EMBEDDED SPARSE REPRESENTATION FOR IMAGE CLASSIFICATION

by

JIN HUANG

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2010

To my parents who support me from overseas

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Heng Huang for constantly motivating and encouraging me, and also for his invaluable advice during the course of my master studies and financially support me. I wish to thank my academic advisors Dr. Chris Ding, Dr. Jeff Lei for their interest in my research and for taking time to serve in my thesis committee.

I would also like to extend my appreciation to Dr Feiping Nie, whose idea motivated this thesis and his contribution to this thesis

I am grateful to all the teachers who taught me during the years I spent in school, first in Palestine, then in and in the Unites States. I would like to thank my previous colleagues for encouraging and inspiring me to pursue graduate studies.

Finally, I would like to express my deep gratitude to my parents who have encouraged and inspired me and sponsored my undergraduate. I also thank several of my friends who have helped me throughout my career.

July 16, 2010

ABSTRACT

EMBEDDED SPARSE REPRESENTATION
FOR IMAGE CLASSIFICATION

Jin Huang , M.S
The University of Texas at Arlington, 2010

Supervising Professor: HENG HUANG

Image classification, such as face recognition and scene categorization, is an important research area in computer vision over the last decade. It has been successfully applied to many image analysis applications. Images usually have a large number of features, hence the dimensionality reduction methods are often employed before the subsequent classification to improve the classification results. A lot of methods have been proposed, including but not limited to PCA, ICA, LDA, and Bayesian Framework.

Recently, compressive sensing and sparse learning have been widely studied and applied into computer vision research. Ma et al. suggested a new method called Sparse Representation Classification (SRC). This new framework provides new insights into two critical issues in image classification: feature extraction and robustness to occlusion. Motivated by this method, we proposed a new method called embedded sparse representation. This masterpiece combined the dimension reduction and classification into one. We proposed three possible objective functions and discussed the possible optimization ways to tackle them. During the optimization, we used Gibbs Optimization method to alternatively find the optimal subspace representation and the sparse weight factor.

In the experiments, we verified the convergence of our method. Our method successfully extracted the subspace structure and got a better performance than SRC and other classical methods. It has also been shown that our method has the potential to be extended to other general high dimensional data. The possible improvement and future work have also been discussed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction to face recognition

1.1.1 Classical methods and application

Face recognition has been a popular research area in computer vision over the last decade. It is considered as one of the most successful applications in image analysis. The goal of face recognition is to identify the object in the scene using a stored database.There are quite a few subspace face recognition algorithms there.

Principal component analysis(PCA):PCA is derived from Karhunen-Loeve's transformation. It is a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.Therefore,given a high dimensional vector representation of each face in a training set of images, PCA tends to find a much lower dimensional subspace whose basis vectors correspond to the maximum variance direction in the original image space. If the image elements are considered as random variables, the PCA basis vectors are defined as eigenvectors of the scatter matrix. There are a few papers related to this, include [5],[6],[7],[8].

Independent Component Analysis (ICA): ICA is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. ICA assumes a generative model for the observed multivariate data,in the model,the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed non-gaussian and mutually independent,and they are called the independent components,that is where the name of the method comes from. The method

minimizes both second-order and higher-order dependencies in the input data and try to find the basis along which the data are statistically independent. Bartlett [9] provided two architectures of ICA for face recognition task: Architecture: 1 statistically independent basis images, 2 factorial code representation. There are also two related articles [10],[11]. ICA looks very similar to PCA, however, it is capable of finding latent factor that PCA might fail.

Linear Discriminant Analysis (LDA): LDA find a linear combination of features which separate two or more classes of objects or events. The resulting combination may be used as a linear classifier or for dimensionality reduction before later classification. For all samples of all classes the between-class scatter matrix s_B and the within-class scatter matrix s_W are defined. The goal is to maximize s_B while minimizing s_W,in other words, maximize the ratio $\frac{s_B}{s_W}$. This ratio is maximized when the column vectors of the projection matrix are the eigenvectors of $s_W^{-1}s_B$, related article include [12], [13].

