tan, and Févotte [10] suggested the improvement on basic NMF model by integrating the Bayesian PCA [1], and sparse Bayesian learning [25] into the model. This technique is called automatic relevance determination (ARD) [26] , and it is successful to estimate the latent number k, and the result has been validated by synthetic datasets.

Model Order Determination

In NMF, the latent K, the model order, is used to discover the ground truth bases from dataset. Therefore, it is can be used to find out the meaningful and hidden information from datasets. However, in practice, K is hard to estimate because we do not have much prior knowledge about it. Recently, many researchers have paid more attention to how to estimate the optimal K_{eff} to get more understanding on datasets we are investigating, and Tan, and Févotte [10] has proposed ARD method to achieve K_{eff}. For example, assuming, we have a single face made up by 4 basic components like: mouth, eyes, nose, and lips. Ideally, if we can decompose the original matrix X into the basis image W , and coefficient matrix H with number of columns and rows on H to be K_{eff}=4, we can easily to reconstruct the face with only 4 ground truth bases. However, we do not know exactly how many basic components we have, so usually we initialize the number of component K to be very large to make sure we do not skip any important components. In this case, the computing cost is very expensive and time consuming.

Mathematical Model of ARD

To estimate the K_{eff}, we need to add one more prior parameter $\beta=[\beta_1, \beta_{2,\ldots,} \beta_k]$ on columns and rows of W and H respectively, and then, we need to find the optimized value of these hyper-parameters β^*, W* and H* by multiplicative updating rules. Accurately, we need to find β^*, W* and H* by optimizing the maximum a posteriori (MAP):

$$\min(\mathrm{W},\mathrm{H},\beta)\, C_{MAP}(W,H,\beta) \triangleq -\log_p(W,H,\beta|\mathrm{X}) (4)$$

Where the posteriori has the form below:

$$-\log_p(W,H,\beta|\mathrm{X}) \triangleq -\log_p(X|W,H) - \log_p(W|\beta) - \log_p(H|\beta) - \log_p(\beta) (5)$$

To maximize the term $\log_p(X|W,H)$, we need to minimize the Kullback-Leibler cost function $D_{KL}(X|W,H) = D_{KL}(X|\widehat{X})$, where:

$$D_{KL}(X|\widehat{X}) \triangleq \sum_{mn} x_{mn} \log \frac{x_{mn}}{\widehat{x_{mn}}} - x_{mn} + \widehat{x_{mn}} (6)$$

Figure 5 is to illustrate the graphical model of NMF in which $w_{mk,}$ or h_{kn} is estimated the hyper-parameter β_k, and on the top level of the model, β_k is estimated by 2 different hyper-parameters a, and b.

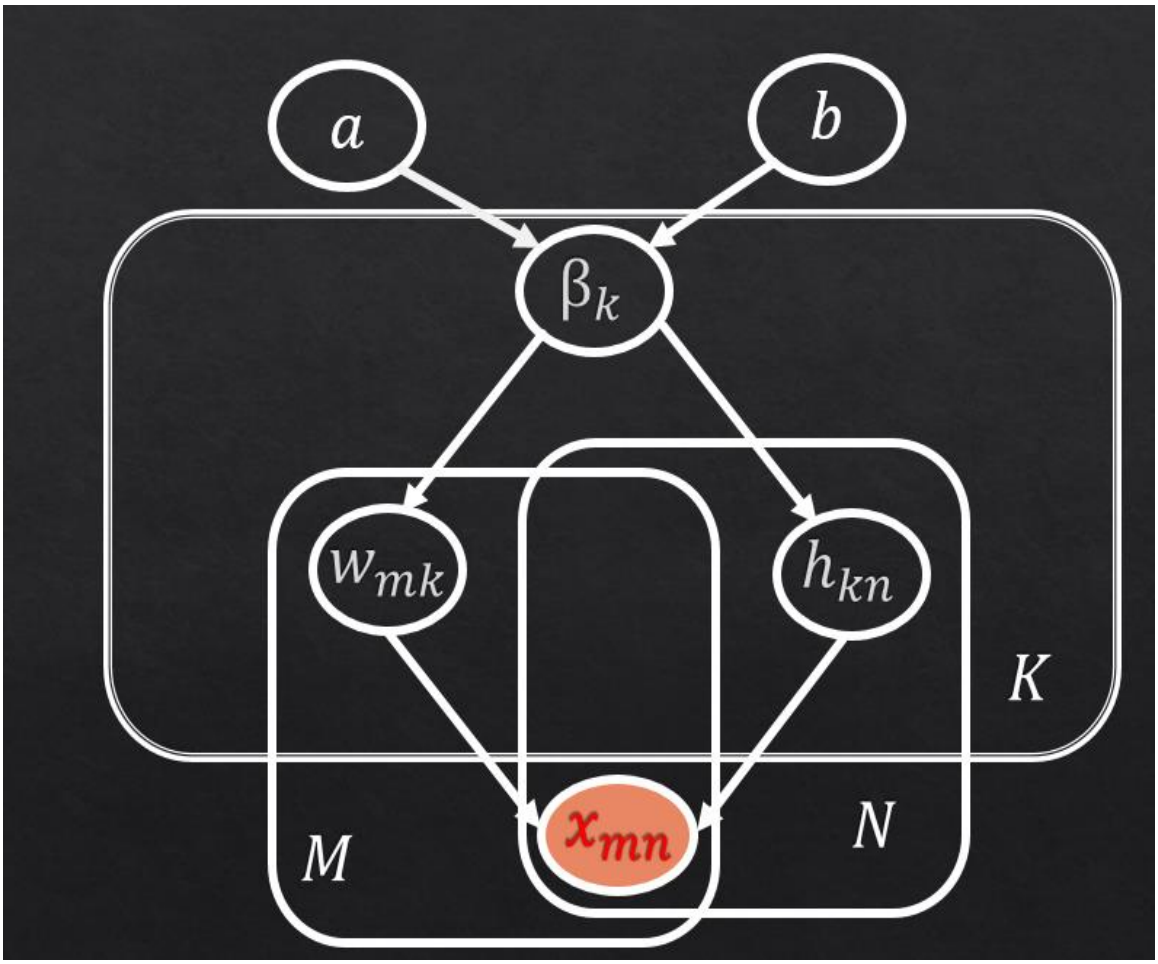

Figure 5. The Graphical Model for NMF where M is Number of Rows on W

Prior model on W and H. We assume the distribution on each column k on W and H is independent half normal distribution, and each prior k is modeled through a parameter β, and it has the form:

$$p(w_{mk}|\beta_k) = \mathcal{HN}\left(w_{mk}\middle|0, {\beta_k}^{-1}\right)(7)$$

$$p(h_{kn}|\beta_k) = \mathcal{HN}\left(h_{kn}\middle|0, {\beta_k}^{-1}\right)(8)$$

Where:

$$\mathcal{HN}(x|0,\beta^{-1}) = \sqrt{\frac{2}{\pi}}\ \beta^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\beta x^2\right) (9)$$

Is the independent half normal distribution for non-negative value of x with inverse variance β^2. From the equation (7) and (8) above, we can obtain the estimation for W and H:

$$-log_p(W|\beta) \sum_k \sum_m \frac{1}{2}\beta_k {w_{mk}}^2 - \frac{M}{2} log\ \beta_k\ (10)$$

$$-log_p(H|\beta) \sum_k \sum_m \frac{1}{2}\beta_k {h_{kn}}^2 - \frac{N}{2} log\ \beta_k\ (11)$$

And then, we define efficient K_{eff}:

$$K_{eff} \triangleq \{\beta_k : \beta_k < L_k - \varepsilon\ \} (12)$$

$$L \triangleq \frac{M + N + 2(a-1)}{b}$$

L_k is the upper bound of β_k, and ε is defined as small value specified by users.

Prior model on β. We are assuming each β_k is distributed as a Gamma distribution with two hyper-parameters a_k, and b_k as known as shape and scale parameters respectively. Therefore, estimation on β has the form:

$$P(\beta_k|a_k, b_k) = \frac{b_k^{a_k}}{\Gamma(a_k)} {\beta_k}^{a_k - 1} \exp(-\beta_k b_k)\,, \beta_k \geq 0\ (13)$$

$$-log_p(\beta) \triangleq \sum_k \beta_k b_k - (a_k - 1)\log(\beta_k)\ (14)$$

The algorithm below shows the step by step how ARD obtains the optimal value of K_{eff} by multiplicative updating rules where V, and F is the original matrix, and the number of rows on W respectively which is equivalent to X matrix, and M

rows mentioned on chapter 2. After the end loop, we can compute the optimal value K_{eff} by equation (12).Figure 7 is to illustrate the ground truth bases discovery via ARD. We know that this dataset has 16 limb positions and one static torso, and ARD can discover 17 basic components. We will discuss more detail about this dataset on later section.

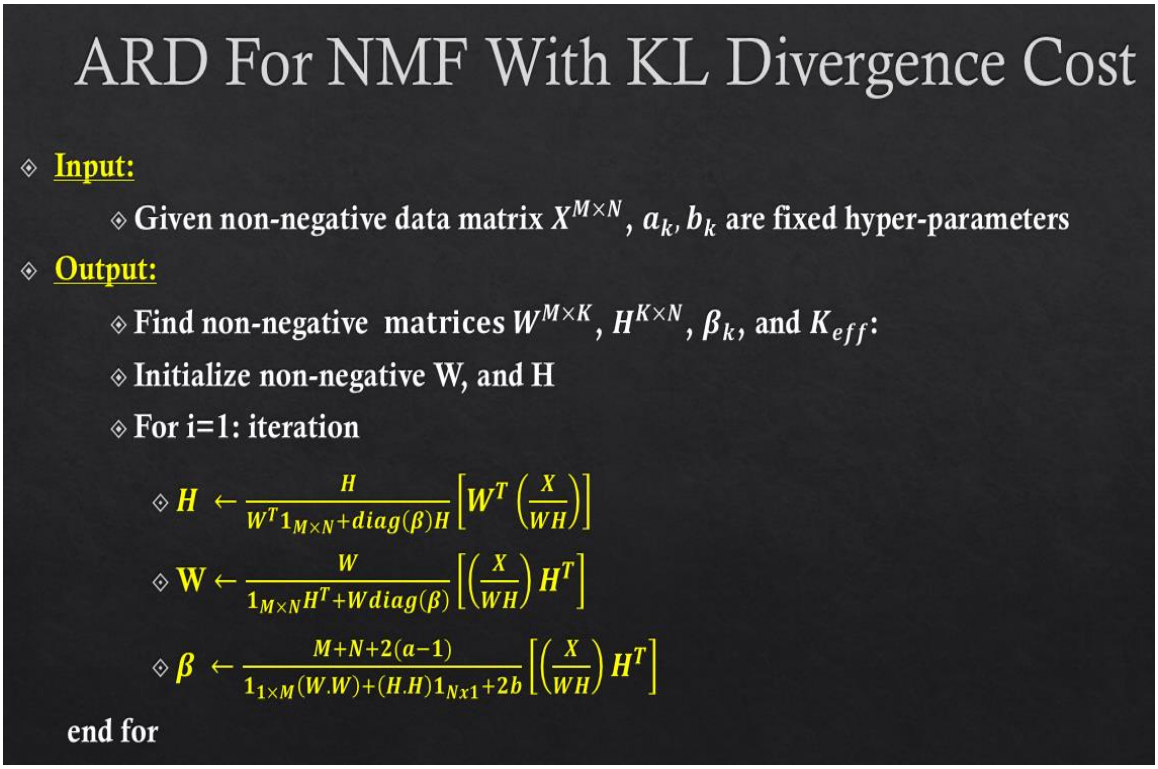

Figure 6. The Algorithm for ARD by Multiplicative Updating Rules

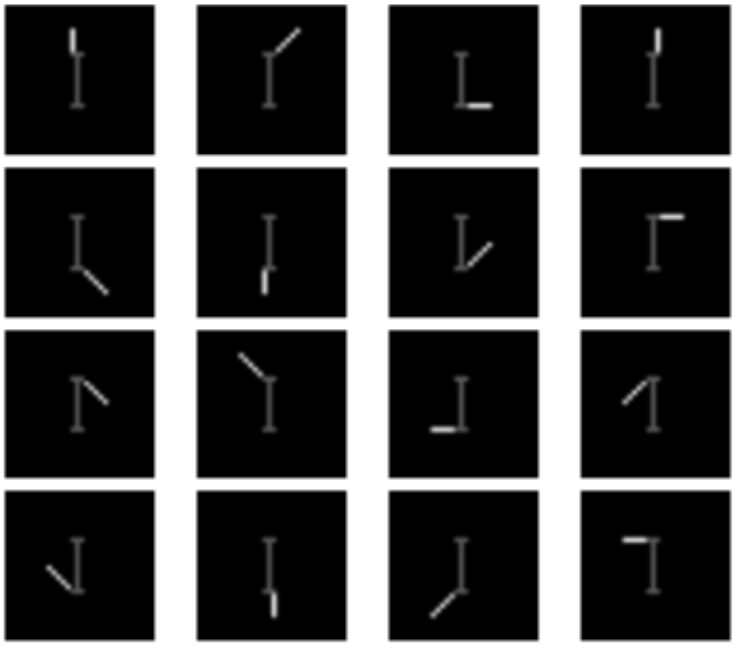

Figure 7. The Ground Truth Bases Discovery of a Swimmer Dataset

CHAPTER THREE

INFORMATIVE MODEL FOR ARD USING EXPECTATION MAXIMIZATION (EM)

Motivation

As mentioned above, we do not have enough prior knowledge to determine the value of K, so we usually choose it randomly, and do many experiments to get the reasonable value of K. This computation is so costly and not practical. Another approach was proposed by Tan and Févotte [10] in which the authors used a technique called automatic relevance determination (ARD) to determine the optimal value of K for the specific data-sets. On this approach, Tan and Févotte assumes the hyper-parameter β_K has the Gamma distribution which depends on 2 other shape and scale parameters denoted a, b respectively. The technique just gets the expected result for some datasets that authors did experiments. For different datasets, we need to adjust hyper-parameters β_K, a, and b to be suitable to new datasets. To avoid involving hyper-parameters to determine the model order and cost function, we propose the expectation maximization (EM) algorithm to determine the model order. Using EM to estimate the model order is well-known technique and suitable to missing or hidden data [11]. We will apply EM to determine the model order of data. Experiments on 5 different data-sets reveal the performance improvement and free hyper-parameters.

Related Work

Researching on selection of model order has not been investigating enough. There are very few literature review discussing model order selection. There are some Bayesian methods to determine model order, but they are not efficient because computation is very costly, and we have to evaluate the corresponded K (model order) based experiments [10]. Tan and Févotte [10] proposed the method to automatically determine the optimal value of K_{eff} given a large initial value of K with less computationally involving. For this proposal, authors try to estimate the number of columns of W as well as the number rows of H to determine the model order and ground-truth basis However, this method has the drawback is that we need to estimate various values of hyper-parameters with variety of data-sets in the model. For complicated data-sets, this method is not practical, and we have to do many experiments to find out the optimal value of K. In addition, authors had two fixed hyper-parameters on the first level parameter structure while another parameter is as a random variable which is not relevant to statistical perspective. Qingquan et al [7] proposed another method called non-informative hierarchical inference in which authors use the hyper-parameter as the random variable rather than a constant to estimate the model order and ground-truth basis. Although, this approach is free hyper-parameters, it is not robust as ARD and Variational Bayesian approach and sensitive to initialization and complexity of the datasets [6] In this thesis, we propose another technique that integrates Expectation Maximization (EM) algorithm to determine

optimal value of model order. Using EM to resolve missing/hidden problems is very powerful, and efficient, and this solution is suitable to our issue. Decomposing the data matrix into W and H without any prior knowledge about the number of columns and rows K_{eff} of W and H respectively. Utilizing the EM algorithm helps us to optimize the value of K to determine correct model order and ground-truth basic. Our advantage is that it is the free hyper-parameters model, but still get the expected result.

Data Models [27]

There are some algorithms developed to determine the model factor K. Some of them are considered as maximum likelihood NMF under the assumption of data distribution. The maximum likelihood (ML) estimate of W and H given by minimizing the negative log likelihood of them [10]

$$\mathrm{ML}(\mathrm{W},\mathrm{H}) = \underset{\mathrm{W},\mathrm{H}\geq 0}{\operatorname{argmin}}\, \mathcal{L}(\mathrm{W},\mathrm{H})(15),$$

where $\mathcal{L}$ (W, H) is the negative log likelihood of the factors. In this section, we will present three common distributions for data layer modeling in NMF optimization: Gaussian distribution, Poisson distribution, and Gamma distribution.

Gaussian Distribution

Assuming the noise in data is following the independent and identical distributed (i.i.d) Gaussian with σ_N^2. We are easy to obtain the Gaussian log-likelihood of W and H

$$p(X|WH,\sigma^2) = \left(\frac{1}{\sqrt{2\sigma^2\pi}}\right)^{M\times N} \prod_M \prod_N \exp\left(\frac{-1}{2}\left(\frac{X-WH}{\sigma}\right)^2\right) (16).$$

The log likelihood function of Eq. (2.3) is obtained accordingly. In fact, the log-likelihood function plays the role as the costing function

$$\log p(X|WH) \propto \frac{1}{2\sigma^2}\sum_M \sum_N (X-WH)^2 (17).$$

Therefore, the ML of W and H could be obtained by taking the gradient of (2.4) [10]

$$\nabla_H \log p(X|WH) = \frac{1}{\sigma_N^2} W^T(WH-X)(18),$$

$$\nabla_W \log p(X|WH) = \frac{1}{\sigma_N^2} (WH-X)H^T(19).$$

Poisson Distribution

If the data is following Poisson distribution that has only one parameter, the entire model is simpler. Furthermore, its cost function will be Kullback-Leibler divergence (KL divergence) that is widely used in NMF optimization [7].

Let θ = [WH] and X denote the parameter of Poisson distribution and random variable, respectively. We can obtain the Poisson probability density function (pdf) with logarithm

$$\mathcal{L}(\theta) = \ln p\,(X|WH) = \ln \prod_i \prod_j \frac{[WH]_{ij}^{X_{ij}} e^{-[WH]_{ij}}}{X_{ij}!}$$

$$= \sum_i \sum_j (X_{ij} \ln [WH]_{ij} - [WH]_{ij} - \ln(X_{ij}!)) = -D_{KL}(X|WH) \quad (20)$$

Based on Stirling's formula [11], the fractal term ln(Xij!) can be simplified and approximated as

$$\ln(X_{ij}!) \approx X_{ij} \ln X_{ij} - X_{ij}. \quad (21)$$

Substituting (2.8) into (2.7), we have

$$L(\theta) = \ln p(X|WH) = \sum_i \sum_j (X_{ij} \ln \frac{[WH]_{ij}}{X_{ij}} - [WH]_{ij} + X_{ij}) = D_{KL}\,(X|W\,H). \quad (22)$$

Obviously, the generalized KL-divergence cold be used as the cost function of the model.

Parameter Models [27]

Half Normal Distribution

In Bayesian PCS [13], each column k of W (respectively row k of H) is given a normal prior with precision parameter β_k. Similarly, independent half-normal priors over each column k of W and row k of H are defined by [12], and the priors are tied together through a single, common precision parameter β_k. We set:

$$p(w_{fk}|\beta_k) = \mathcal{HN}\left(w_{fk}\middle|0, \beta_k^{-1}\right) (23),$$

$$p(h_{kn}|\beta_k) = \mathcal{HN}(w_{kn}|0, \beta_k^{-1}) \text{ (24)}$$

$$\mathcal{HN}(x|0, \beta^{-1}) = \sqrt{\frac{2}{\pi}}\beta^{-\frac{1}{2}}\exp\left(\frac{-1}{2}\beta x^2\right) (25).$$

Eq. (2.14) is the half-normal probability density function (defined for x ≥ 0) parameterized by the precision (inverse variance) β^2.

The minus log-priors can be written as:

$$-\log p(W|\beta) = \sum_k \sum_f \frac{1}{2}\beta_k w_{fk}^2 - \frac{F}{2}\log\beta_k \text{ (26)},$$

$$-\log p(H|\beta) = \sum_k \sum_n \frac{1}{2}\beta_k h_{kn}^2 - \frac{N}{2}\log\beta_k \text{(27)}$$

In practice, it is found that the effective dimensionality can be deduced from the distribution of the β_k' s, and cluster into 2 group: a group of values in same order of magnitude to relevant components and a group of similar values of much higher magnitude corresponding to irrelevant components [12]. They defined effective K as

$$K_{eff} = |\{\beta_k : \beta_k < L_k - \varepsilon\}| \text{ (28)}$$

where L_k is the upper bound dependent on the prior's parameters and $\varepsilon \geq 0$ is a user-defined small constant. The goal is to compute precisely the value of L_k in terms F, N and the parameter of the prior on β_k.

Exponential Distribution

In order to enable our model to be automatic and feasible, we assume that base matrix W and feature matrix H are independent, and we choose to use the

same parameter to model both the columns of basis matrix and the rows of feature matrix. We define an independent exponential distribution for each columns of W and each row of H with prior λ_k to simplify the complexity of the model. The reason to choose exponential model is that it has the sharper performance and free of second parameter. From our assumption, the likelihood of columns of W and rows of H can be represented by [7]

$$p(W_{mk}|\lambda_k) = \lambda_k \cdot e^{-\lambda_k W_{mk}} (29)$$

$$p(H_{kn}|\lambda_k) = \lambda_k \cdot e^{-\lambda_k H_{kn}} (30).$$

Then we can obtain the log-likelihood of the priors as:

$$\ln p(W|\lambda) = \sum_m \sum_k (\ln\lambda_k - \lambda_k W_{mk}) (31),$$

$$\ln p(H|\lambda) = \sum_k \sum_n (\ln\lambda_k - \lambda_k H_{kn}) (32).$$

The inference procedure to find the optimal values of the priors equals to the optimization process to converge to the ground-truth bases. Through the L_2-norm selection, we could discover that the vectors in W and H finally emerge to two clusters. One cluster includes the vectors whose L_2-norm is much larger than 0, while the other cluster contains the vectors of which the L_2-norm is close to 0. As a matter of fact, the vectors with large L_2- norm values are the ground-truth bases, and the others are the irrelevant bases. In addition, the number of such vectors that have larger L_2-norms is the real model order [7].

Tweedie Distribution

The β-divergence is a family of cost functions that includes the squared Euclidean distance, Kullback-Leibler and Itakura-Saito divergences as special cases. The β-divergence can be mapped to a log likelihood function for the Tweedie distribution, parametrized with respect to its mean. In particular, the values β = 0, 1, 2 underlie the multiplicative Gamma observation noise, Poisson noise and Gaussian additive observation noise respectively. The Tweedie distribution is a special case of the exponential dispersion model [14], and it has the mean and variance:

$$\mathrm{var}[\mathrm{x}] = \Phi\mu(2-\beta) \quad (33)$$

where μ = E[x] is the mean, β is the shape parameter, and Φ is referred to as the dispersion parameter The Tweedie distribution is only define for β ≤ 1 and β ≥ 2. For β ≠ 0, 1 its pdf has the form

$$\mathcal{T}(\mathrm{x}|\mu,\Phi,\beta) = \mathrm{h}(\mathrm{x},\Phi)\exp\left(\frac{1}{\Phi}\left(\frac{1}{\beta-1}\mathrm{x}\mu^{\beta-1} - \frac{1}{\beta}\mu_{\beta}\right)\right) \quad (34)$$

where h (x, Φ) is the base function. $\mathcal{T}$(x|μ, Φ, β) varies with the value of β, but the set of values that μ can take on is generally IR^{+}, except for β=2, it is IR, and the Tweedie distribution coincides with the Gaussian distribution of mean μ and variance Φ. For β = 1 and Φ = 1, the Tweedie distribution coincides with the Poisson distribution. For β = 0, it coincides with the Gamma distribution with shape parameter α = 1/μ and scale parameter μ/α. The base function admits a closed

form only for β ∈ {−1, 0, 1, 2} [15] The deviance of Tweedie distribution, i. e., the log likelihood ratio of the saturated (μ = x) and general model, is proportional to the β-divergence.

$$\log\frac{T(x|\mu=x,\Phi,\beta)}{\mathcal{T}(x|\mu,\Phi,\beta)} = \frac{1}{\Phi}d_\beta(x|\Phi) \quad (35)$$

where dβ(·|·) is the scalar cost function defined:

$$d_\beta(x|y) = \begin{cases} \frac{x^\beta}{\beta(\beta-1)} + \frac{y^\beta}{\beta} - \frac{xy^{\beta-1}}{(\beta-1)}, \beta \in R\{0,1\}, \\ x\log\frac{x}{y} - x + y, \beta = 1, \\ \frac{x}{y} - \log\frac{x}{y} - 1, \beta = 0 \end{cases} \quad (36)$$

β-divergence acts as a minus log-likelihood for the Tweedie distribution whenever the latter is defined.

Non-Informative Model for ARD Using EM Algorithm

EM algorithm is the most well-known algorithm to estimate the parameters from incomplete or mixture data in machine learning. It is the iterative algorithm through the E-step(expectation) and M-step(Maximization). In the E-step, the conditional expectation of the complete-data log-likelihood is computed on the basis of the observed data and parameter estimates. In the M-step, parameters are estimated by maximizing the complete-data log-likelihood from E-step. Therefore, EM has been applied to obtain maximum a posteriori(MAP) estimate of mixing matrix [28] such as the base and feature matrix in our model order

determination and ground-truth base recognition. In the audio-visual scene analysis, a speaker may face to the camera while he/she 4 keeps silent, or a speaker turns away of a camera while he/she is speaking. Speech signals have the sparse structure and have the mixture of different sources such as voice, noise, music background. Applying the EM on this scene is well-suited to find audio-visual clusters and to discriminate between speaking and silent people [12]. Another application of EM from incomplete data is that it is used to learn the driving behavior in multiclass users traffic flow. In this study, the speed is considered as the result of driving behavior, and the speed distribution on the road is assumed as the mixture of Gaussian distribution. EM algorithm was applied to train and classify different user-classes [29].

EM Mathematical Model [27]

Prior Assumption

Our goal is using the EM to estimate accurate the model order for nonnegative matrix factorization. This method is an extension of sparse regression via EM proposed by Figueiredo M [11]. Considering $\beta=[\beta_1, \beta_{2,\ldots,} \beta_k]$ is the hidden/missing data. If in some ways, we could observe the complete log-posterior log p (β, σ^2 |W H, β) which has the form below:

$$p(WH, \sigma^2|X, \beta) \propto p(X|WH, \sigma^2)p(WH|\beta)\ p(\sigma^2) \quad (37)$$

Denote $X \in IR^{M\times N}$ as the data matrix, base matrix $W \in IR^{M\times K}$, and feature matrix $H \in IR^{K\times N}$, we assume:

$$p(W|\beta_1) = \prod_{k=1}^{k} \mathcal{N}(W_k|0, \beta_{1k}) = \mathcal{HN}\left(W\middle|0, \psi(\beta_1)\right) (38)$$

$$p(H|\beta_2) = \prod_{k=1}^{k} \mathcal{N}(H_k|0, \beta_{2k}) = \mathcal{HN}(H|0, \phi(\beta_2)) \text{ (39)}$$

where $\psi(\beta_1) = \text{diag}(\beta_{11}^{-1}, \beta_{12}^{-1}, \ldots, \beta_{1k}^{-1})$, and $\varphi(\beta_2) = \text{diag}(\beta_{21}^{-1}, \beta_{22}^{-1}, \ldots, \beta_{2k}^{-1})$ (40)

Gaussian Log-likelihood

We are easy to obtain the Gaussian log-likelihood

$$p(X|WH, \sigma^2) = \left(\frac{1}{\sqrt{2\sigma^2\pi}}\right)^{M\times N} \prod_{M}\prod_{N} \exp\left(\frac{-1}{2}\left(\frac{X - WH}{\sigma}\right)^2\right) (41)$$

$$\text{Log } p(X|W\ H) = -N \times M\log\sqrt{2\pi\sigma} - \frac{1}{2\sigma^2}\sum_{M}\sum_{N}(X - WH)^2 \ (42)$$

EM Algorithm Implementation

First, apply logarithms to (15) since p (σ^2) is flat, we have:

$$\log p(WH, \sigma^2|X, \beta) \propto \log p(X|WH, \sigma^2) + \log p(W|\beta_1) + \log p(H|\beta_2)$$

$$\propto -M\text{x}N log(\sigma^2) - \frac{||V - WH||^2}{\sigma^2} - W^T\psi W - H^T\phi H \ (43)$$

Second, From (19), the complete log-posteriors is linear with respect to ψ, and φ, and other two terms do not depend on β, the E-step reduces to computing the conditional expectation of ψ, and φ, given and the current $\widehat{\sigma_t^2}$, $\widehat{W_t}$, $\widehat{H_t}$which we denote as

$$P_{1(t)} = E[\psi(\beta_1)|X, \widehat{\sigma_t^2}, \widehat{W_t}]$$

$$= diag\{E[\beta_1^{-1}|\widehat{\sigma_t^2}, \widehat{W_t}], \dots, E[\beta_{1k}^{-1}|\widehat{\sigma_t^2}, \widehat{W_t}]\} \quad (44)$$

$$P_{2(t)} = E[\phi(\beta_2)|X, \widehat{\sigma_t^2}, \widehat{W_t}]$$

$$= diag\{E[\beta_2^{-1}|\widehat{\sigma_t^2}, \widehat{W_t}], \dots, E[\beta_{2k}^{-1}|\widehat{\sigma_t^2}, \widehat{W_t}]\} \quad (45)$$

As for $p(\beta_1|X, W, \sigma^2) = p(\beta_1|W)$ because given W, and β_1 does not depend on X, σ^2, or H. So, $p(\beta_1|X, \widehat{\sigma_t^2}, W) \propto p(\widehat{W_t}|\beta_1)p(\beta_1)$. Similarly, we could get the same thing $p(\beta_2|X, \widehat{\sigma_t^2}, H) \propto p(\widehat{H_t}|\beta_2)p(\beta_2)$.Since $p(W|\beta_1) = \mathcal{HN}(W|0, \psi(\beta_1))$. $p(\beta_1)$, and $p(\beta_2)$ are the exponential hyper-priors, elementary integration yields:

$$E[\beta_{1,i}^{-1}|X, \widehat{\sigma_t^2}, \widehat{W_t}] = \frac{\int_0^\infty \frac{1}{\beta_{1,i}} \mathcal{HN}(\widehat{W_t}|0, \beta_{1,i}) \frac{\gamma_1}{2} \exp\left(-\frac{\gamma_1}{2}\beta_{1,i} d\beta_{1,i}\right)}{\int_0^\infty \mathcal{HN}(\widehat{W_t}|0, \beta_{1,i}) \frac{\gamma_1}{2} \exp\left(-\frac{\gamma_1}{2}\beta_{1,i} d\beta_{1,i}\right)} = \frac{\sqrt{\gamma_1}}{|\widehat{W_t}|} \quad (46)$$

Similarly, we can obtain:

$$E[\beta_{2,i}^{-1}|X, \widehat{\sigma_t^2}, \widehat{W_t}] = \frac{\sqrt{\gamma_2}}{|\widehat{H_t}|} \quad (47)$$

<u>E step.</u> Thus,

$$P_{1(t)} = \sqrt{\gamma_1} diag\{|\widehat{W_{1(t)}^{-1}}|, \widehat{W_{2(t)}^{-1}}, \dots, \widehat{W_{k(t)}^{-1}}\} \quad (48)$$

$$P_{2(t)} = \sqrt{\gamma_2} diag\{|\widehat{H_{1(t)}^{-1}}|, \widehat{H_{2(t)}^{-1}}, \dots, \widehat{H_{k(t)}^{-1}}\} \quad (49)$$

The Q-function, the expected value with respect to W and H as the missing variables of the complete log -posterior, is obtained by plugging $P_{1(t)}$ and $P_{2(t)}$ in the place of ψ and φ

$$Q\left(WH, \sigma^2 \middle| \widehat{W}_{(t)}, \widehat{H}_{(t)}, \widehat{\sigma^2}_{(t)}\right)$$

$$= -M \times Nlog(\sigma^2) - \frac{||X - WH||^2}{\sigma^2} - W^T P_1(t) W - H^T P_1(t) H \quad (50)$$

Finally, the M-step consists in maxing $Q(\mathrm{WH}, \sigma^2 | \widehat{\mathrm{W}}_{(t)}, \widehat{\mathrm{H}}_{(t)}, \widehat{\sigma^2}_{(t)})$ with respect to σ^2 and WH, yielding:

M-step.

$$\widehat{\sigma}^2_{t+1} = \mathrm{argmax}_{\sigma^2}\left(-\mathrm{M} \times \mathrm{N}\log(\sigma^2) - \frac{||\mathrm{X} - \mathrm{WH}|_2^2}{\sigma^2}\right)$$

$$= \frac{||\mathrm{X} - \mathrm{WH}|_2^2}{\mathrm{MN}} \quad (51)$$

$$\widehat{\mathrm{WH}}_{(t)} = \mathrm{argmax}_{\beta}\left(-\frac{||\mathrm{X} - \mathrm{WH}|_2^2}{\sigma^2} - \mathrm{W}^{\mathrm{T}}\mathrm{P}_1(t)\mathrm{W} - \mathrm{H}^{\mathrm{T}}\mathrm{P}_2(t)\mathrm{H}\right) \quad (52)$$

And then, we need to take the integral of (26), and we have:

$$-Q\left(W, H \middle| \widehat{W}, \widehat{H}\right) = \frac{1}{2}||\mathrm{V} - \mathrm{WH}|_2^2 + \frac{1}{2}Tr(WVW^T) + \frac{1}{2}Tr(H^T TH) \quad (53)$$

$$\frac{\partial \mathrm{Q}}{\partial \mathrm{W}} = -\mathrm{XH}^T + \mathrm{W}\,\mathrm{HH}^T + \mathrm{W}\,\mathrm{V} \quad (54)$$

$$\frac{\partial \mathrm{Q}}{\partial \mathrm{H}} = -\mathrm{W}^T\mathrm{X} + \mathrm{W}^T\mathrm{WH} + \mathrm{TH} \quad (55)$$

Where, V, and T are diagonal matrix of estimated $\beta_{1,i}, \beta_{2,i}$ respectively. Finally, we can get the updating rules for W^* and H^*:

$$W^* = W \frac{X\mathrm{H}^T}{(WH\mathrm{H}^T + WV)} \quad (56)$$

$$H^* = H \frac{\mathrm{W}^T X}{(\mathrm{W}^T WH + TH)} \quad (57)$$

CHAPTER FOUR

NMF BASED ON SPARSENESS CONSTRAINTS

Sparse Coding

The approach of sparse distributed coding indicates that there are very few active units corresponding the large input datasets [30]. Therefore, sparseness is the effective representation of the data in which redundant features have very low probability (close to zero) and represented features have higher probability (greater than zero). Therefore, sparseness representation has the ability to represent basic components of the objects. Figure 8 is to illustrate the sparse coding diagram in which a very few of output actively represents multi-dimensional data inputs (e.g. only 4 actively unit outputs correspond to multi input vector).[30]

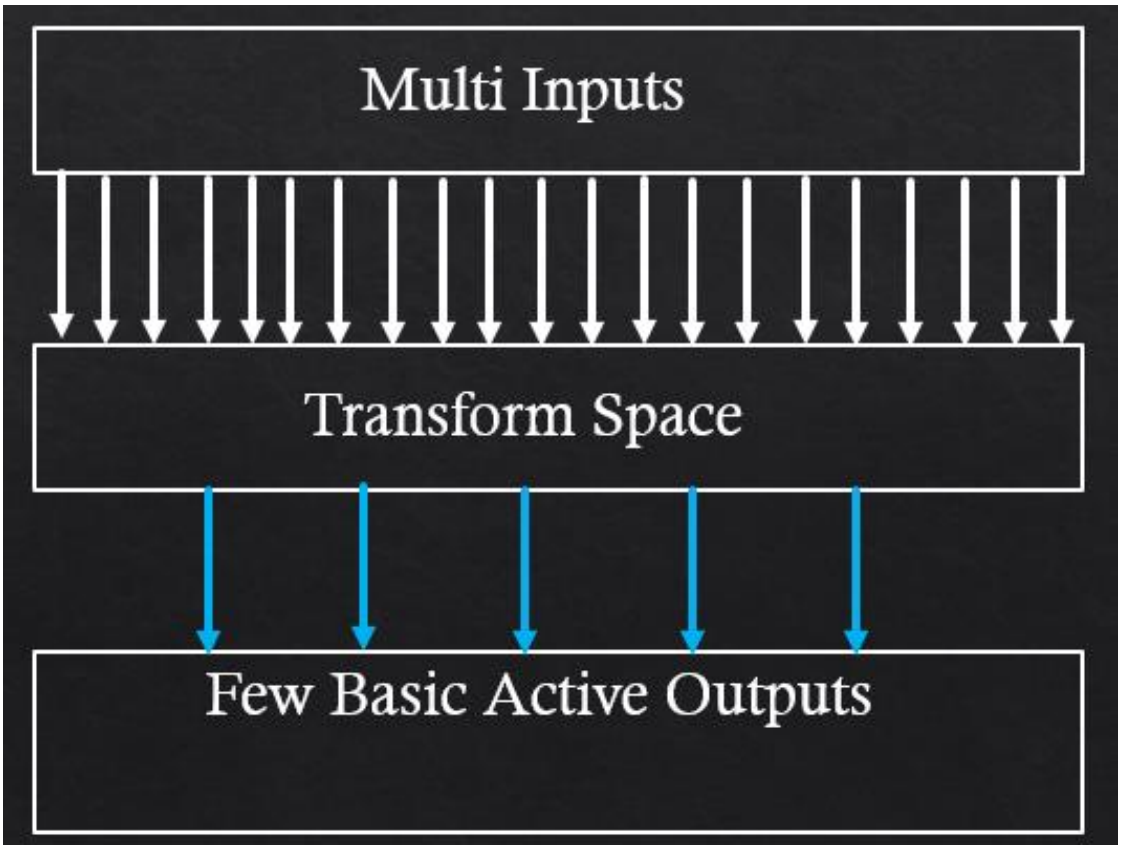

Figure 8. The Sparse Coding Diagram

Sparseness is also applied in the image processing process learning about the objects. Figure 9 is to illustrate the sparse coding network. The image patch shows the 12× 12 pixel values on the pixel values bar chart, and inputs are transformed to the sparser scheme as shown on the top bar chart

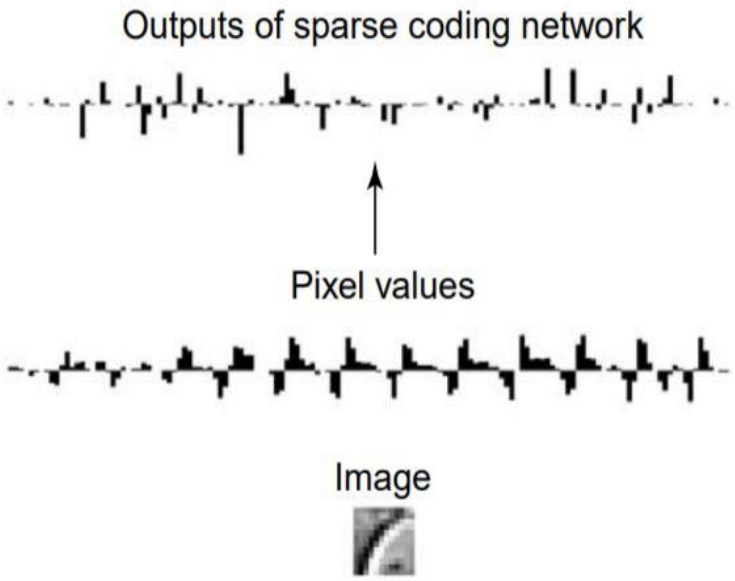

Figure 9. The Sparse Coding Network for the Image [14]

Sparseness Constraints Concepts

The sparseness constraints mentioned above is the representation that there are few active units as the output vector [30]. Indeed, the inactive units have the values closely to zero while the significant units have higher values than zero. Figure 10 is to illustrate the representation and different sparseness constraints level on four different output vectors

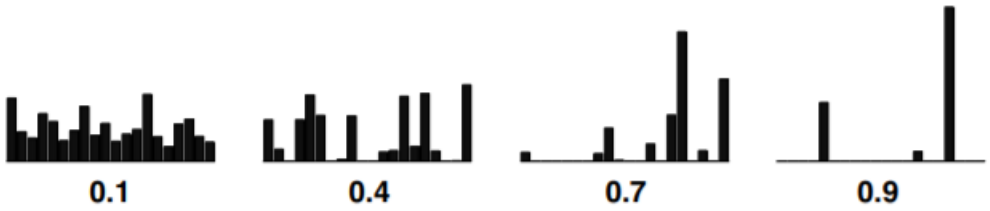

Figure 10. Different Sparseness Constraints Level on Vectors [5]

Another example of the sparsity is applied in face recognition. Figure 11 is to illustrate variety of sparseness constraints on ORL faces. When we applied the sparseness constraints level (0.5), we can get the whole faces globally (Figure 11-a). However, when we increase the level to 0.6 (Figure 11-b), the whole faces gradually change to local features. At this point, we can see more clearly eyes, noses, lips,…, etc. Finally, we change the constraints level to 0.75 (Figure 11-c), the global faces convert to local features completely.

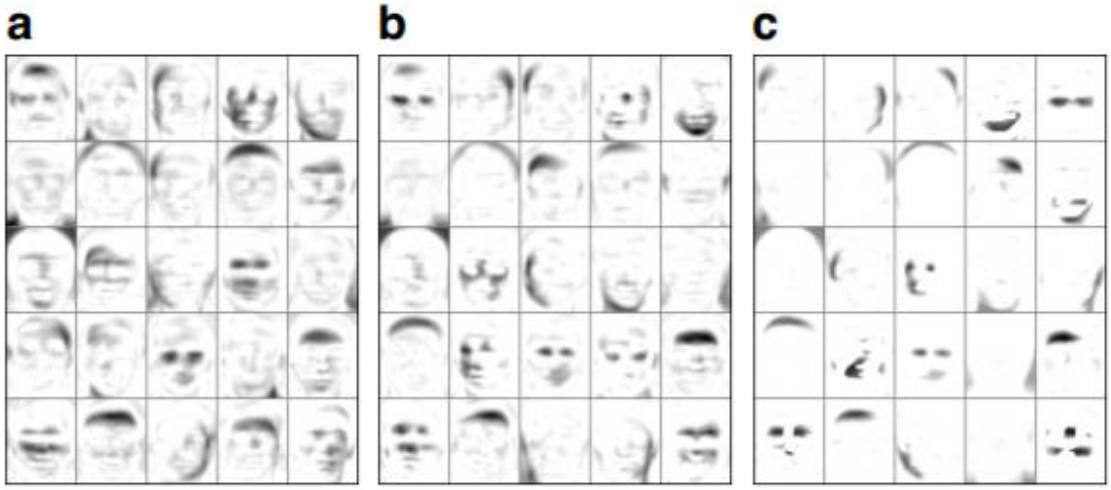

Figure 11. Changes of Sparseness Constraints Level on ORL Faces [5]

There are many sparseness measures proposed on research papers recently. The general idea is mapping from R^n to R to measure the energy of a vector consisting of few active units. The simple sparseness measure is computed based on the relationship between the L_1 norm and L_2 norm [5]:

$$sparseness(x) = \frac{\sqrt{n} - (\sum|xi\ |)/\sqrt{\sum x_i^2}}{\sqrt{n} - 1} \quad (58)$$

where n is the dimensionality of x. he function above will reach the maximum value at one if and only if x contains only a single non-zero component, and reach the zero values if and only if all components are equal interpolating smoothly between the two extremes [5]

NMF with Sparseness Constraint

Our goal is to apply constraints levels on NMF to find the optimal sparseness values. But in the real application, we do not know how much sparseness constraints we should apply on W and H. Generally, it depends on the application we are working on to apply suitable constraints levels to get better understanding of data

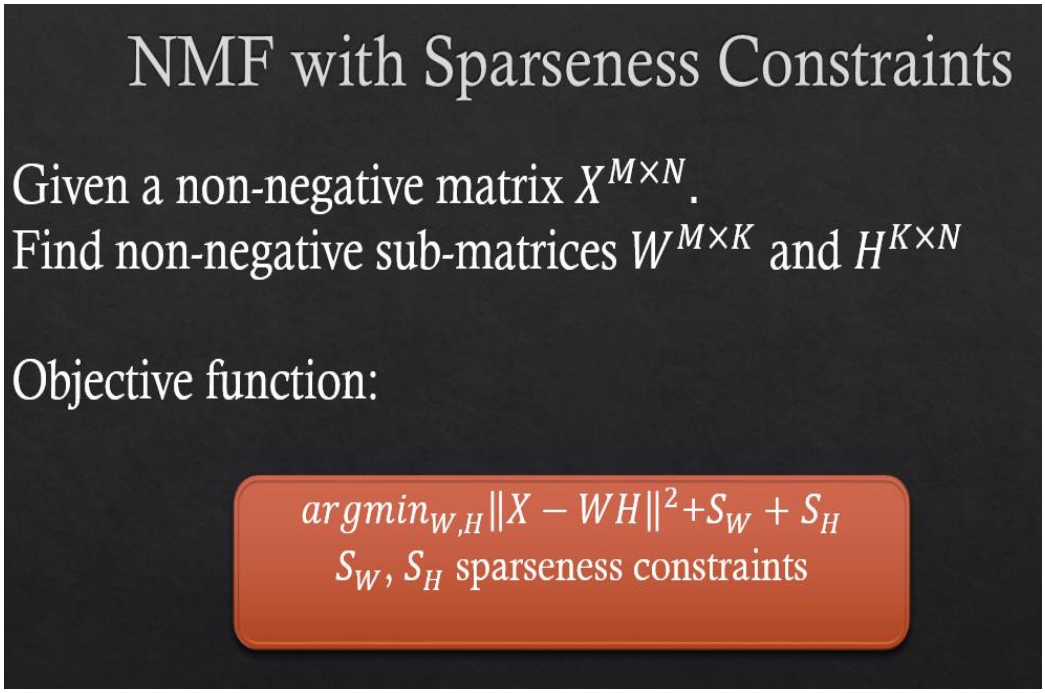

Figure 12. NMF with Sparseness Constraints

Sparseness Constraints Mathematical Model

In order to enforce sparseness on *W* or *H* in the NMF presented in Equation (1). Two formulations are the corresponding algorithms for sparse NMFs, i.e.