Bayesian Framework: It is a probabilistic similarity measure based on the belief that the image intensity differences are characteristic of typical variations in appearance of an individual. Two classes of facial image variations are defined: intrapersonal variations and extrapersonal variations[13]. Similarity among faces is measures using Bayesian rule.

All these methods have been applied successfully to some datasets to become the classical methods. However, numerous methods still have been coming out due to the wide applications of face recognition. Face recognition has a wide variety of applications in everyday life, please see table 1.1 from partial of the table in [14],

1.1.2 New development

3D facial recognition: it uses a 3D model, which seems to provide more accuracy. it Captures a real-time 3D image of a person's facial surface, uses the most distinctive features of the face,for example, the rigid tissue and bone,the curves of

Table 1.1. Face recognition applications.

$Biometrics$	$driverlicense, passport$
$InformationSecurity$	$ApplicationSecurity, internetsecurity$
$LawEnforcement$	$Shopliftingtracking$
$AccessControl$	$FacilityAccess$

the eye socket,nose and chin – to identify the subject. These areas are all unique and remains unchanged. Using depth and an axis of measurement that are not affected by illumination, 3D facial recognition can even be used in the dark and has the ability to recognize a subject at different view angles with the potential to recognize up to 90 degrees (a face in profile).Using the 3D software, the system goes through several steps to verify the identity of an individual.

The first is the detection: Acquiring an image can be accomplished by digitally scanning an existing photograph (2D) or by using a video image to acquire a live picture of a subject(3D).

Next is to do the alignment: Once it detects a face, the system determines the head's position, size and pose.

Measurement:The system then measures the curves of the face on a sub-millimeter scale and creates a template.

Representation: the system translates the template into a unique code. This coding gives each template a set of numbers to represent the features on a subjects face.

Matching: if the image is 3D and the database contains 3D images, then matching will take place without any changes being made to the image. However, there is a challenge currently facing databases that are still in 2D images. 3D provides a live, moving variable subject being compared to a flat, stable image.

Verification or Identification: in verification, an image is matched to only one image in the database. For example, an image taken of a subject may be matched to

an image in the Department of Motor Vehicles database to verify the subject is who he says he is. If identification is the goal, then the image is compared to all images in the database resulting in a score for each potential match (1:N). In this instance, you may take an image and compare it to a database of mug shots to identify who the subject is.

In conclusion, image classification has always been an interesting research topic for a long time and has wide applications, the new technique keeps on developing, it is expected new method and technology would make the classification more accurate and faster.

1.2 Related work

In this section, I want to give a brief introduction about the dimension reduction and classification methods we would compare during the experiment part.

For PCA, it is quite well known. Therefore I omit the main stuff. I just want to point out that , for PCA, as the number of dimension increase, the portion of the variance each additional eigenvector explains is decreasing, therefore, for PCA, its performance at very low dimension is very hard to beat due to its nature, however, in order to boost the accuracy, increase the subspace dimension to a very high level will not help. It belongs to unsupervised category since no label information is utilized during the training.

Here comes another dimension reduction method, called locality preserving projection(LPP)[2], compared with PCA, LPP is less well known. So I would introduce the main idea here.

It first builds a graph incorporating neighborhood information of the data set. Using the concept of Laplacian matrix, a transformation matrix which maps the data points to a subspace is computed. This linear transformation optimally preserves the local neighborhood information to some extent. This technique has the following

merits: first, it incoperates the local information; second,it is linear, therefore its speed is fast; Last,it is defined everywhere.

This method does not enclose the label information during the training process, therefore, it also belongs to unsupervised category.

Next part, I want to mention two widely used classification methods, KNN and SVM.

KNN(k-nearest neighborhood) is a method for classifying objects based on closest training examples in the feature space. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor. This method is quite simple, however, it is very sensitive to local information.

Next, I want to mention a little bit about SVM. In 1995, Corinna Cortes and Vladimir Vapnik suggested a modified maximum margin idea that allows for mislabeled examples.If there exists no hyperplane that can split the "yes" and "no" examples, the Soft Margin method will choose a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples.Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. Intuitively, an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest

training datapoints of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

These dimension reduction and classification methods are all very popular. They are combined perfectly for a lot of data sets. For the dimension reduction methods I mentioned, they both belong to the unsupervised category as no label information is encoded during the training. For SVM and KNN, they are both widely used classification methods. In the experiment part, I would compare the performance of these combinations with our methods. Assume we are given a set of images A_i, $A_i = [x_{i,1}, x_{i,2}, ..., x_{i,n_i}] \in R^{m \times n_i}$ from k different classes,reshape them into column vectors and combine into $A = [A_1, ..., A_k] \in R^{m \times n}$ y is the i^{th} column of A, A_{-i} is A without y. we want to minimize, the weight vector in the following equation

$$\min_{\alpha_i} \frac{1}{2}||w^T A_{-i}\alpha_i - w^T x_i||_2^2 + \lambda||\alpha_i||_1 \tag{1.1}$$

Note that the term $\lambda||\alpha_i||_1$ is the regularity term here,λ is the controlling coefficient. Let us explain the purpose of introducing this term here.

Let us assume the new test image $y \in R^m$ can be represented by the training samples from the same class i, $y = a_{i,1}x_{i,1} + a_{i,2}x_{i,2} + ... + a_{i,n}x_{i,n_i}$ for some scalars $a_{i,j} \in R, j = 1, 2, ..., n_i$. As i is unknown,if we use entire training set $\mathbf{A}$ to represent the test image, where $y = A\alpha_0 \in R^m$, its entries are all 0 except those i_{th} class ones. If $m > n$,the image size is larger than the total number of samples, this problem is overdetermined and the correct is unique. However, if $m < n$, the solution is not unique.

In the past literature, the solution is seeked by choosing the minimum L_2 -norm solution, $\alpha = \arg\min ||x||_2$ subject to $Ax = y$.

Note that α is dense,large number of nonzero entries corresponding to training samples from many different classes. Naturally, this property is not desirable. Therefore, we are seeking a norm that could bring the sparse characteristics to. One candidate would be $||.||_0$, which counts the number of nonzero entry,however,for a large sample

data of size n, pick k entries such that minimize the zero norm is NP-hard. Plus, the $||.||_0$ is not positive homogeneous. Therefore,we have to seek something else.

Recently, Dick [16] discussed the usage of L_1 norm for exact recover the signal with Basis Pursuit method. Candes [17] introduced the usage of L_1 norm in image reconstruction. Ma [15] used equation $\alpha = \arg\min ||x||_1$ subject to $||Ax - y||_2^2 < \varepsilon$ to find the sparse representation in his SRC algorithm. In this paper, we use the SLEP package by Ye [18] to solve the equation. This is a very fast package to solve the L_1 equation. Assume we got α_i^0,now come to the core part.

In 2008, Ma [15] proposed a new classification method, sparse representation-based classification(SRC), this method provides new insights into feature extraction and robustness to occlusion. Ma showed that the number of features and the correct form of sparse representation is more important than the choice of features. What is more, under the assumption that the occlusion and corruption are sparse with respect to the standard pixel basis, this framework can handle errors due to occlusion and corruption uniformly.

Based on SRC, in this paper, we want to introduce our novel method for face recognition. It blended the functionality of subspace embedding and classification perfectly. At the first stage, we use leave one out strategy, find out the initial representation vector of each image via other images via SRC method. Note that by introducing L_1 regulation norm, the representation vector would be sparse, ideally, those significant non-zero elements would be corresponding to the same class as the test image, as those images share a lot of features in common. Also, as stated in the abstract, this initial solution is robust to occlusion and corruption.

CHAPTER 2

METHODOLOGY

2.1 Introduction

In this section, we first apply SRC method to get an initial sparse solution, after that, we provide an algorithm to alternatively optimize the objective function in seeking specified dimension subspace and the updated representation vector.