SNMF/L for sparse *W* (where 'L' denotes the sparseness imposed on the left factor) and SNMF/R for sparse *H* (where 'R' denotes the sparseness imposed on the right factor). The sparse NMF formulations that impose the sparsity on a factor of NMF utilize L_1-norm minimization [15]

SNMF/R. To apply sparseness constraints on *H*, we formulate the following SNMF/R optimization problem from Equation (1)

$$\min(W,H) = \frac{1}{2}\{||X - WH||_F^2 + \eta||W||_F^2 + \beta\sum_{j=1}^{n}||H(:,j)||_1^2\} \quad (59)$$

where H(:,j) is the *j*-th column vector of *H*, η is a parameter to suppress , and $||W||_F^2$ is a regularization parameter to balance the trade-off between the accuracy of the approximation and the sparseness of *H*

SNMF/L. To impose sparseness constraints on *W*, we introduce the SNMF/L formulation

$$\min(W,H) = \frac{1}{2}\{||X - WH||_F^2 + \eta||H||_F^2 + \alpha\sum_{j=1}^{n}||W(i,:)||_1^2\} \quad (60)$$

where *W (i, :)* is the *i*-th row vector of *W*, η is a parameter to suppress , and $||H||_F^2$ is a regularization parameter to balance the trade-off between the accuracy of the approximation and the sparseness of *W*

CHAPTER FIVE

NMF BASED ON GRAPH EMBEDDING

Introduction

Pattern recognition and classification tasks have been paid attention recently and applied widely in computer vision, object recognition. Yan et al. [31] suggested that most of machine learning algorithms can be implemented on the general framework called graph embedding. Graph embedding is one of the machine learning techniques used in dimensionality reduction. In the graph embedding framework, the original data is decomposed into 2 parts as known as the intrinsic graph and penalty graph. In the intrinsic graph, a dataset has been characterized by data pairs which are similar. On the contrary, the penalty graph has been characterized by unfavorable relationship of the original data. Finally, 2 parts have been connected to reconstruct the original data approximately [32]

Graph Creation

Supposedly, we have the set of data, we can build up the intrinsic graph G which is undirected and weighted. Let X be the set of vertices of graph G, and let E be the set of edges of graph G, and the edges is the connection of similar pairs of the original data, and G can be denoted by this form G = {X, E}. *Figure 13* illustrates the graph construction from intrinsic graph and penalty graph. In the

intrinsic graph, each point in the same class has been grouped based on the k-nearest neighbors (left hand side). In the penalty graph, each point characterized by unfavorable relationship on each class connected to another class which also is characterized by unflavored similarities (right hand side).

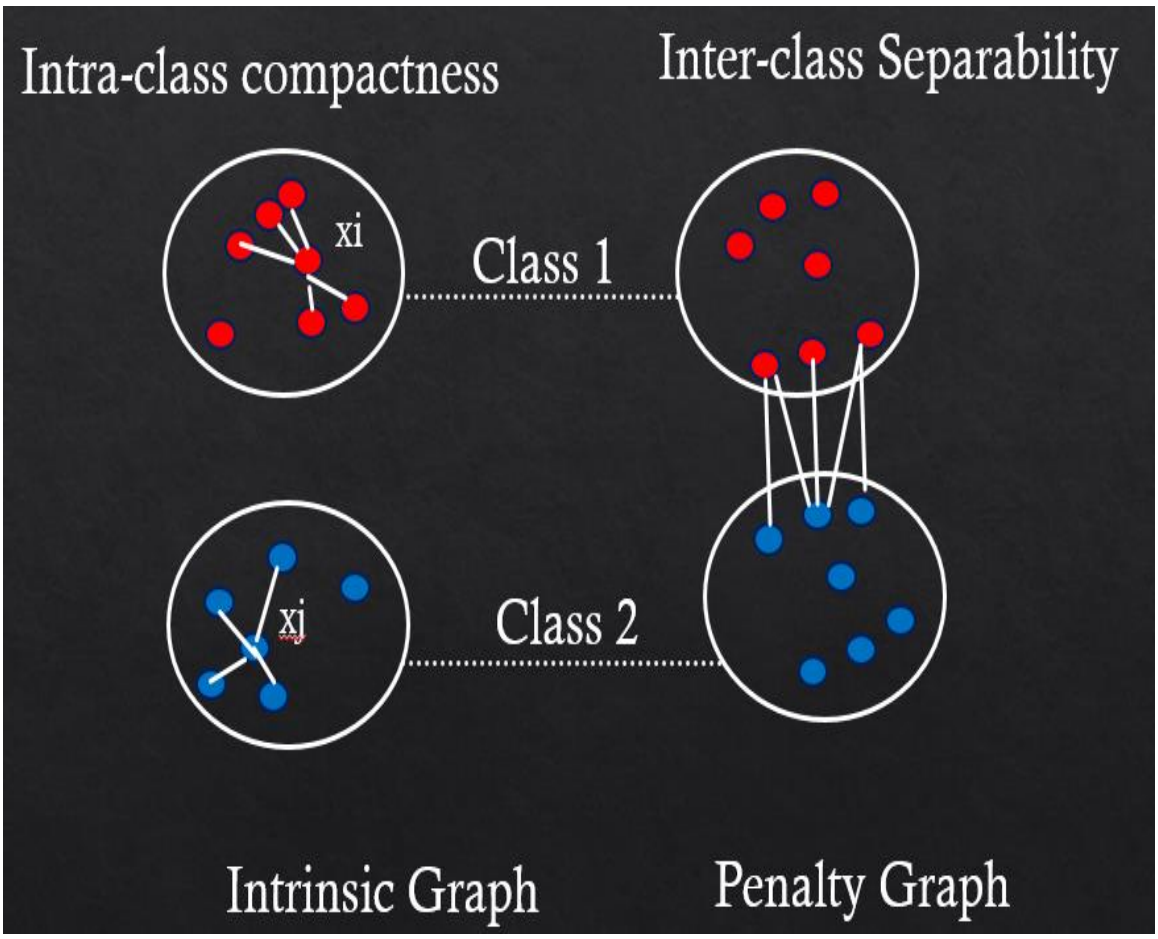

Figure 13. Adjacency Relationship Intrinsic and Penalty Graph [31]

Graph Embedding

Non-Negative Matrix Factorization (NMF) factorizes the data matrix X into one lower-rank non-negative basis matrix and one non-negative coefficient matrix. Its objective function is:

$$\frac{\min}{W,H}||X - WH||, s.t\ W, H \geq 0\ (61)$$

Yan et al. [9] claimed that most of them can be explained within a unified framework, called graph embedding. Let G = {X, S} be an undirected weighted graph with vertex set X and similarity matrix S ∈ $R^{N\times N}$. Each element of the real symmetric matrix S measures for a pair of vertices the similarity, which is assumed to be non-negative in this work. The diagonal matrix D and the Laplacian Matrix L of a graph G are defined as:

$$L = D - S, D_{ii} = \sum_{i\neq j} S_{ij}, \forall\ \mathrm{i}\ (62)$$

Graph embedding generally involves an intrinsic graph G, which characterizes the favorite relationship among the training data, and a penalty graph Gp = {X, Sp}, which characterizes the unfavorable relationship among the training data, with Lp = Dp − Sp, where Dp is the diagonal matrix as defined in Eq.(3) [10]

<u>Graph Embedding Mathemcatical Model</u>

For given H = H_t, update the basis matrix W as:

$$W_i^{t+1} = \lambda X_i H^T (K(W_i^t) + 2D^h)^{-1}\quad (63)$$

Where: $K^i = \mathrm{diag}\{K^1(h^t_1)_{ii}, \cdots, K^N(h^t_N)_{ii}\}$, and W_i^{t+1} is the i^{th} row vector of W^{t+1}, H^T is the transpose matrix of H,

For given $W = W^{t+1}$, update the matrix H as:

$$H_i^{t+1} = \lambda w_i^T X\left(K^i + 2L^p\right)^{-1} \quad (64)$$

H_i^{t+1} is the i^{th} row vector of H^{t+1}, W^T is the transpose matrix of W [9]

CHAPTER SIX

EXPERIMENTS AND EVALUATIONS

In this section, we will evaluate our proposed method (EM) on 5 different datasets: Fence, Swimmer, ORL faces, Japanese faces, and Yale extension faces. The data is from simple one: Fence, Swimmer, to complicated one: faces

Datasets

The Fence Dataset

The Fence data is the synthetic dataset introduced by Sun et al [7]. It is consisting of 69 binary images having 32x32 pixel image on each. Each image has 4 vertical bars and 4 horizontal bars placed on different position from top to bottom, and from left to right. We classify this dataset into 4 groups based on the number of bars (not position of each bar). For more detail, group 1 has 2-bar images, group 2 has 4-bar images, group 3 has 6-bar images, and finally group 4 has 8-bar images. Figure 14 illustrates the 16 samples of Fence dataset

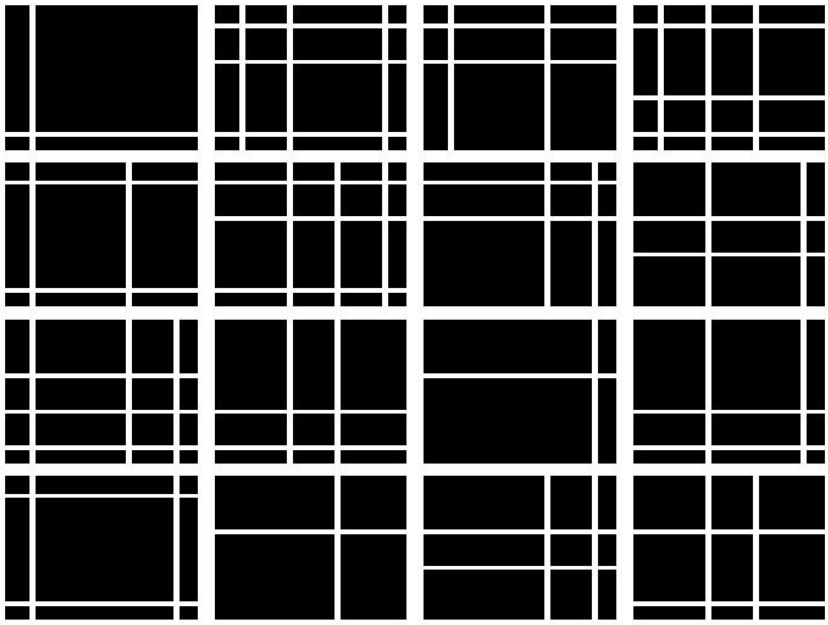

Figure 14. The Sample of Fence Dataset [7]

The Swimmer Dataset

The Swimmer dataset [8] is the one of the most famous synthetic dataset used in machine learning research because its simplicity. It is containing the set of 256 images, each image illustrates on the subplot with one static part called torso, and 4 moving parts called the limbs, each part has four different positions. The goal is to use our proposed method (EM) to extract 16 limb position and one torso separately[16]. There are 256 images on this set, so we separate it into 4 groups, each group is containing 64 consecutive images. Figure 15 illustrates the 16 samples of Swimmer dataset.

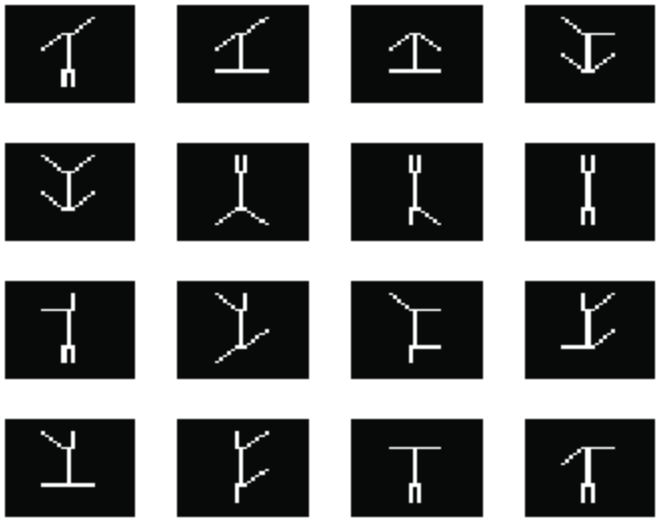

Figure 15. Sample of Swimmer Dataset [8]

The ORL Faces Dataset

The ORL face dataset is including 400 face images of 40 people. There are we have 10 samples on each person. The image of each person is taken at various conditions such as: the different level of light intense, opening and closed eyes, smiling or not smiling, wearing glasses or not wearing glasses., and then each image is cropped into 32x32 image pixels. Figure 16 is to illustrate some samples of face image from ORL dataset.[21].

Figure 16. The Sample Images of ORL Dataset [21]

The JAFFE Faces Dataset

JAFFE dataset [33] is consisting of 213 face images from 10 Japanese female models [34]. There are 7 facial expressions (6 facial expressions and 1 neutral one) on this dataset. Each person shows different expressions such as: angry, disgust, fear, happy, sad, and surprise [35], and we cropped these images into 49x49 image pixels. Figure 17 is to illustrate the sample of JAFFE dataset.

Figure 17. The Sample of Japanese Faces Dataset [34]

The Extended Yale Faces Dataset

The extended Yale Faces is consisting of 16128 images from 28 people with 9 poses and 64 illumination conditions [36], and then the data is cropped in to 32x32 pixel images [37]

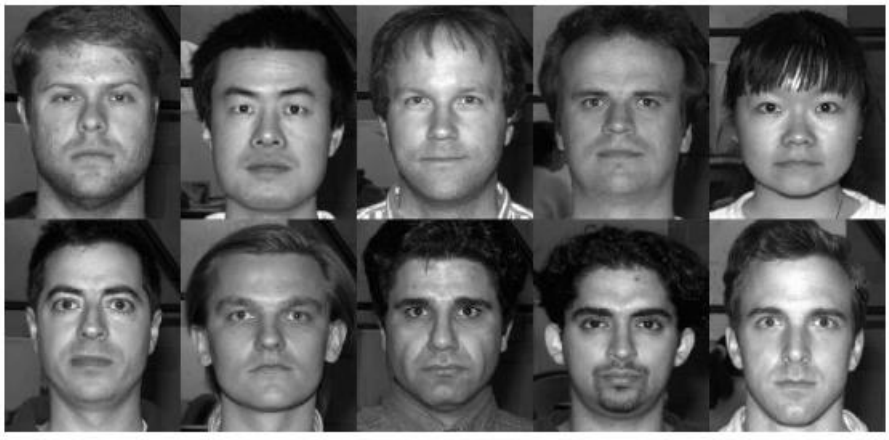

Figure 18. Sample Faces Images of Extended Yale Dataset [36]

Ground Truth Bases Discovery

Ground truth bases discovery is often used as the basis for training pattern recognition algorithms to generate thematic maps or to detect objects of interest [38]. High accuracy ground-truth data plays the important roles for the development and evaluation of algorithms related to computer vision[39]

In this section, we will apply our proposed algorithm called EM on different datasets mentioned above: Fence, Swimmer, ORL, and Jaffe to extract ground truth bases data. Our advantage is that it could discover the ground truth bases from datasets while the other methods: non-negative matrix factorization (NMF)[11], principle component analysis (PCA)[4], NMF with sparseness constraint on W or H (NMFSC) [5] and graph embedding (GE) [18] couldn't discover it.

First of all, we will split our data in 10-fold cross validation. It means the data will be split into 10 random partition as a training set (90%) and a test set (10%), and then we will run 5 epochs in which each epoch will consist of 1 one full training cycle. This set up will be applied to all datasets. For each specific data, we will edit or add more parameters that is suitable to our situations.

For the Fence data, there are 4 vertical bars and 4 horizontal bars. Therefore, we have totally 8 ground truth bases, and the EM can discover exactly 8 ground-truth bases. We will choose the initial base number that is greater than the number of ground truth bases. In this experiment, we will set the initial model order to be 18 components, and we will run 1000 iteration on each training cycle. Figure 19 to Figure 23 is to illustrate how EM can discover the number of ground-truth bases compared to others (traditional NMF,

PCA, NMF with sparseness constraints and NMF with graph embedding). As you can see, the EM method has extracted correctly the number of ground-truth bases as expected while the other methods (NMF, PCA, NMF with sparseness constraints, NMF with Graph embedding) cannot extract the unique components as EM. Moreover, there are lots of duplicated components on each subplot when the other methods have been applied. There are 8 bases components with 1 variation via EM based ARD.

Figure 19. Ground Truth Bases Discovery via EM Based ARD for Fence

Figure 20. Pattern Discovery via NMF for Fence

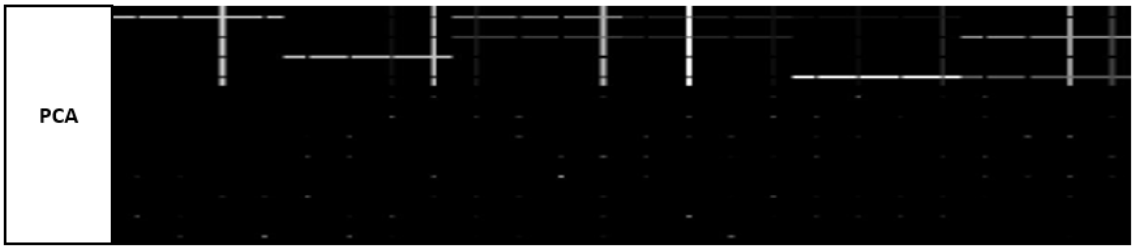

Figure 21.Pattern Discovery via PCA for Fence

Figure 22. Pattern Discovery via NMF with Sparseness Constraint for Fence

Figure 23. Pattern Discovery via NMF with Graph Embedding for Fence

For the Swimmer dataset, we already know there are 16 limb positions, and one static torso. Our goal will extract ground truth bases from swimmer datasets, and it should extract 16 unique patterns (17 unique patterns if we include one static torso) as expected. The setup is also similar to fence dataset, but the only difference is that we will choose the initial model order K= 25, and we will run 1000 iterations on each training cycle. is to illustrate the EM extracts exactly 16 ground truth bases of swimmer dataset on different subplots while these others cannot recover them. They have more than one component on each subplot compared to EM. In PCA method, it just discovers principle components on first 12 subplots, and eliminates components that are less important. Therefore, PCA might miss some necessary components from dataset. As the result, there are 16 ground truth bases with variation of 1 component when running 5 full training cycles. *Figure 24* to Figure 28 illustrates the ground truth bases images via EM based ARD method compared to other methods (PCA, NMF, NMF with sparseness constraint, and NMF with graph embedding)