Assume we are given a set of images A_i, $A_i = [x_{i,1}, x_{i,2}, ..., x_{i,n_i}] \in R^{m \times n_i}$ from k different classes,reshape them into column vectors and combine into $A = [A_1, ..., A_k] \in R^{m \times n}$ y is the i^{th} column of A, A_{-i} is A without y. we want to minimize, the weight vector in the following equation

$$\min_{\alpha_i} \frac{1}{2}||A_{-i}\alpha_i - y||_2 + \lambda||\alpha_i||_1 \tag{2.1}$$

Note that the term $\lambda||\alpha_i||_1$ is the regularity term here,λ is the controlling coefficient. Let us explain the purpose of introducing this term here.

Let us assume the new test image $y \in R^m$ can be represented by the training samples from the same class i, $y = a_{i,1}x_{i,1} + a_{i,2}x_{i,2} + ... + a_{i,n}x_{i,n_i}$ for some scalars $a_{i,j} \in R, j = 1, 2, ..., n_i$. As i is unknown,if we use entire training set $\mathbf{A}$ to represent the test image, where $y = A\alpha_0 \in R^m$, its entries are all 0 except those i_{th} class ones. If $m > n$,the image size is larger than the total number of samples, this problem is overdetermined and the correct is unique. However, if $m < n$, the solution is not unique.

As mentioned, the solution is seeked by choosing the minimum L_2 -norm solution, $\alpha = \arg\min ||x||_2$ subject to $Ax = y$.

Note that α is dense,large number of nonzero entries corresponding to training samples from many different classes. Naturally, this property is not desirable. Therefore, we are seeking a norm that could bring the sparse characteristics to. One candidate would be $||.||_0$, which counts the number of nonzero entry,however,for a large sample

data of size n, pick k entries such that minimize the zero norm is NP-hard. Plus, the $||.||_0$ is not a true norm,it is not positive homogeneous. Therefore,we have to seek something else.

Recently, Dick [16] discussed the usage of L_1 norm for exact recover the signal with Basis Pursuit method. Candes [17] introduced the usage of L_1 norm in image reconstruction. Ma [15] used equation $\alpha = \arg\min ||x||_1$ subject to $||Ax - y||_2^2 < \varepsilon$ to find the sparse representation in his SRC algorithm. In this paper, we use the SLEP package by Ye [18] to solve the equation. This is a very fast package to solve the L_1 equation. Assume we got α_i^0,now come to the core part.

Based on all these, we want to find a better representation in the embedded subspace, i.e, find a solution to minimize the empirical loss function, we want to preserve the nice perspective of the initial solution, i.e, the new solution should be quite close to the initial solution, meanwhile minimize the empirical test error in the embedded manifold space. As the loss function is a non-convex function and carries two parameters, the subspace and weight representation,we choose to alternatively optimize at each step, finally we would get a convergent optimal solution of the subspace embedding representation.

In this part,I want to discuss several possible objective functions, we would discuss the advantages and disadvantages for each one and would also compare their performances in the experiment part.We want to find an optimal representation in the embedded space,so of course we need to minimize the overall residual error over the training samples. So here comes the first candidate of objective function called unconstrained L_1 objective function.

$$\min_{\alpha_i} \frac{1}{2}||w^T A_{-i}\alpha_i - w^T x_i||_2^2 + \lambda||\alpha_i||_1 \tag{2.2}$$

w is the projection matrix. To minimize this objective function, we implemented an iterative gibbs optimization method.

When α_i is fixed, to solve the minimization problem with , this is equivalent to solve

the following problem:

$$\min_{w} \sum_{i} ||w^T A_{-i}\alpha_i - w^T x_i||_2^2 \tag{2.3}$$

It can be solved analytically, note that the original problem is equivalent to the following problem:

$$\min_{w} Tr(w^T(A_{-i}\alpha_i - x_i)(A_{-i}\alpha_i - x_i)^T w) \tag{2.4}$$

The solution to 2.4 is the k eigenvectors corresponding to the smallest k eigenvalues of the covariance matrix . Assume we get the **w** to 2.4, then plug into 2.2, we can optimize 2.2 with respect to α_i again, next plug in α_i again and use the similar idea to get new **w**, repeat the above process, and alternatively optimize with **w** and α_i. The above process stops when $\frac{\max|w_{t+1}w_{t+1}^T - w_t w_t^T|}{||w_{t+1}w_{t+1}^T||_F}$ falls with the specified tolerance, i.e, w converges.

Summarized procedure goes as below:

INPUT: a matrix of training samples $A = [A_1, A_2, ..., A_k] \in R^{mxn}$ for k classes, x_i is the i^{th} image, A_{-i} is A without x_i, the specified dimension k_1 for **w**, max number of iterations N, and the error tolerance ε

1. solve the equation 2.1 to get α_i^0
2. solve the equation 2.4 to get w_0

I want to emphasize that our algorithm guarantees the convergence of the objective function. It is non-negative lower bounded and our algorithm makes it monotone decreasing for each time optimization,therefore theoretically we can find the optimal w that minimize the objective function.

Now I want to discuss another objective function called constrained L_2 objective function

$$\min_{w,\alpha_i} \sum_{i} \frac{1}{2}(||w^T A_{-i}\alpha_i - w^T x_i||_2^2 + \beta||\alpha_i - \alpha_i^0||_2^2) \tag{2.5}$$

Algorithm 1 Nie's L_1

w_{old}=w_0

repeat

for every w_{old} **do**

solve 2.2 with w_{old}

plug in the solution from 2.2 to solve 2.3 to get w_{new}

end for

if convergence criteria of w met **then**

break

else

w_{old}=w_{new}

end if

until w converged or max number of iterations reached

Same as previous one, we first get the initial α_i^0,next we let $\alpha_i = a_i^0$ and similar to previous method we can get the initial w_0,after that we plug in, now 2.5 becomes

$$\min_{\alpha_i} \sum_i \frac{1}{2}||w_0^T A_{-i}\alpha_i - w_0^T x_i||_2^2 + ||\alpha_i - \alpha_i^0||_2^2 \tag{2.6}$$

To minimize 2.6,we just need the sum of two terms for each i, the analytical solution of can be derived as follows:

$$\min_{\alpha_i} \sum_i \frac{1}{2}||w_0^T A_{-i}\alpha_i - w_0^T x_i||_2^2 + ||\alpha_i - \alpha_i^0||_2^2 \Leftrightarrow \min_{\alpha_i} \frac{1}{2}||w_0^T A_{-i}\alpha_i - w_0^T x_i||_2^2 + ||\alpha_i - \alpha_i^0||_2^2 \tag{2.7}$$

as α_is' are independent.

There is an analytical closed solution to equation 2.7, we can derive it as follows: equation 2.7 is equivalent to

$$\frac{1}{2}(w^T A_{-i}\alpha_i - w^T x_i)^T(w^T A_{-i}\alpha_i - w^T x_i) + \beta(\alpha_i - \alpha_i^0)^T(\alpha_i - \alpha_i^0) \tag{2.8}$$

Take derivative with respect to α_i and set to zero, we get its formula

$$\alpha_i = [(w^T A_{-i})^T (w^T A_{-i} + 2\beta I)]^{-1} (w^T A_{-i})^T w^T x_i + 2\beta\alpha_i^0) \tag{2.9}$$

After we got new α_i,plug in and we can get new w as previous idea, we can repeat this process to iteratively optimize the objective to find the best w.The algorithm is very similar to algorithm one, the only difference is that different objective function lead to different forms of w and α_i.