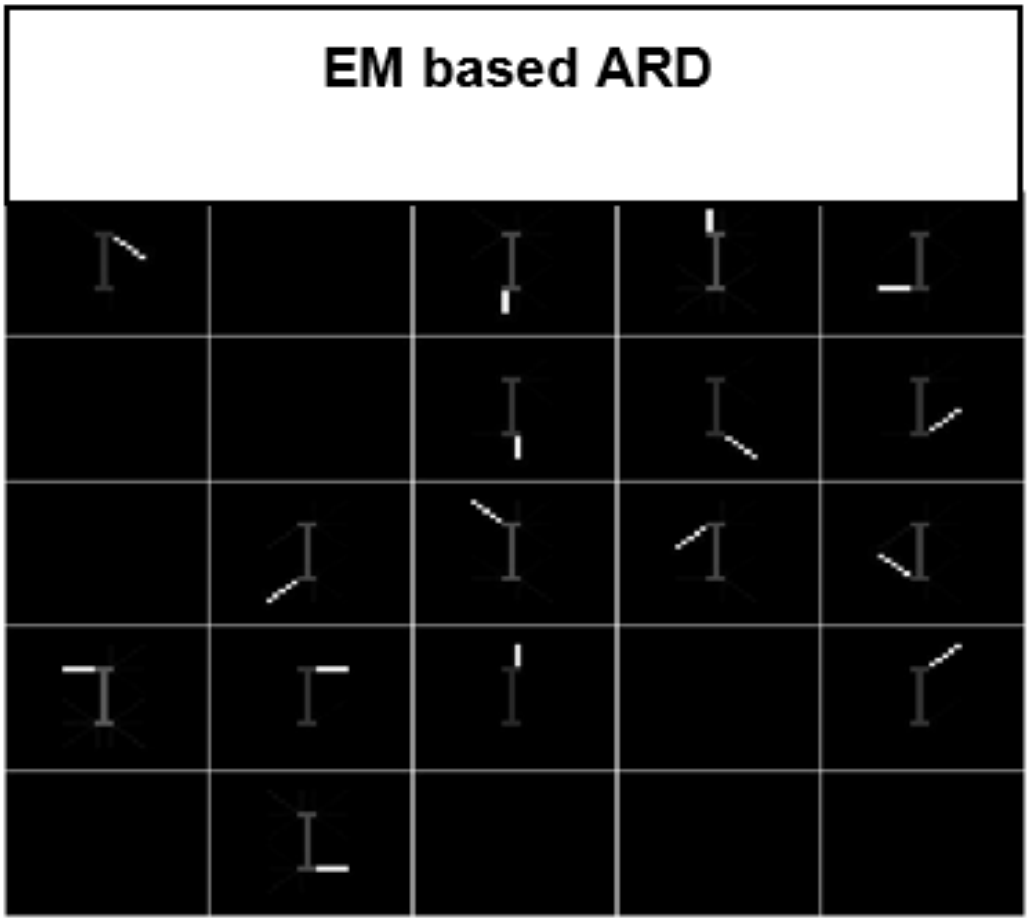

Figure 24. Ground Truth Bases Discovery with EM Based ARD of Swimmer

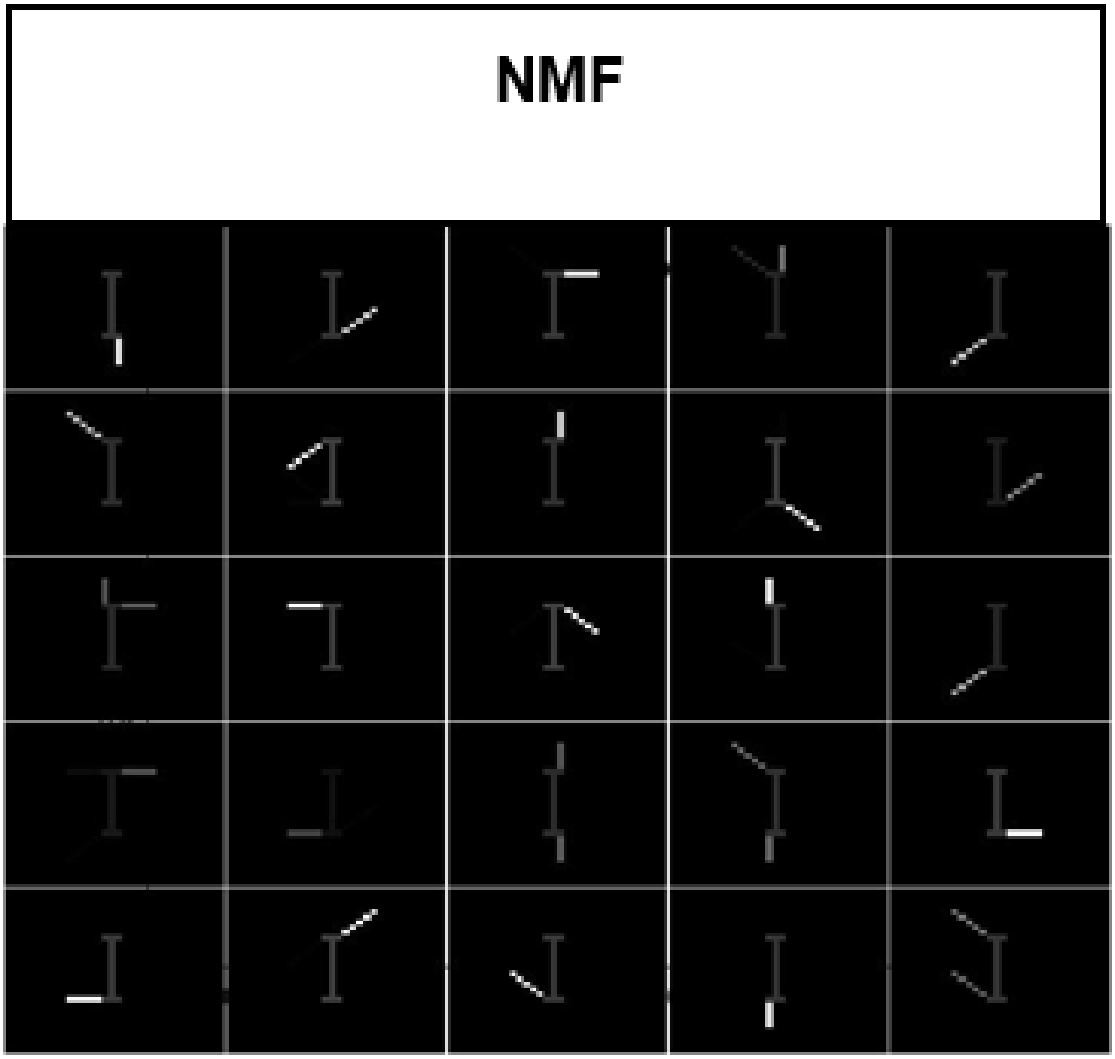

Figure 25. Basic Images Discovery with NMF for Swimmer

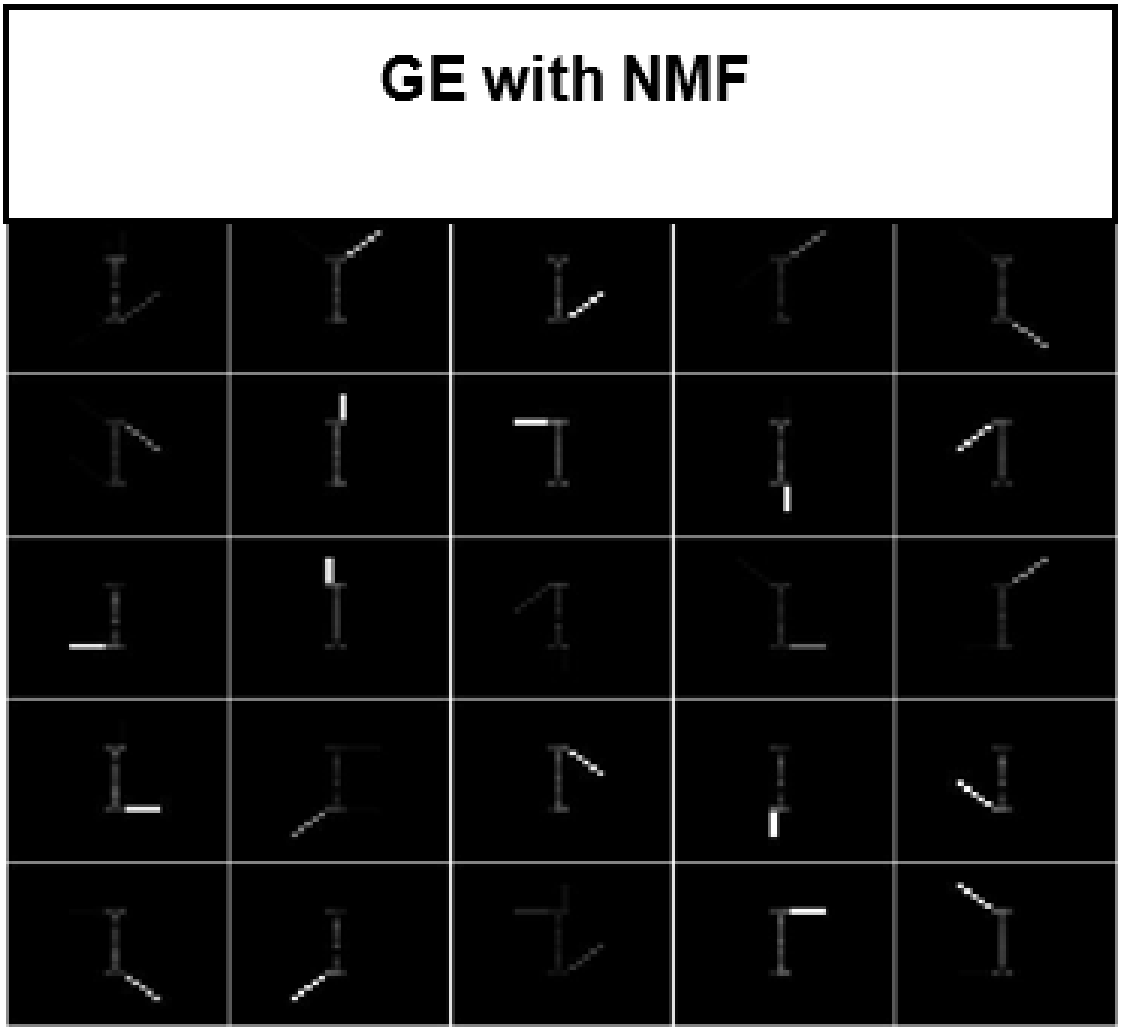

Figure 26. Basic Images Discovery via NMF with GE for Swimmer

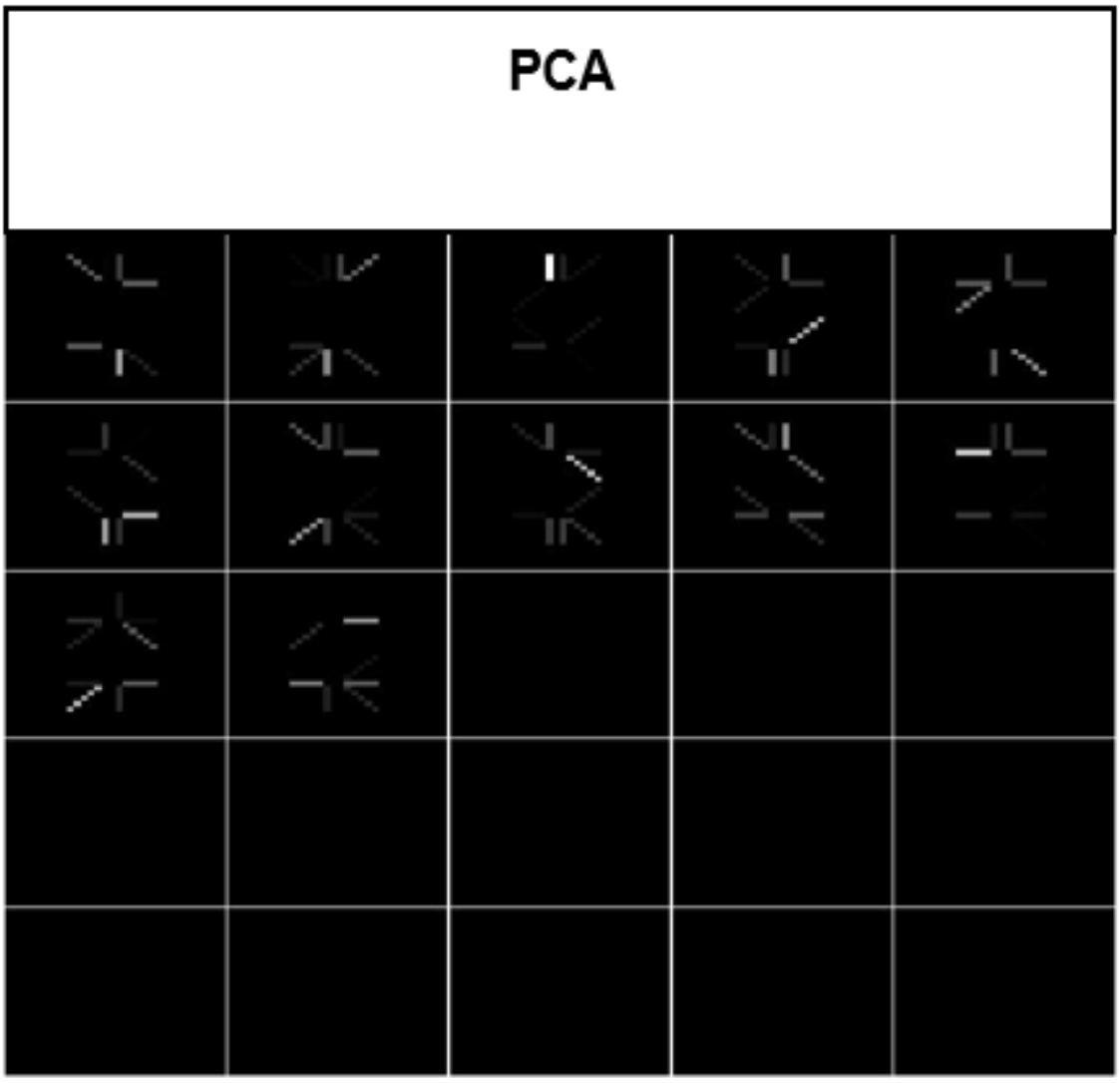

Figure 27. Basic Images Discovery via PCA for Swimmer

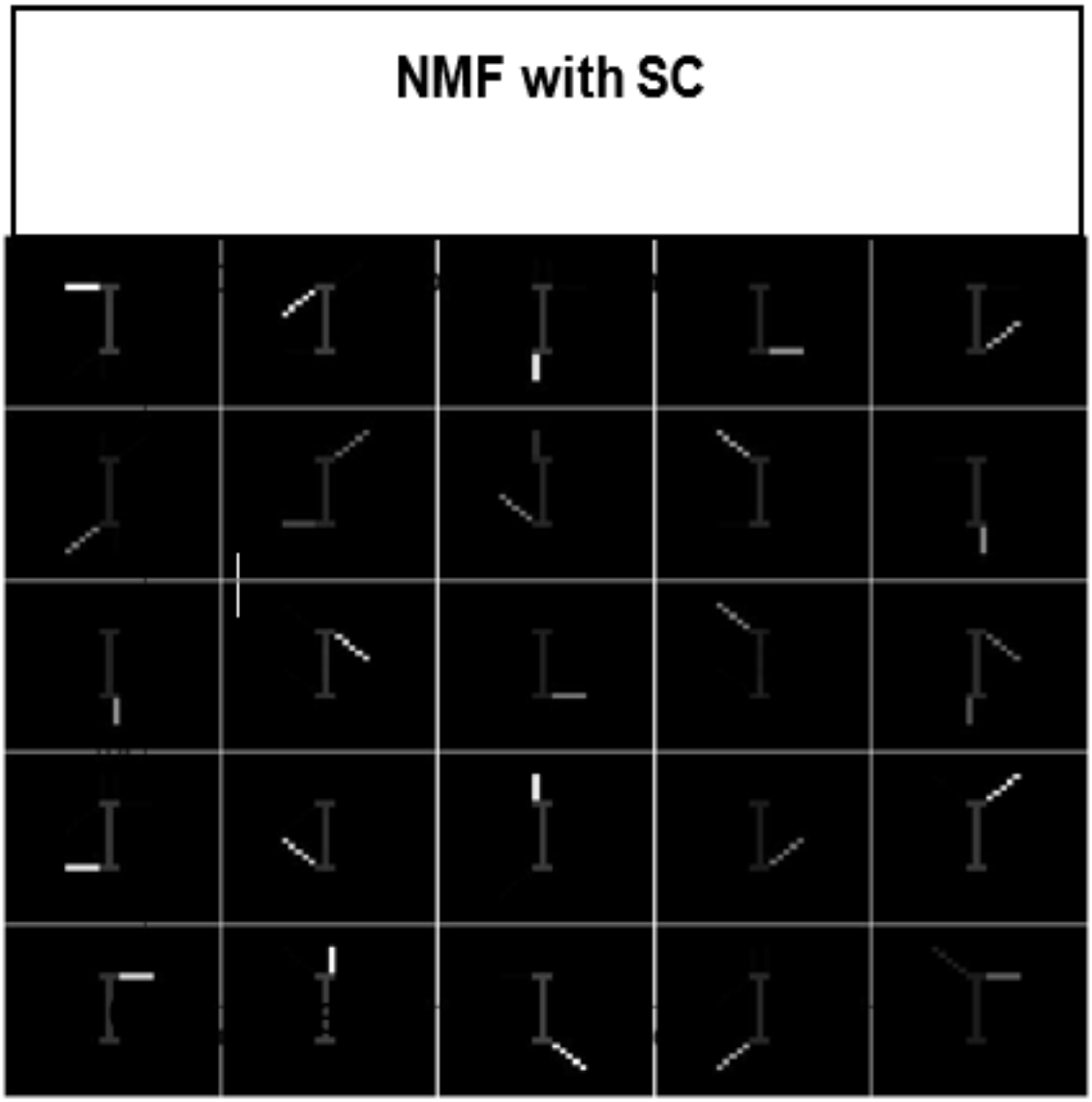

Figure 28. Basic Images Discovery via NMF with SC for Swimmer

Now, we move forward to the ORL faces data which is real and complicated to see how our proposed method can discover basic components from the data. Because we do not have any prior knowledge of ground truth bases components from this data, so

we will choose the initial value of model order K=100,121, 144 to be big enough, and we will run 4000 iterations on each training cycle, and then apply EM method to see how many basic components EM can discover. Finally, we get the result of 62 ground truth bases with variation of 7 components. *Figure 29* to *Figure 33* is to illustrate the basic components on each method, and EM takes over the other ones when it can recover the ground truth bases from dataset such as: mouth, eyebrows, eyes, nose,…