Now I want to introduce another objective function called constrained L_1 objective function:

$$\min_{w,\alpha_i} \sum_i \frac{1}{2}(||w^T A_{-i}\alpha_i - w^T x_i||_2^2 + \beta||\alpha_i - \alpha_i^0||_1) \tag{2.10}$$

To tack this, as before get α_i^0 first,next let $\alpha_i = \alpha_i^0$, we can get w_0. Next step, we need to make a transformation $\eta = \alpha_i - \alpha_i^0$ to make equation 2.9 a canonical form , then equation 2.9 becomes $\min_{w,\eta} \sum_i \frac{1}{2}(||w^T A_{-i}(\eta + \alpha_i^0) - w^T x_i||_2^2 + \beta||\eta||_1)$

i.e,

$$\min_{w,\eta} \frac{1}{2} \sum_i (||w^T A_{-i}\eta - w^T(x_i + A_{-i}\alpha_i^0)||_2^2 + \beta||\eta||_1) \tag{2.11}$$

in other words, here we are minimizing η in $\mathrm{L}_1 norm, the difference of each\ \alpha_i$ and α_i^0,the reason of doing this is that the equation form 2.11 can be solved efficiently by the SLEP package by Ye [18] while equation 3.1 can't.

Here I want to make brief comments about the advantage and disadvantage of three objective functions.

Unconstrained L_1 is a concise one, which is straightforward and has a harmony form,while for the objective function, since it has no constraint about α_i, we have no control of the difference of α_i and α_i^0,the final α_i could diverge far away from the initial α_i^0,which would potentially compromise the accuracy.Indeed,the later experiment show that if the tolerance is small, say 10^{-4}, the w does not seem even converge!

Constrained L_2 has an explicit regularity term, would make α_i and α_i^0 close in the L_2 norm sense. As there is no L_1 norm involved, during the steps of minimizing the objective function, α_i has analytical close form solution, this would make the converge much faster. However, α_i would not be able to keep the property of sparse due to the L_2 regularity term.

Constrained L_1 use the regularity term in the L_1 norm, in the hope of α_i being close to α_i^0, meanwhile maintain its sparse property. The cost of doing this is that there is no closed form analytical solution, in other words, more time and resources consuming.

After we found the final w, the optimal subspace representation, we would like to discuss the classification part. Now, given a new test sample y from one of the classes in the test set, we first compute its sparse representation $\alpha_i^{'}$ via equation 2.1. If the images are noise-free in the ideal case, the nonzero entries in the $\alpha_i^{'}$ will all be corresponding to the class of training set where the test image comes from,however,due to the noise and modeling error,the actual $\alpha_i^{'}$ may have multiple classes of small nonzero entries. We can simply assign the test image to the class which has the largest magnitude nonzero entry. however,this simplification does not utilize the subspace structure,instead we want to classify the test image based on how well the coefficients associated with all training samples of each object reproduce the test image.

For each class i, let $\Gamma_i : R^n \to R^n$ be the characteristic function which selects the nonzero coefficients. We can classify the test image based on these approximations by assigning it to the class that minimizes the residual.

$$\min_i r_i(y) = ||w^T y - w^T A\Gamma_i(\alpha_i)||_2 \tag{2.12}$$

Our whole procedure can be summarized as follows:

1 Divide the images into training set and test set

2 For every image from the training set, we get an initial weight estimate of every image using other images, here we impose the regularized term L1 norm, so that the weight vector is a sparse vector.

3 We alternatively minimize the objective function with the subspace projection and the weight vector until the subspace representations converge.

4 Based on w and training set, for each image from the test set, classify it based on equation 2.12.

CHAPTER 3

EXPERIMENT

3.1 Classification

In this section, we want to compare the classification performance of our method, which consists of different objective functions, with other classification methods. Note that our method consists of two clear aims: first,dimension reduction, via find the best embedded subspace representation;second, classification, via take advantage of subspace structure to find the minimum error class.Note that our dimension reduction method belongs to the unsupervised category as there is no label information enclosed during the training process.

There are two popular unsupervised methods, one is PCA,principal component analysis,which makes the orthogonal projection of the data onto a lower dimensional linear space, known as the principal subspace, such that the variance of the projected data is maximized; the other one is LPP, locality preserving projection, which finds the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the manifold.

There are also quite a few classification methods, such as KNN, K-nearest neighborhood, via which an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors. SVM, support vector machine,which models a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. In this experiment, I would try both the linear kernel and RBF kernel.