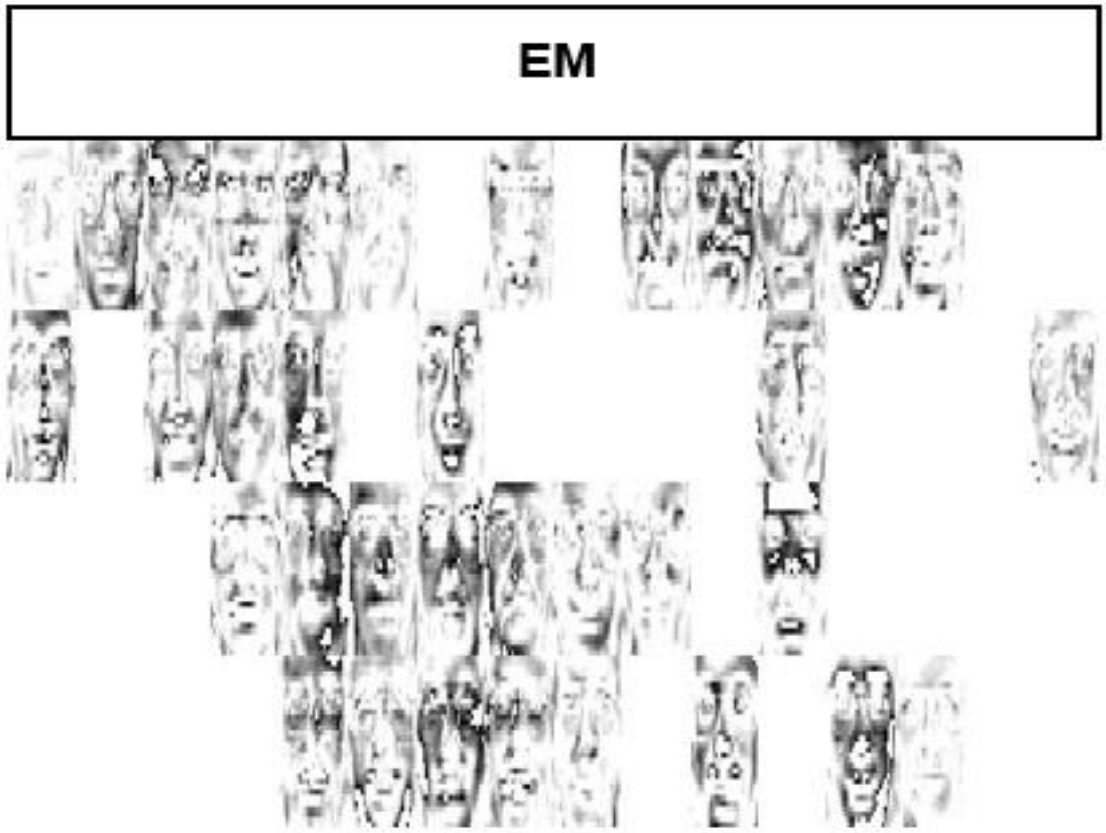

Figure 29. Ground Truth Bases Discovery via EM Based ARD for ORL

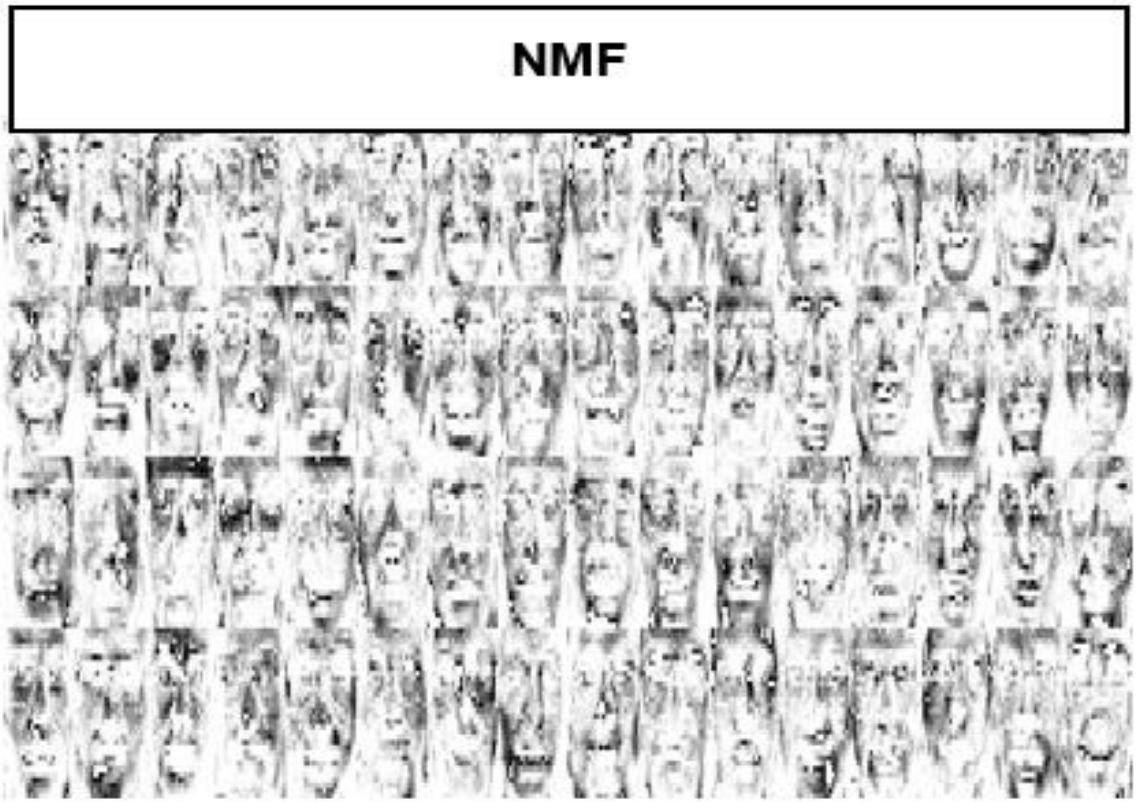

Figure 30. Basic Images Discovery via NMF for ORL

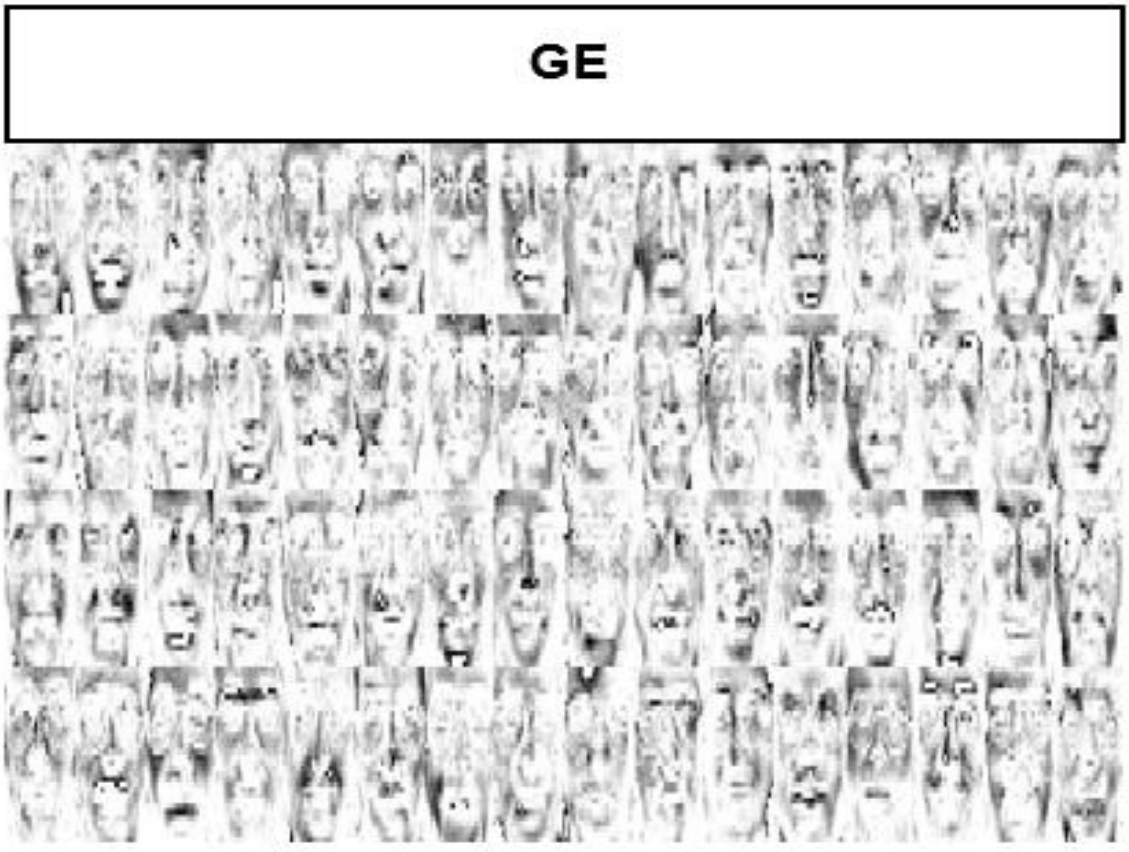

Figure 31. Basic Images Discovery via NMF with GE for ORL

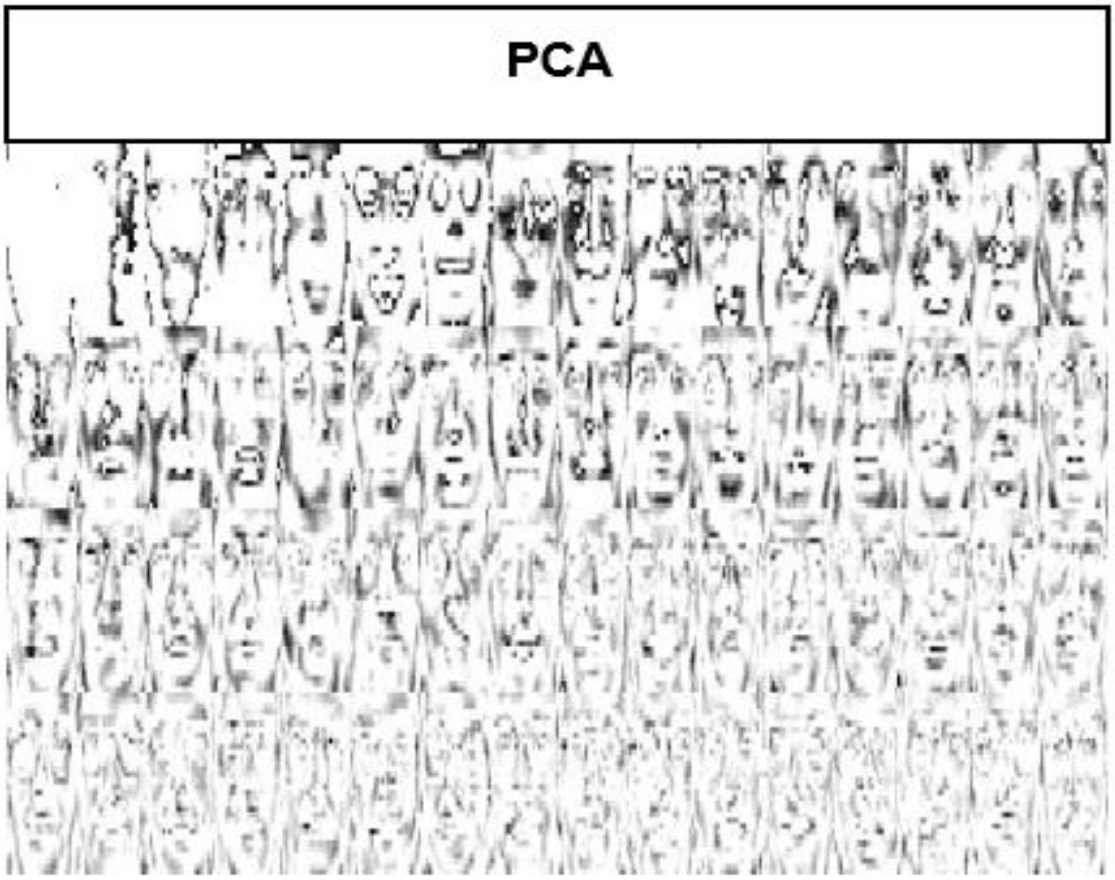

Figure 32. Basic Images Discovery via PCA for ORL

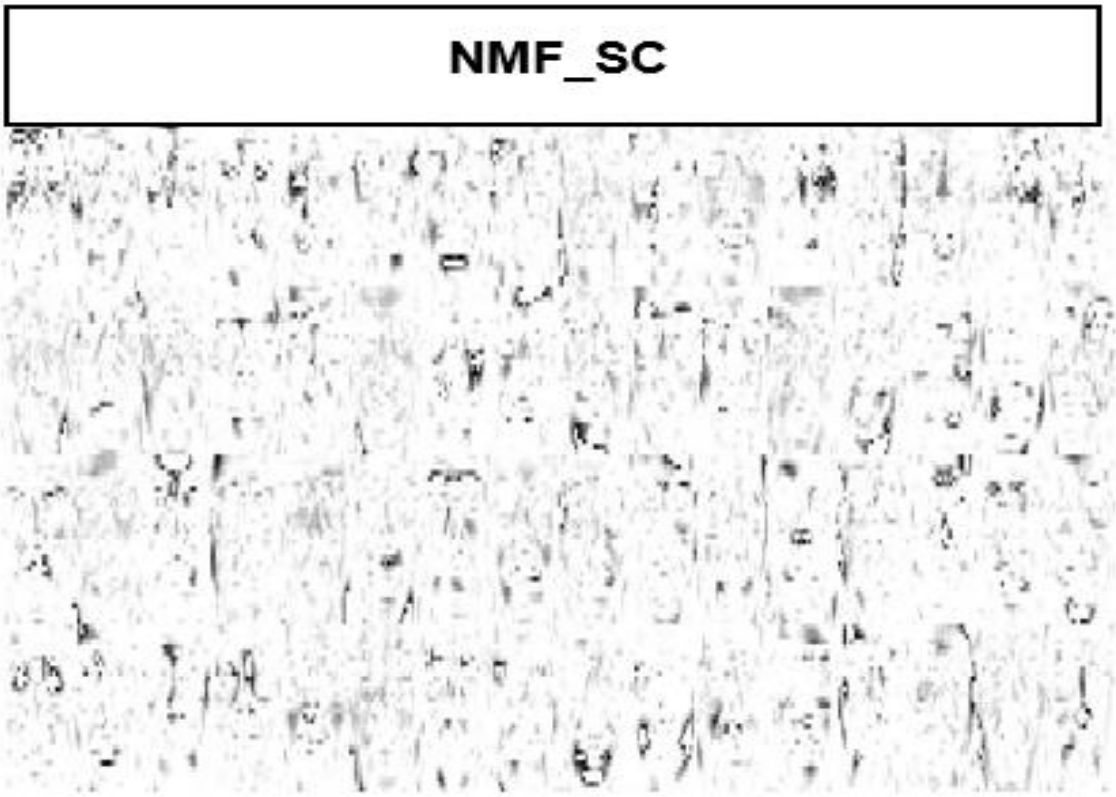

Figure 33. Basic Images Discovery via NMF with SC for ORL

The Jaffe faces dataset is similar with ORL faces datasets, and we also do not know the correct number of ground truth bases. Therefore, we have to choose the initial model order K=100,1221,144 to define the estimated number of basic components. *Figure 34* to *Figure 38* is to illustrate the sample of basic components from the Jaffe dataset. Apparently, EM can extract the unique pattern from the dataset compared to the others. The result shows that there are 64 ground truth bases with variation of 3 components.

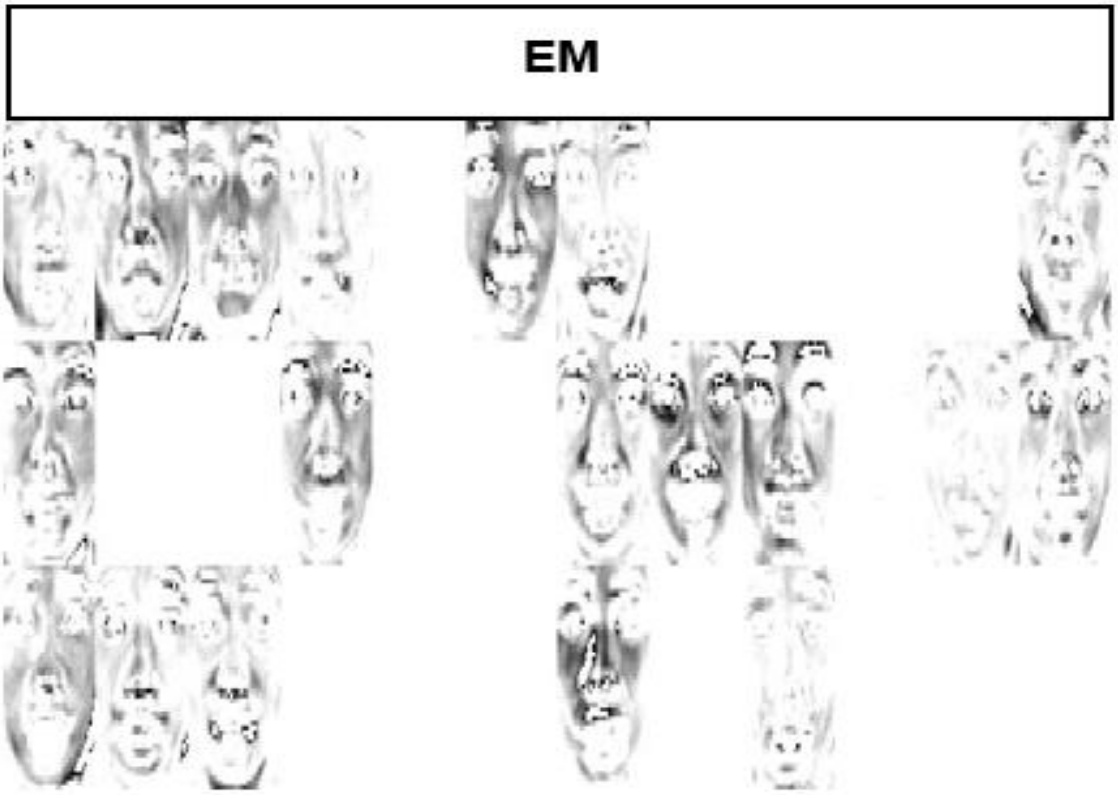

Figure 34. Ground Truth Bases Discovery via EM Based ARD for Jaffe

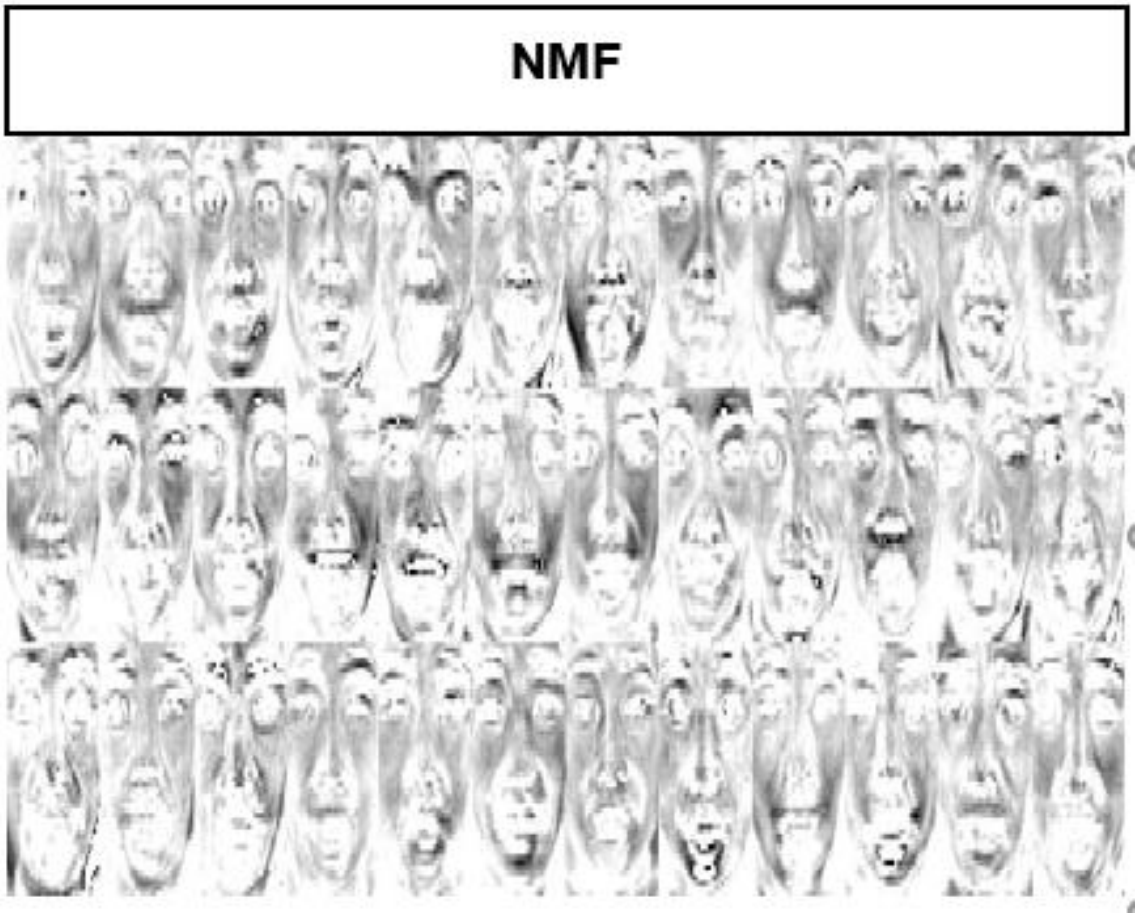

Figure 35. Basic Images Discovery via NMF for Jaffe

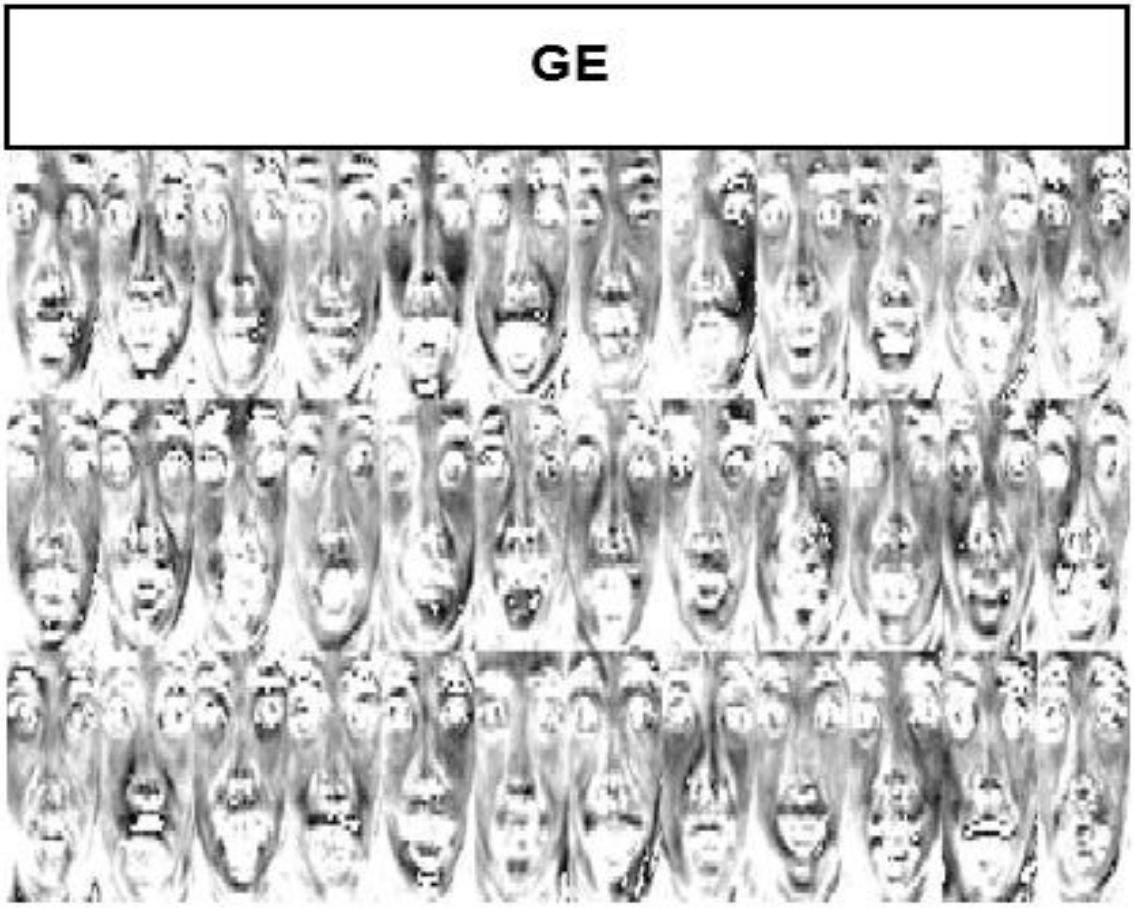

Figure 36. Basic Images Discovery via NMF with GE for Jaffe

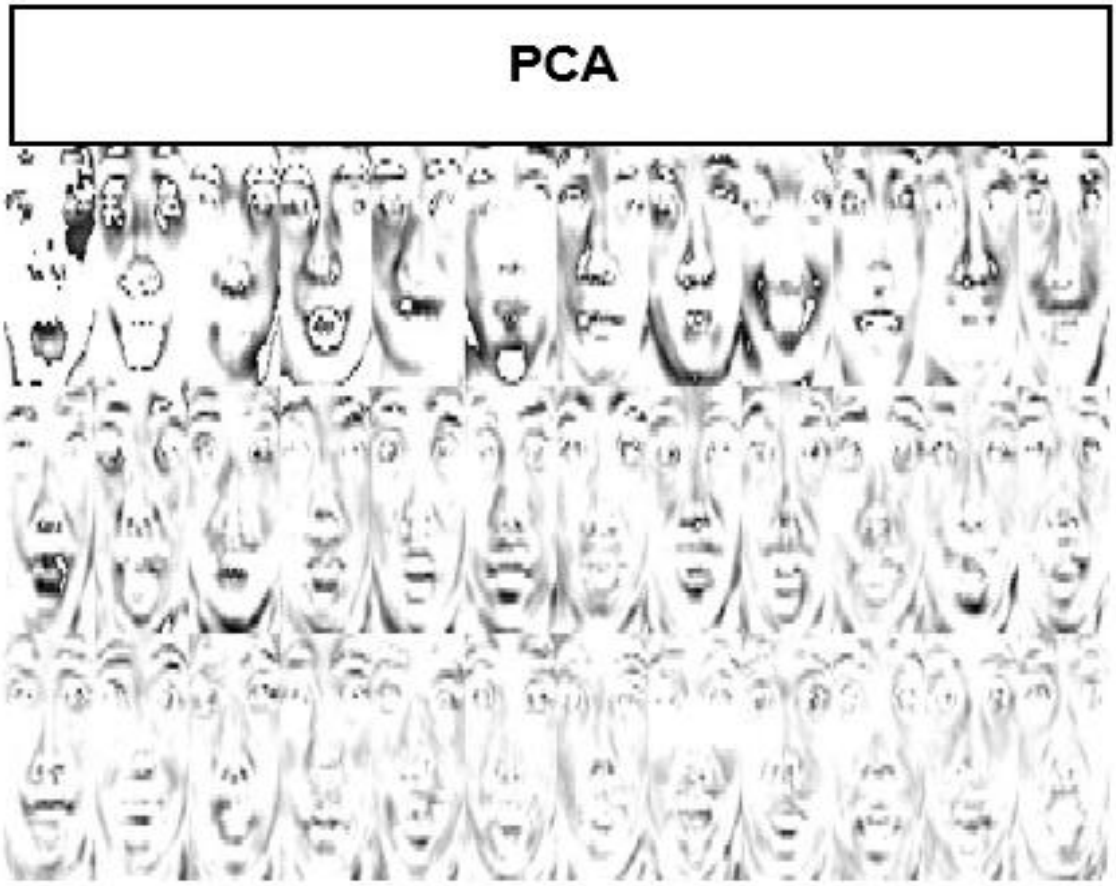

Figure 37. Basic Images Discovery via PCA for Jaffe

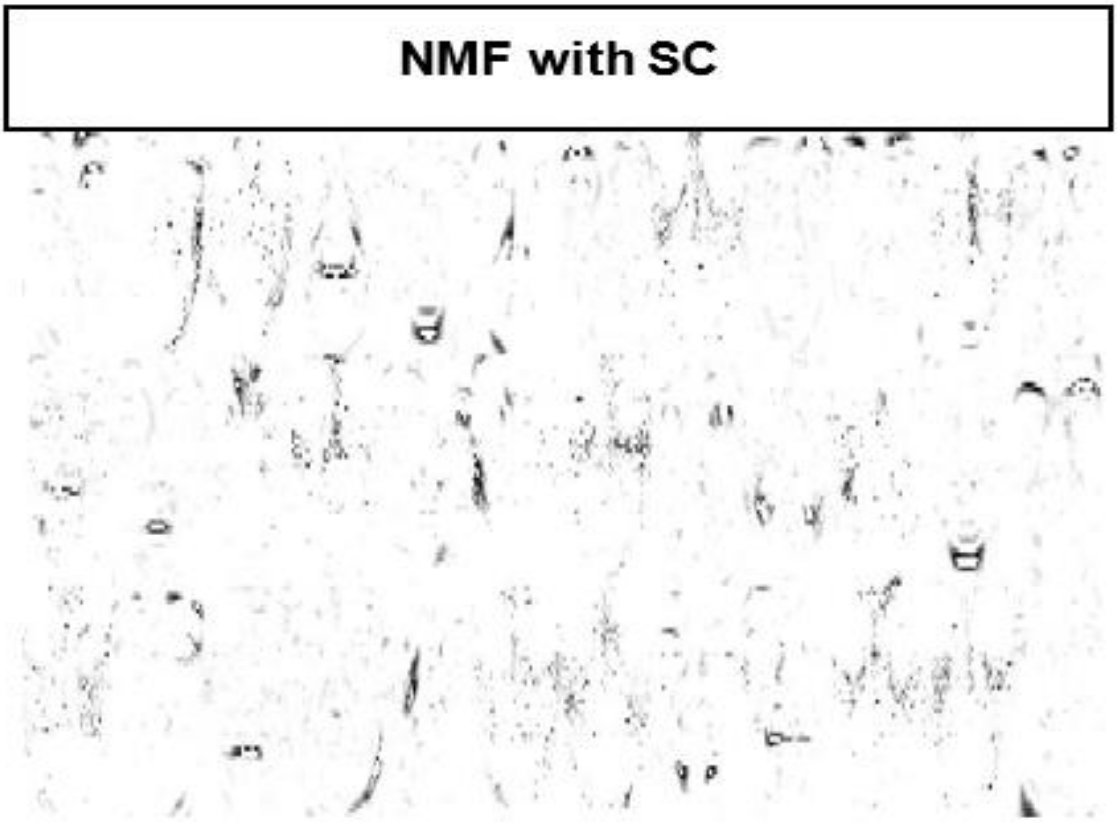

Figure 38. Basic Images Discovery via NMF with SC for Jaffe

The Extended Yale faces dataset is similar with ORL faces, and Jaffe faces, but the size is bigger than others. It's more than 1000 face images, and we also set up the experiment like the others. The result shows that there are 73 ground truth bases with variation of 9 components. Figure 39 is to illustrate the ground truth bases of extended Yale faces

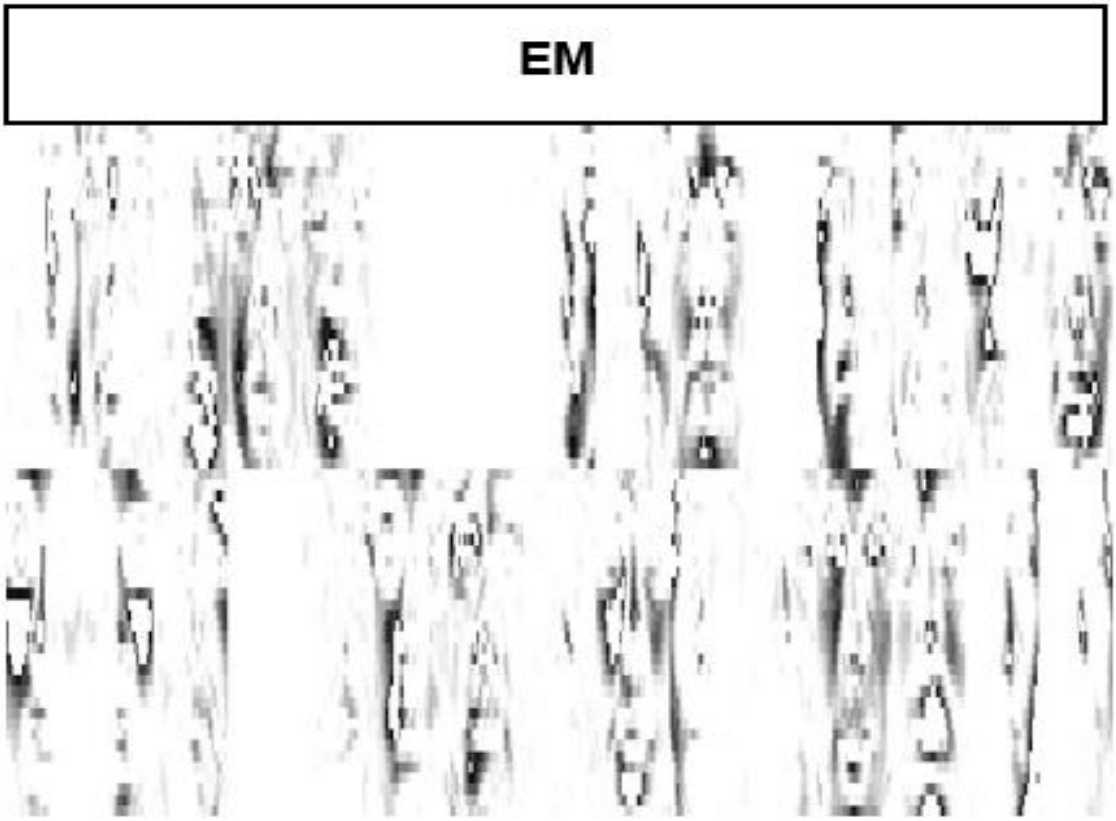

Figure 39. Ground Truth Bases Discovery via EM for Extended Yale

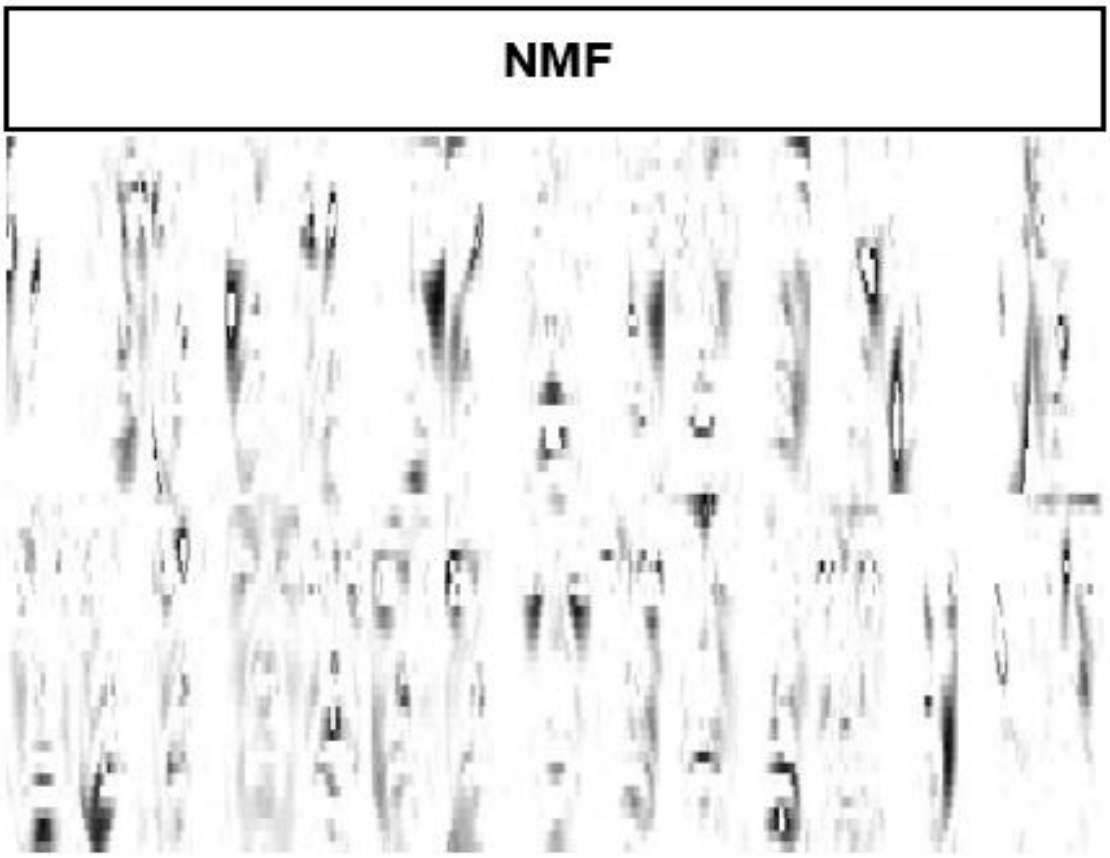

Figure 40. Basic Images Discovery via NMF for Extended Yale

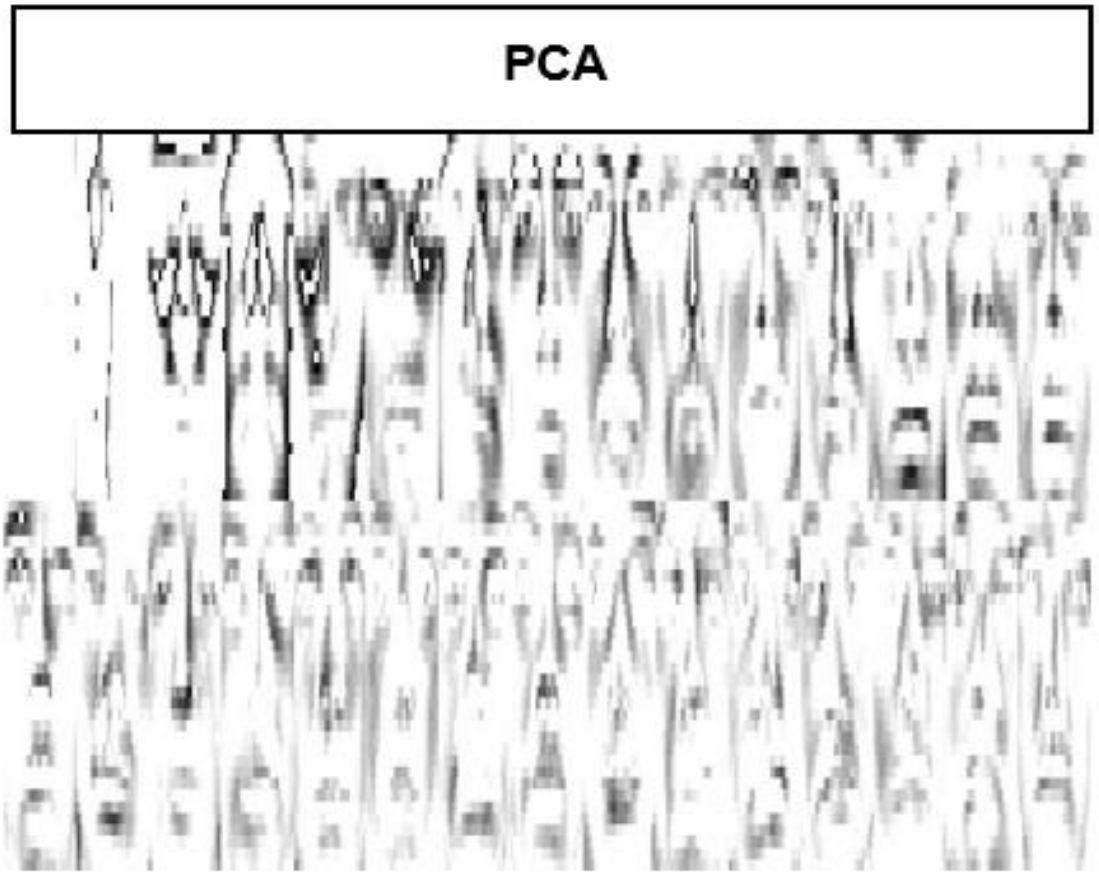

Figure 41. Basic Images Discovery via PCA for Extended Yale

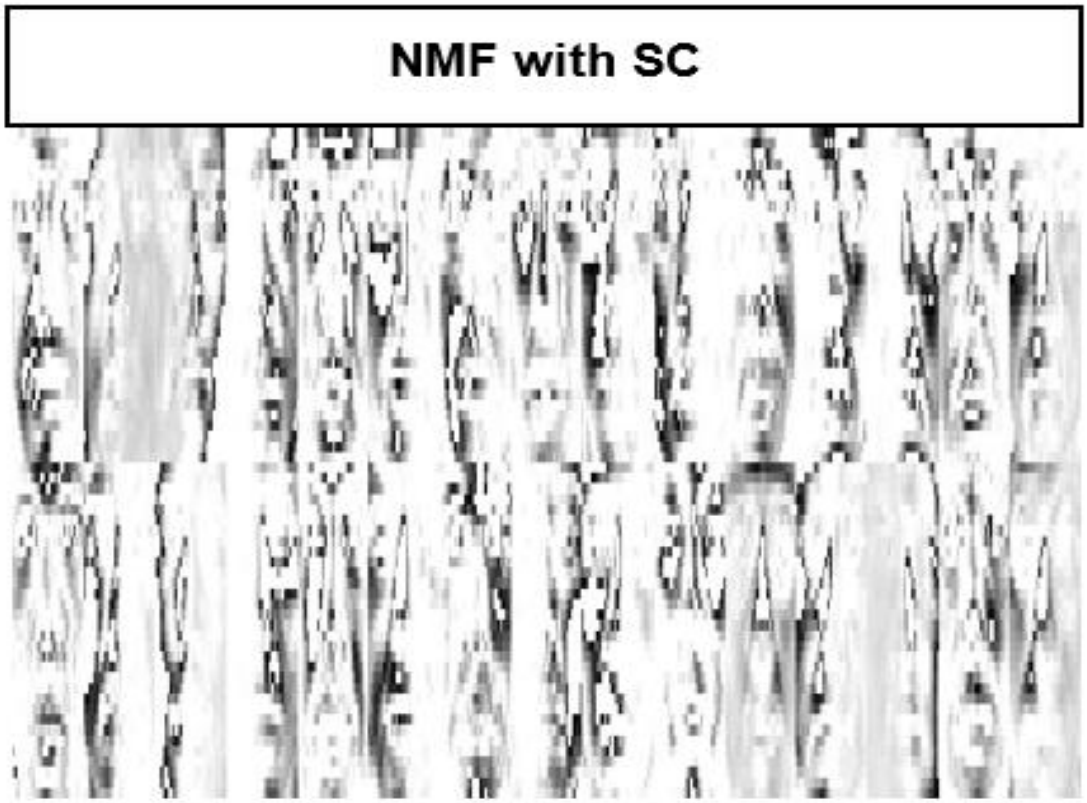

Figure 42. Basic Images Discovery via NMF with SC for Extended Yale

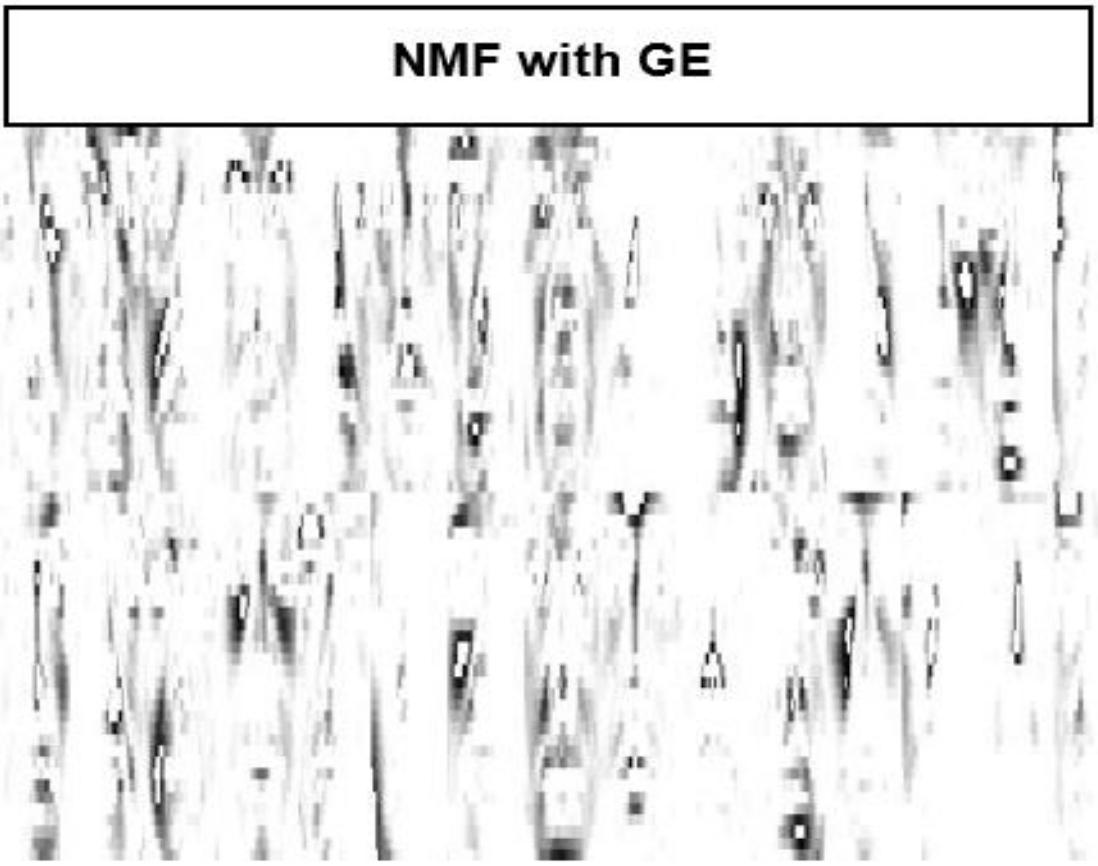

Figure 43. Basic Images Discovery via NMF with GE for Extended Yale

We are doing the experiments on both ORL and Jaffe faces, and Yale datasets, and all of them can recover the ground truth bases in the range form [64-73]. It is reasonable with the physical images because faces have same basic components whatever the datasets are. Therefore, our EM can recover the correct number of basic components from datasets. Table 1 shows the summarization of ground truth bases discovery via EM method over different datasets.

Table 1. Ground Truth Bases Discovery of Fence, Swimmer, ORL, Jaffe

Datasets	Model Order K	Iteration	Number of simulation	Ground truth bases discovery
Fence	18	1000	50	**8 (±1)**
Swimmer	25	1000	50	**16 (±1)**
ORL	100,121,144	4000	50	**62 (±7)**
Jaffe	100,121,144	4000	50	**64 (±3)**
Extended Yale	100,121,144	4000	50	**73 (±9)**

In addition, we also define the optimal model order by calculating the L_2 norm of bases for Fence, Swimmer, ORL, and Jaffe, and extended Yale. Apparently, with EM algorithm, the L_2 norm graph shows that the ground truth bases (red circles) have more energies than the others (empty circles), and the number of red circles also equal to the ground truth bases we already discover from these subplots mentioned above. Figure 44 to Figure 48 illustrates the L_2 norm that discovers the ground truth bases from different datasets

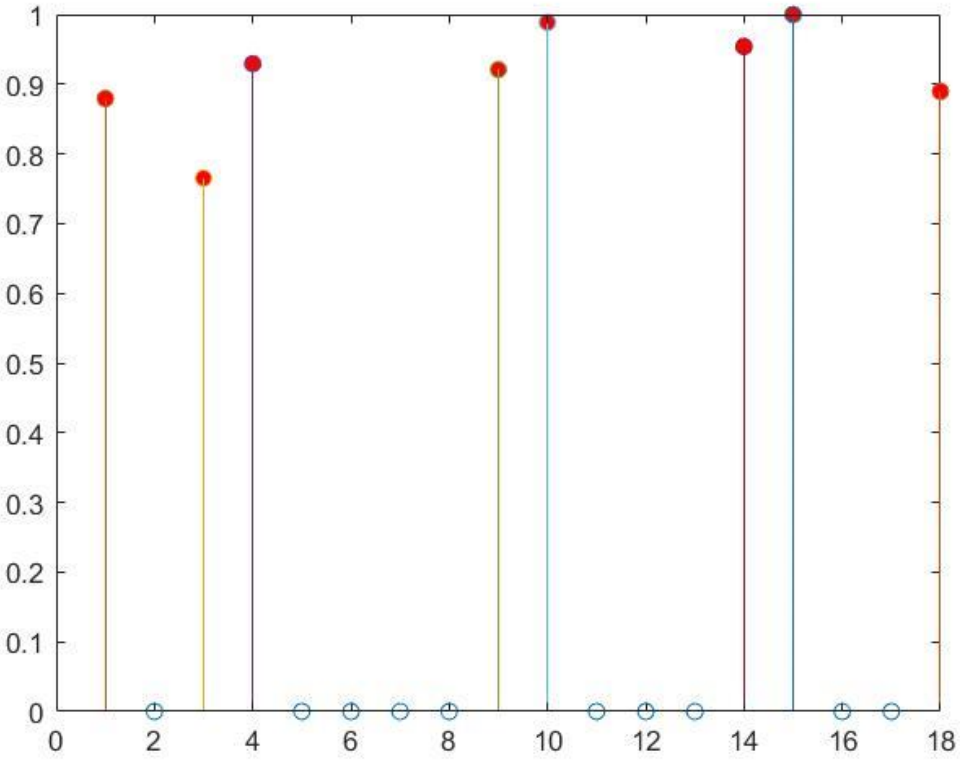

Figure 44. L_2 Norm is to Discover 8 Ground Truth Bases for Fence Dataset

On Figure 44, We set the initial value K =18, and the EM discover 8 ground truth bases (4 horizontal bars, and 4 vertical bars). Apparently, the 8 ground truth bases have positive values (red circles) which are greater than less important components (empty circles) that have values around zero.

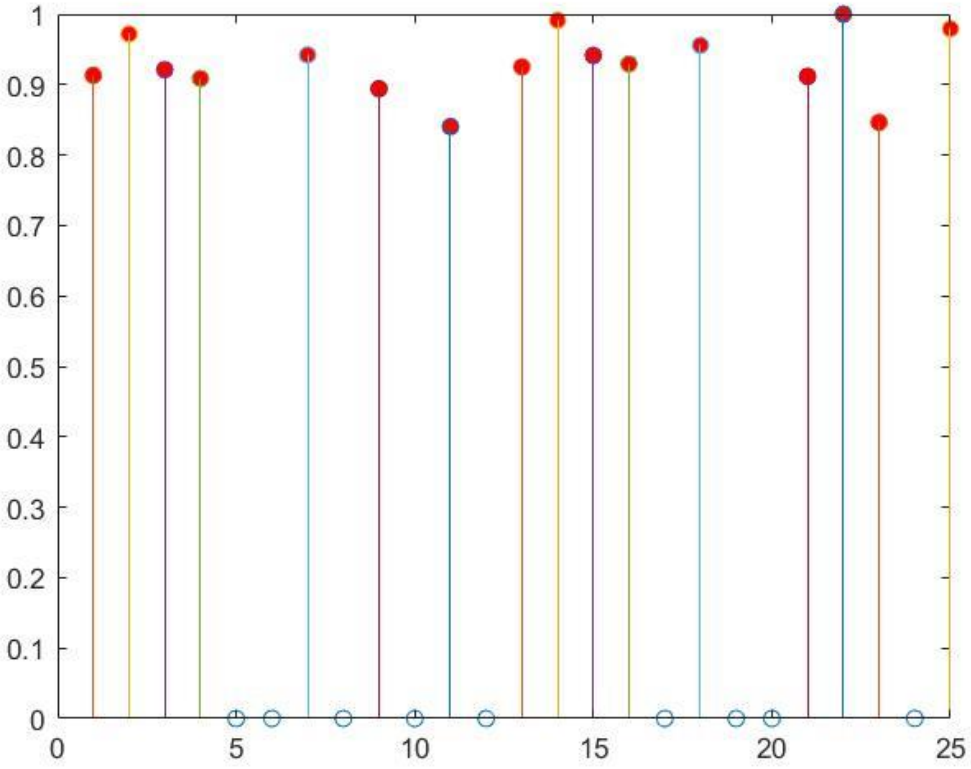

Figure 45. L_2 Norm Discovers 16 Ground Truth Bases of Swimmer Dataset

On Figure 45, We set the initial value K = 25, and the EM discover 16 different limb positions ground truth bases . Apparently, the ground truth bases have positive values (red circles) which are greater than less important components (empty circles) that have values around zero. In this experiment, we choose the threshold value 0.095 to choose basic components. It means that any component is greater than the threshold value = 0.095 we will consider them as the ground truth bases, and skip the zero values.

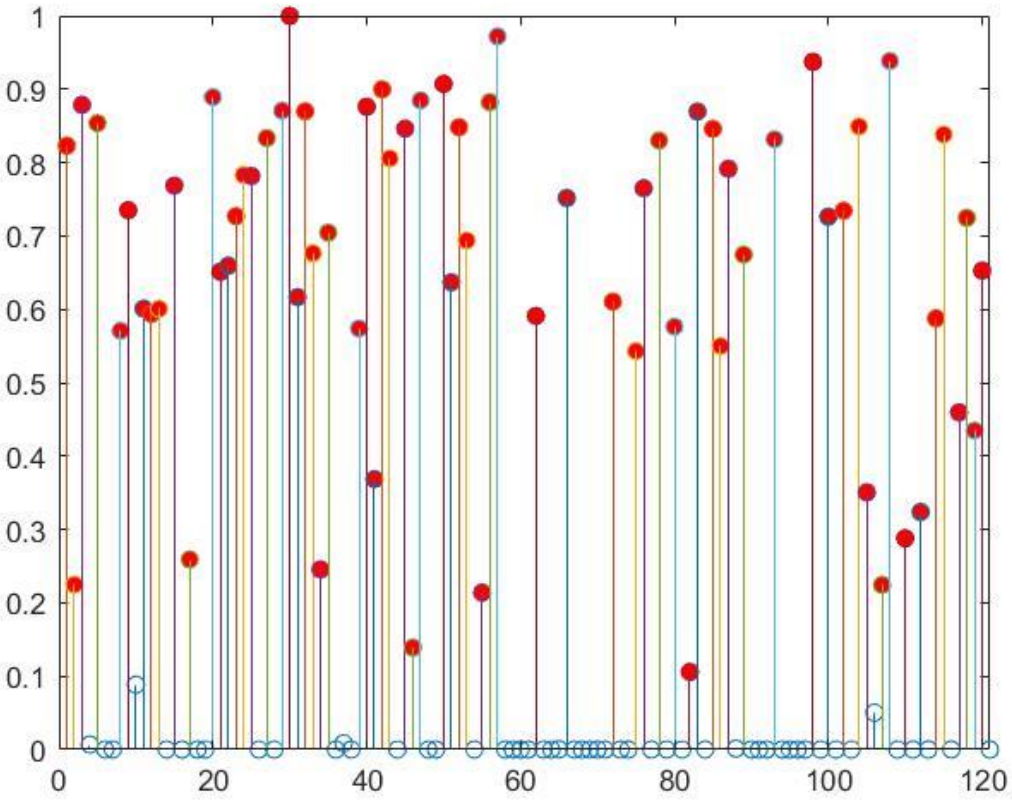

Figure 46. L_2 Norm is to Discover 62 Ground Truth Bases for ORL

On Figure 46, We set the initial value K = 120, and the EM discover 62 ground truth bases (eyes, lips, eyebrows, nose,..,etc) . Apparently, the ground truth bases have positive values (red circles) which are greater than less important components (empty circles) that have values around zero. In this experiment, we choose the threshold value 0.15 to choose basic components. It means that any component is greater than the threshold value = 0.15 we will consider them as the ground truth bases, and skip the values which is less than

the threshold value, and totally, we can discover around 63 basic components for this dataset.

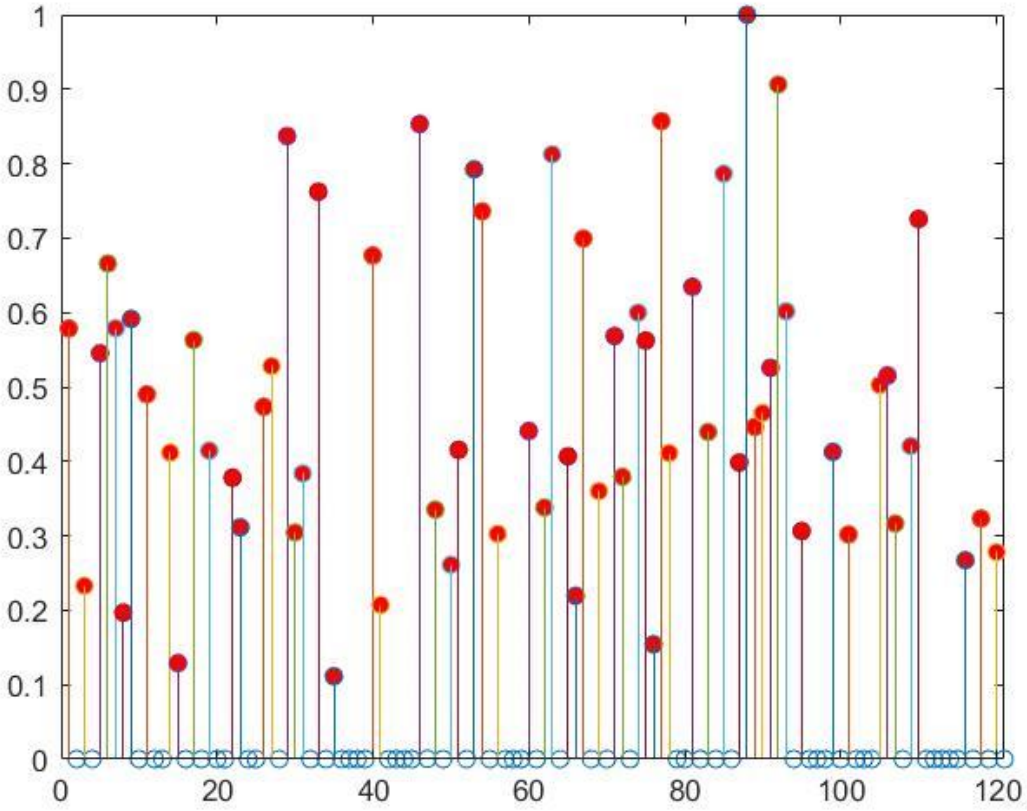

Figure 47. L_2 Norm is to Discover 64 Ground Truth Bases for Jaffe Dataset

On Figure 47, We set the initial value K = 120, and the EM discover 64 ground truth bases (eyes, lips, eyebrows, nose,..,etc) . In this experiment, we choose the threshold value 0.15 to choose basic components. It means that any component is greater than the threshold value = 0.15 we will consider them as

the ground truth bases, and skip the values which is less than the threshold value, and totally, we can discover around 64 basic components for this dataset.

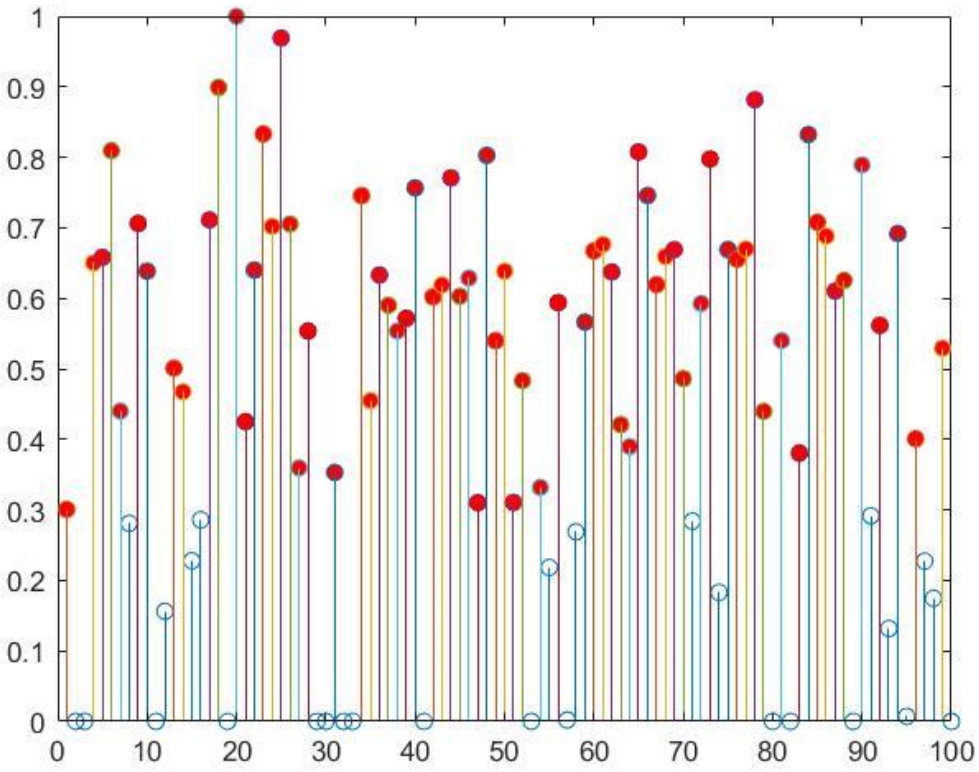

Figure 48. L_2 Norm is to Discover 73 Ground Truth Bases for Yale Dataset

On Figure 48, We set the initial value K = 100, and the EM discover 73 ground truth bases (eyes, lips, eyebrows, nose,..,etc) . In this experiment, we choose the threshold value 0.3 such that we easily recognize the basic components on subplots mentioned above.

Recognition Accuracy Comparison to Unsupervised , and Supervised Learning

EM vs. Principal Component Analysis (PCA), NMF, Linear Discriminant Analysis (LDA), We also compare our proposed method with others (NMF, PCA, LDA) to see how EM can improve the recognition accuracy. In the experiment, we integrate our algorithm EM into specific datasets, and measure the accuracy of the coefficient matrix H in training set over coefficient Matrix H in test set. We will set the different of initial bases for different datasets and will get the best result with respect the number of bases. For example, we set the number of bases for ORL datasets K=36,49,61,81,100, and then we observed that the number of bases K=100 will get the best result as the best recognition accuracy. From the Table 2, we can see that all algorithms are working well on real datasets(ORL, and Jaffe) than synthetic datasets (swimmer, and fence). In any cases, our proposed algorithm EM has better performance than others based on the recognition accuracy rate.

Table 2. Recognition Accuracy Rate for Fence (K=16); Swimmer(K=35), ORL Faces (K=100), and Jaffe (K=100) via EM; NMF, PCA, and LDA

	Datasets			
Algorithms	Swimmer	ORL Faces	Jaffe Faces	Yale B
EM	91.34(±0.85)	99.5 (±0.9)	94.5 (±0.42)	87.17 (±0.02)
NMF	79.32 (±1.24)	96.21 (±0.39)	92.5 (±0.36)	84.25(±0.25)
PCA	75.02(±0.48)	85.16 (±0.16)	87.95(±0.7)	82.03 (±0.18)
LDA	75.59(±0.57)	88.75(±0.14)	89.50(±0.3)	82.98 (±0.5)

Based on the recognition accuracy rate on each dimensionality reduction, we can obtain the bar graph for comparison between EM and others. Figure 49 to Figure 52 illustrate the comparisons of EM to others (NMF,PCA, LDA). In the swimmer dataset, EM has dramatically greater values than others, and the maximum recognition rate can get up to 90%. In the ORL datasets, the recognition rates between different algorithms are similar, but the EM also had slightly higher recognition rate compared to others, and the recognition rate can get up to 95 % at 100^{th} dimensionality. In the Jaffe dataset, the EM has dramatically higher recognition rate at 36^{th}, and 49^{th} dimensionality, but only slightly higher than others when the dimensionality goes up to 100. At this point, and recognition rate is almost 100 %. In the Yale faces dataset, it is more complicated than others (ORL and Jaffe), so the recognition rate is not as high as others, but the EM recognition rate is still higher than other algorithms

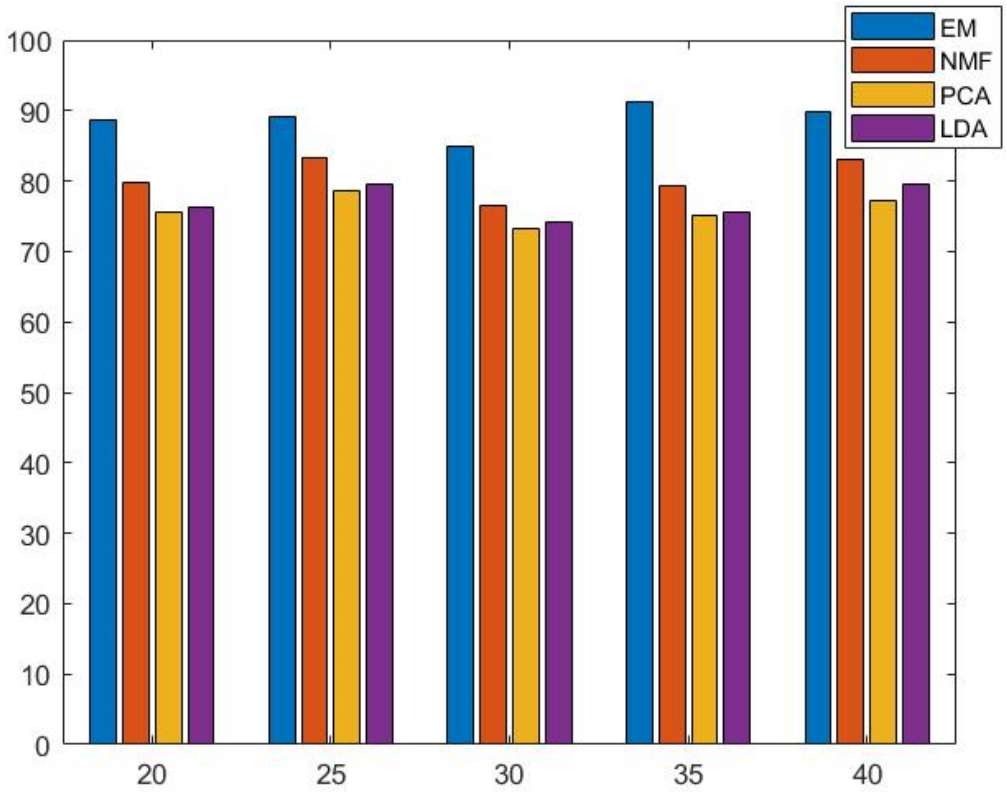

Figure 49. Recognition Accuracy Comparison of EM to Others for Swimmer

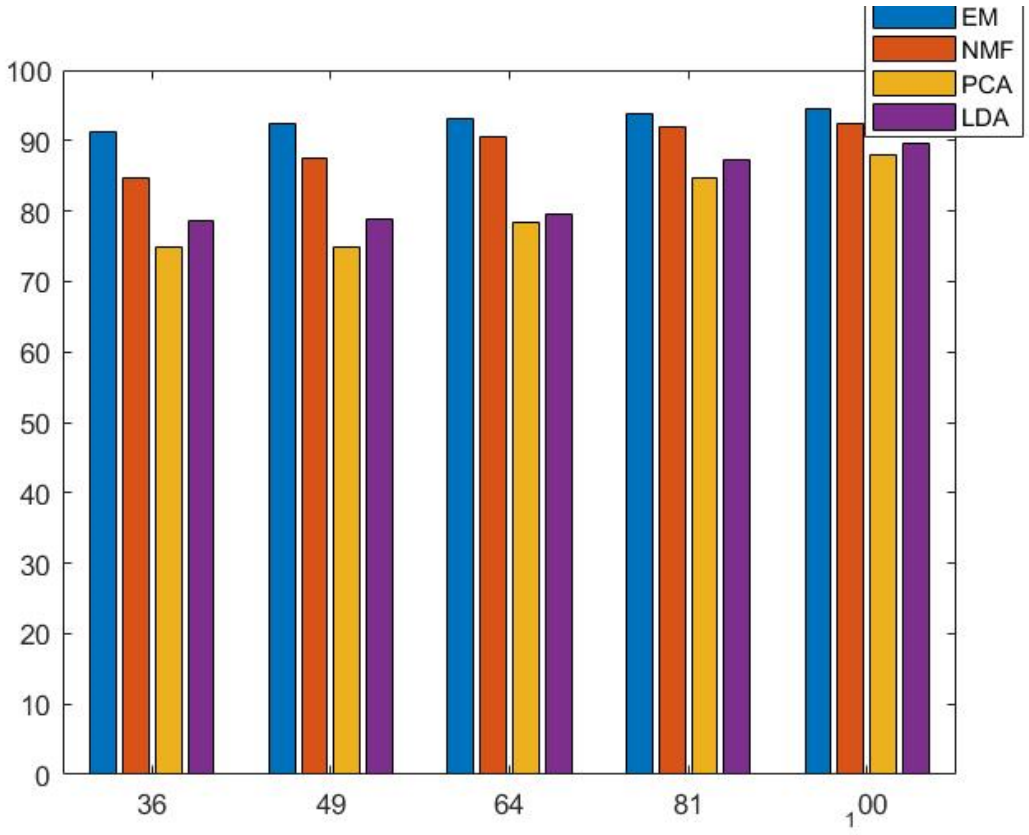

Figure 50. Recognition Accuracy Comparison of EM to Others for ORL

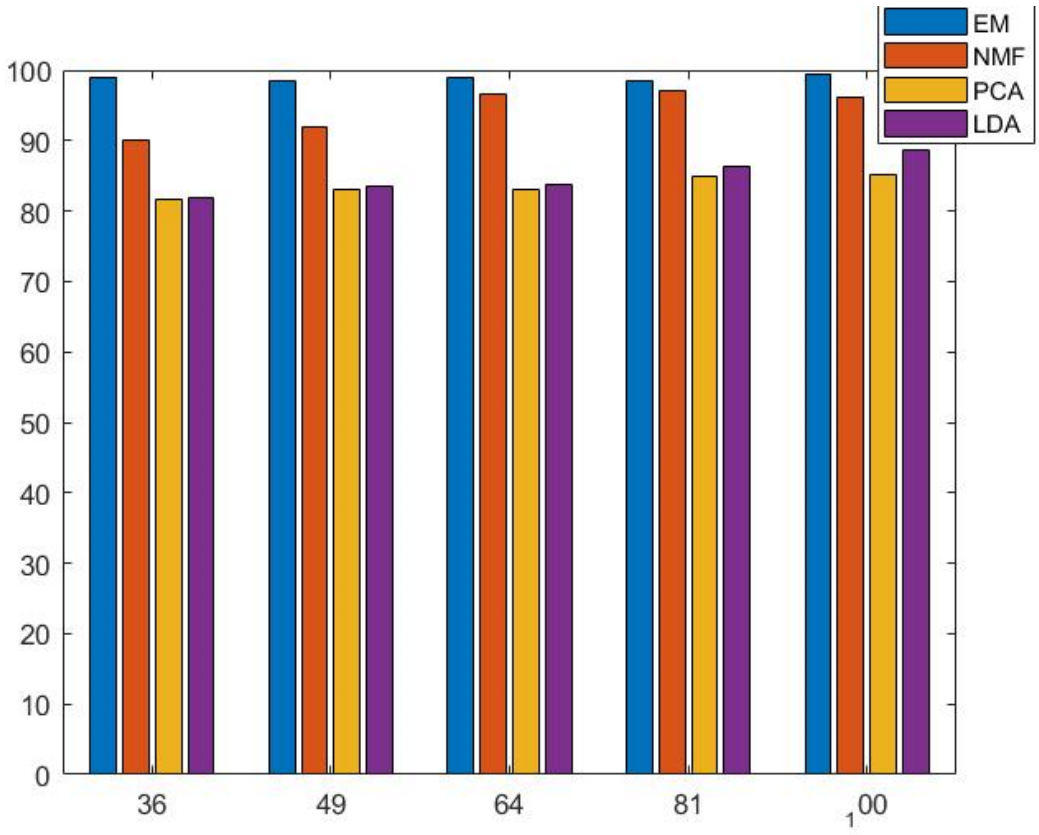

Figure 51. Recognition Accuracy Comparison of EM to Others for Jaffe

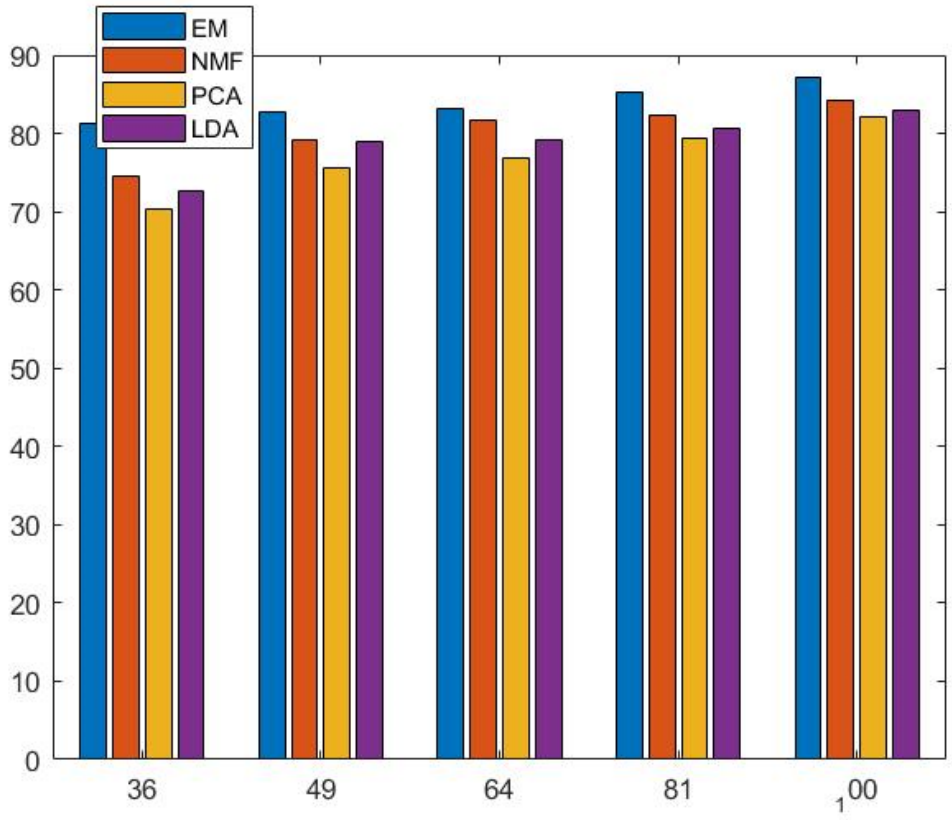

Figure 52. Recognition Accuracy Comparison of EM to Others for Yale

Recognition Accuracy Comparison to Sparsity Based , and Graph Embedding

In this section, we will compare EM to other algorithms such as Sparsity (NMF with SC) based on the Euclidean distance (I) [5] with the sparseness constraint on W, or H, and Kullback-Leibler distance (II) [40]; and graph embedding (I) [17] (II) [18]. In the swimmer dataset, the EM recognition rate achieves the maximum value at 35^{th} dimensionality which is pretty higher than others, but it just gets slightly higher than others when the dimensionality becomes bigger. For the ORL, Jaffe, and Yale datasets, although the EM has higher recognition rate than others, but it is just slightly higher. The most recognition rate can be observed in ORL faces with almost 100 %

Table 3. Comparison EM to Sparsity Based, and Graph Embedding

	Datasets			
Algorithms	Swimmer	ORL Faces	Jaffe Faces	Yale B
EM	91.34(±0.85)	99.5 (±0.9)	94.5 (±0.42)	87.17 (±0.02)
NMF	79.32 (±1.24)	96.21 (±0.39)	92.5 (±0.36)	84.25(±0.25)
NMF_SC (I)	81.26(±1.46)	98.16 (±0.16)	93.67(±0.7)	85.26 (±0.18)
NMF_SC(II)	80.17(±0.93)	97.75(±0.14)	93.13 (±0.3)	84.57 (±0.5)
GE(I)	80.65(±0.62)	96.63 (±0.67)	91.29(±0.63)	86.11 (±0.41)
GE(II)	81.76(±0.25)	97.82(±0.91)	92.68(±0.13)	86.78 (±0.53)

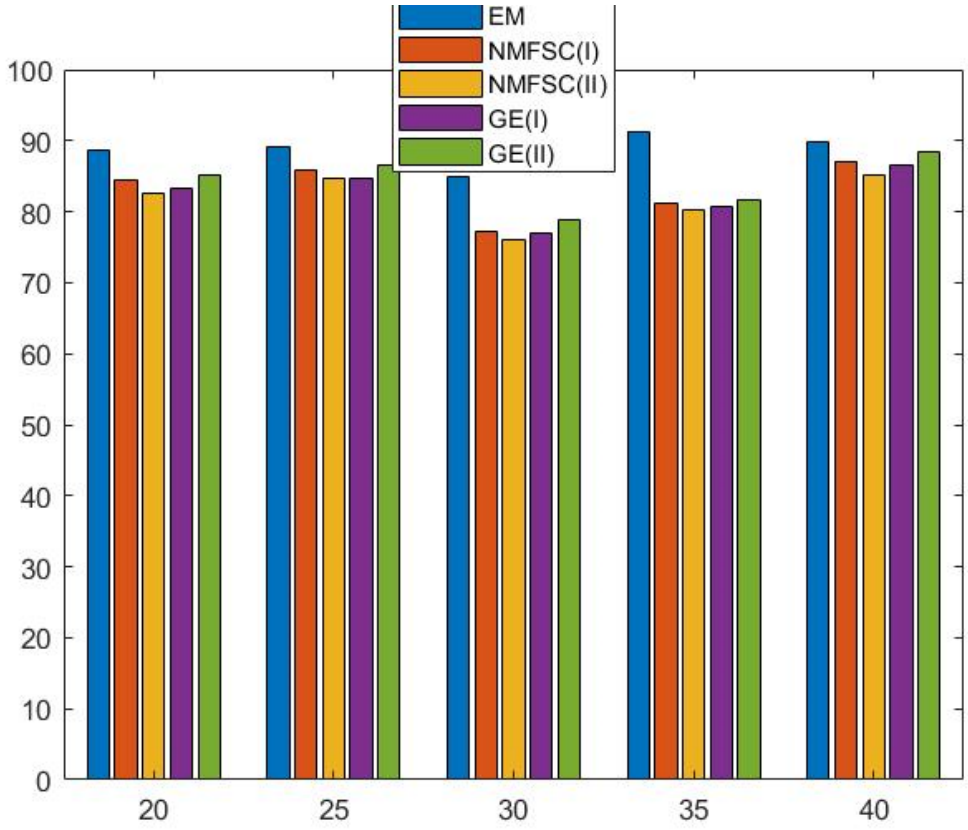

Figure 53. Comparison of EM to Others for Swimmer Dataset

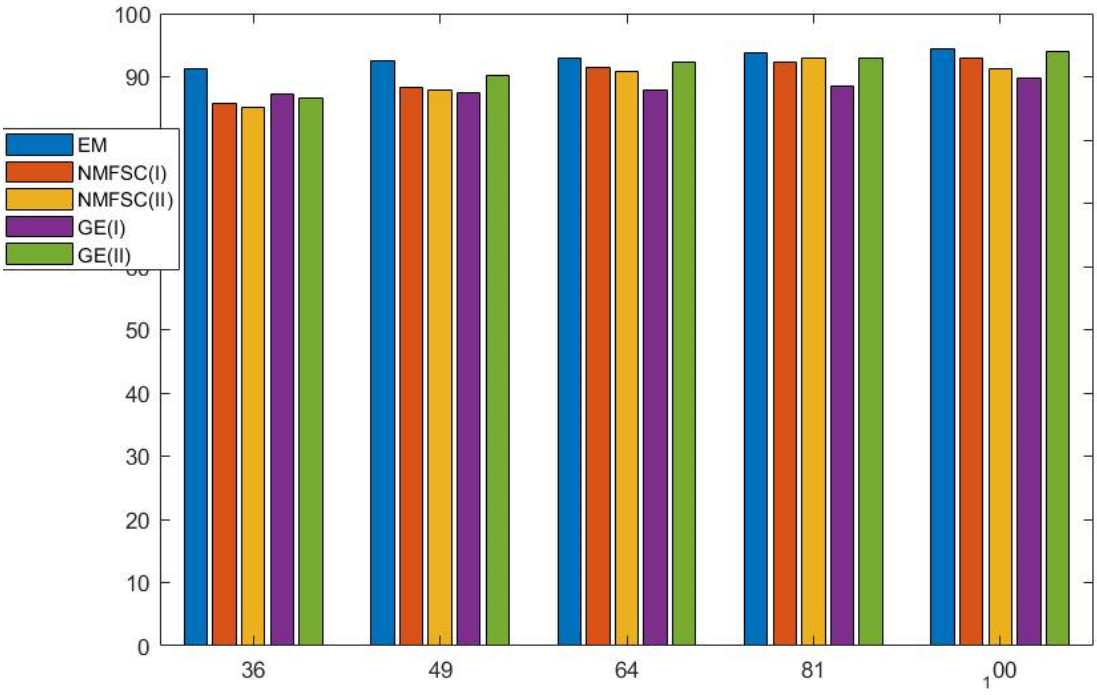

Figure 54.Comparison of EM to Others for ORL Dataset

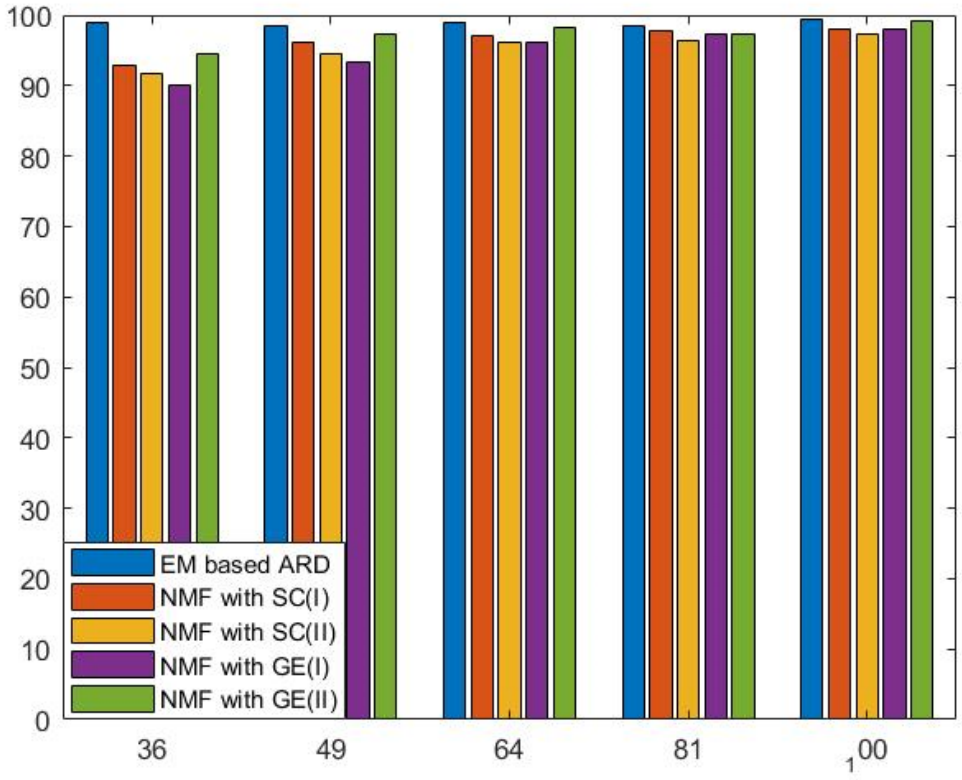

Figure 55. Comparison of EM to Others for Jaffe Dataset

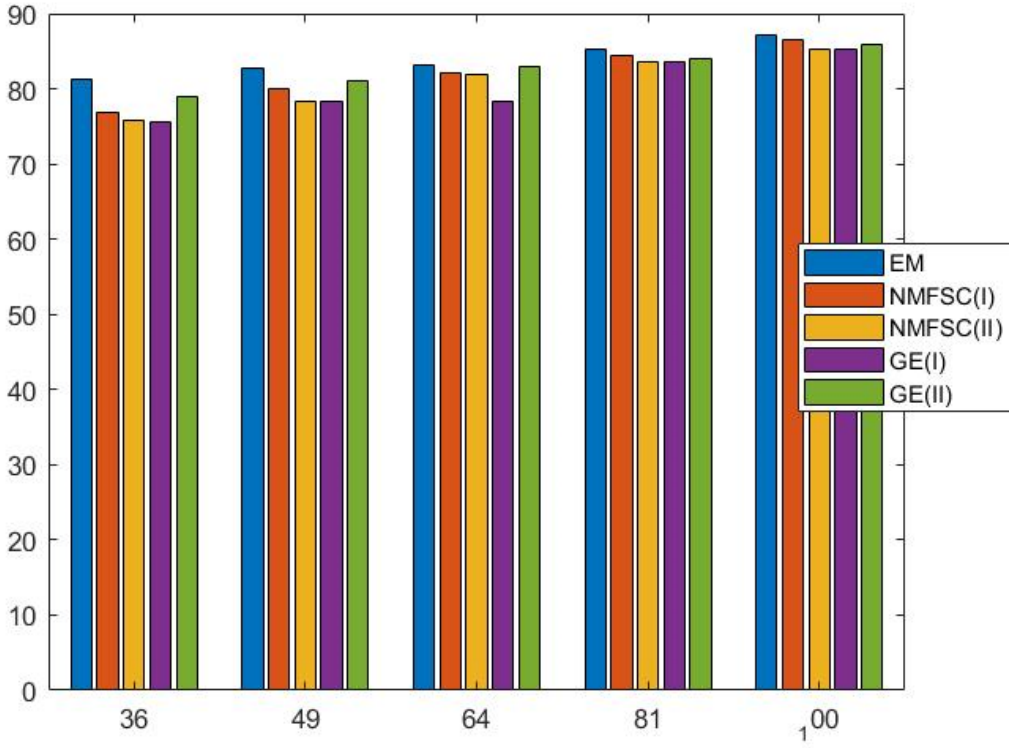

Figure 56. Comparison of EM to Others for Yale Dataset

CHAPTER SEVEN

CONCLUSION AND FUTURE WORKS

Conclusion

In conclusion, our proposed algorithm has successfully discovered the ground truth bases as well as the model order K in the different datasets from simple ones such as: swimmer, and fence to complicated ones: ORL, Jaffe, and Yale faces datasets. In addition, the EM algorithm with ARD can achieve the higher recognition rate than other algorithms such as: NMF, LDA, PCA, Sparsity based, and graph embedding. Therefore, our new algorithm has achieved 2 goals: ground truth bases extraction, and improve the recognition rate.

Future Works

Our EM can discover the ground truth bases from the dataset, it is easily the recognized these ground truth bases on simple datasets such as swimmer and fence. We can count and visualize clearly the unique patterns in fence (4 horizontal bars and 4 vertical bars), and swimmer (16 limb positions, and one static torso). However, for the complexity dataset such as: ORL, Jaffe, and extended Yale faces datasets, it discovers the faces with mix components, and it is hard to visualize what the ground truth bases are. In the future, we can find out

the way to integrate the sparse coding to the EM so that it can both discover the ground truth bases and easily to recognize them on subplots.

REFERENCES

[1] C. M. Bishop, *Pattern recognition and machine learning*. New York: Springer, 2006.

[2] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[3] Chih-Jen Lin, “On the Convergence of Multiplicative Update Algorithms for Nonnegative Matrix Factorization,” *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1589–1596, Nov. 2007.

[4] S. Wold, K. Esbensen, and P. Geladi, “Principal Component Analysis,” p. 16.

[5] P. O. Hoyer and P. Hoyer, “Non-negative Matrix Factorization with Sparseness Constraints,” p. 13.

[6] X. Liu, S. Yan, and H. Jin, “Projective Nonnegative Graph Embedding,” *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1126–1137, May 2010.

[7] Q. Sun, J. Lu, Y. Wu, H. Qiao, X. Huang, and F. Hu, “Non-informative hierarchical Bayesian inference for non-negative matrix factorization,” *Signal Process.*, vol. 108, pp. 309–321, Mar. 2015.

[8] D. Donoho and V. Stodden, “When Does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts?,” in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. K. Saul, and B. Schölkopf, Eds. MIT Press, 2004, pp. 1141–1148.

[9] A. T. Cemgil, “Bayesian Inference for Nonnegative Matrix Factorisation Models,” *Computational Intelligence and Neuroscience*, 2009. [Online]. Available: https://www.hindawi.com/journals/cin/2009/785152/. [Accessed: 29-Sep-2018].

[10] V. Y. F. Tan and C. Févotte, “Automatic Relevance Determination in Nonnegative Matrix Factorization,” p. 6.

[11] M. A. T. Figueiredo, “Adaptive sparseness for supervised learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1150–1159, Sep. 2003.

[12] I. D. Gebru, X. Alameda-Pineda, F. Forbes, and R. Horaud, “EM Algorithms for Weighted-Data Clustering with Application to Audio-Visual Scene Analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 12, pp. 2402–2415, Dec. 2016.

[13] J. Karvonen and A. Kaarna, “Sea Ice SAR Feature Extraction by Non-Negative Matrix and Tensor Factorization,” in *IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium*, Boston, MA, USA, 2008, pp. IV-1093-IV–1096.

[14] J. Eggert and E. Korner, “Sparse coding and NMF,” in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, Budapest, Hungary, 2004, vol. 4, pp. 2529–2533.

[15] H. Kim and H. Park, “Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis,” *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, Jun. 2007.

[16] Hanwang Zhang, Zheng-Jun Zha, Shuicheng Yan, Meng Wang, and Tat-Seng Chua, “Robust Non-negative Graph Embedding: Towards noisy data, unreliable graphs, and noisy labels,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, 2012, pp. 2464–2471.

[17] Deng Cai, Xiaofei He, Jiawei Han, and T. S. Huang, “Graph Regularized Nonnegative Matrix Factorization for Data Representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[18] C. Lin and M. Pang, “Graph Regularized Nonnegative Matrix Factorization with Sparse Coding,” *Math. Probl. Eng.*, vol. 2015, Mar. 2015.

[19] D. D. Lee and H. S. Seung, “Unsupervised Learning by Convex and Conic Coding,” in *Advances in Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. MIT Press, 1997, pp. 515–521.

[20] N. Gillis, “The Why and How of Nonnegative Matrix Factorization,” *undefined*, 2014. [Online]. Available: /paper/The-Why-and-How-of-Nonnegative-Matrix-Factorization-Gillis/634ca7583f9ee4e0c68c9ce278e10227e2c4e819. [Accessed: 23-Oct-2018].

[21] J. Zheng, H. Zhang, C. Cattani, and W. Wang, “Dimensionality Reduction by Supervised Neighbor Embedding Using Laplacian Search,” *Computational and Mathematical Methods in Medicine*, 2014. [Online]. Available: https://www.hindawi.com/journals/cmmm/2014/594379/. [Accessed: 30-Sep-2018].

[22] D. D. Lee and H. S. Seung, “Algorithms for Non-negative Matrix Factorization,” in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2001, pp. 556–562.

[23] P. Paatero and U. Tapper, “Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values,” *Environmetrics*, vol. 5, no. 2, pp. 111–126, Jun. 1994.

[24] K. Devarajan, “Nonnegative Matrix Factorization: An Analytical and Interpretive Tool in Computational Biology,” *PLoS Comput. Biol.*, vol. 4, no. 6, pp. 1–12, Jun. 2008.

[25] Y. Zhou, M. Kantarcioglu, and B. Thuraisingham, “Sparse Bayesian Adversarial Learning Using Relevance Vector Machine Ensembles,” in *2012 IEEE 12th International Conference on Data Mining*, 2012, pp. 1206–1211.

[26] D. J. C. Mackay, “Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks,” *Netw. Comput. Neural Syst.*, vol. 6, no. 3, pp. 469–505, Jan. 1995.

[27] Q. Sun, H. Tao, “Robust Hierarchical Bayesian Modeling for Automatic Model Order Determination”, *Advances in Signal Processing: Reviews' Book Series*. Chapter 2, pp. 41-59. ISBN: 978-84-09-04329-3

[28] F. Gu, H. Zhang, W. Wang, and S. Wang, “An Expectation-Maximization Algorithm for Blind Separation of Noisy Mixtures Using Gaussian Mixture Model,” *Circuits Syst. Signal Process.*, vol. 36, no. 7, pp. 2697–2726, Jul. 2017.

[29] S.-C. Lo, “Expectation-maximization based algorithm for pattern recognition in traffic speed distribution,” *Math. Comput. Model.*, vol. 58, no. 1–2, pp. 449–456, Jul. 2013.

[30] D. J. Field, “What Is the Goal of Sensory Coding?,” *Neural Comput.*, vol. 6, no. 4, pp. 559–601, Jul. 1994.

[31] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, “Graph Embedding and Extensions: A General Framework for Dimensionality Reduction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[32] Jianchao Yang, Shuicheng Yang, Yun Fu, Xuelong Li, and T. Huang, “Non-negative graph embedding,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008, pp. 1–8.

[33] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, “Coding facial expressions with Gabor wavelets,” in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 200–205.

[34] “Facial Expression Database: Japanese Female Facial Expression (JAFFE) Database.” [Online]. Available: http://www.kasrl.org/jaffe.html. [Accessed: 30-Sep-2018].

[35] Y. Tu and C. Hsu, “Dual subspace nonnegative matrix factorization for person-invariant facial expression recognition,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012, pp. 2391–2394.

[36] “Yale Face Database.” [Online]. Available: http://vision.ucsd.edu/~leekc/ExtYaleDatabase/Yale%20Face%20Database.htm. [Accessed: 28-Oct-2018].

[37] “Popular Face Data Sets in Matlab Format.” [Online]. Available: http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html. [Accessed: 28-Oct-2018].

[38] “Advances in Neural Information Processing Systems | The MIT Press.” [Online]. Available: https://mitpress.mit.edu/books/advances-neural-information-processing-systems. [Accessed: 08-Oct-2018].

[39] V. Haltakov, C. Unger, and S. Ilic, “Framework for Generation of Synthetic Ground Truth Data for Driver Assistance Applications,” in *Pattern Recognition*, vol. 8142, J. Weickert, M. Hein, and B. Schiele, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 323–332.

[40] T. Virtanen, “Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria,” *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 3, pp. 1066–1074, Mar. 2007.