In this experiment, I would compare my method with the combination of dimension reduction method and classification method, therefore, there are six possible

Figure 3.1. PIE sample image.

choices as follows: PCA + KNN, PCA + SVM(linear), PCA + SVM(RBF),LPP + KNN, In addition to those, I would also compare the performance with SRC suggested by Ma, after all, our method is inspired and closely related to this.

The first data set we would test on is a subset of the pie data set, the sample images are below 3.1, which consist of 680 face images, 68 classes, each class has 10 images, each image has been resized into 16 x 16 size. This dataset is quite difficult due to the large number of classes, varying expressions and illuminations. You can compare with the second data set I would mention. We randomly divide the whole data set into training set and test set, each set consists of 5 images for each class, 340 in total. The parameter setting is dimension=60, the coefficient for initial Ł$_1$ regularity term is 32, the coefficient β for controlling the solution difference is 1, the error tolerance for constrained L_1 method and Ding's method is = 10^{-4}, while for Nie's is = 10^{-2}, max number of iteration is 200. The parameters in SVM have been optimized based on the baseline parameters tested using the 2 folder cross validation. Table 3.1 gives the accuracy table based on the average of our experiment.

From 3.1,it can be observed that our methods(constrained L_1 L_1 method and Ding's L_2 method are significant better among all other unsupervised dimension reduction with classification combination methods. It also showed that our methods's accuracies are lower than Ma Yi's method, however, notice that we are doing the classification based on the subspace whose dimension is only about a quarter of Ma's dimension.

Table 3.1. Classification accuracy for dataset pie when dim=60.

constrained L_1	0.8632
constrained L_2	0.8423
unconstrained L_1	0.7922
$Mamethod$	0.8794
$LPP + MY$	0.798
$LPP + KNN$	0.787
$LPP + SVM(linear)$	0.811
$LPP + SVM(RBF)$	0.815
$PCA + MY$	0.728
$PCA + KNN$	0.466
$PCA + SVM(linear)$	0.723
$PCA + SVM(RBF)$	0.737

Table 3.2. Classification accuracy for dataset pie when dim=100.

constrained L_1	0.912
constrained L_2	0.905
unconstrained L_1	0.882
$PCA + MY$	0.766
$PCA + KNN$	0.5
$PCA + SVM(linear)$	0.743
$PCA + SVM(RBF)$	0.754

If we increase the dimension to 100, the accuracy would increase, please refer to Table 3.2 which based on the average of the experiment. It can be observed that increased dimension of the subspace would make our method beat MaYi's classification result. Also our methods still among the best among all dimension reduction plus classifier methods

One thing I want to emphasize, as also pointed out before, is that unconstrained L_1 method did not converge, the accuracy seems to be fine when $= 10^{-4}$, however, when is reduced to 10^{-4}, the accuracy becomes average around 0.02 meanwhile the maximum number of iterations reached, in other words, it did not even converge in the real sense, it keep rotating a small angle that ends up with a undesirable subspace which gives bad performance.

Figure 3.2. ORL sample images.

Table 3.3. Classification accuracy for dataset ATT when dim=40.

constrained L_1	0.962
constrained L_2	0.961
unconstrained L_1	0.732
$MYmethod$	0.953
$LPP + MY$	0.897
$LPP + KNN$	0.959
$LPP + SVM(linear)$	0.969
$LPP + SVM(RBF)$	0.891
$PCA + MY$	0.961
$PCA + KNN$	0.937
$PCA + SVM(linear)$	0.96
$PCA + SVM(RBF)$	0.878

The second data set we would look at is the ORL data set from cambridge university computer lab,the sample images are 3.2, which consist of 400 face images, 40 classes, each class has 10 images, each image has been resized into 14 x 11. We again randomly divide the whole data set into training set and test set, each set consists of 5 images for each class, 200 in total. The parameter setting is dimension=40, the coefficient for initial Ł$_1$ regularity term is 1000, the coefficient β for controlling the solution difference is 100, the error tolerance = 10^{-4}, max number of iteration is still 200. Table 3.3 gives the accuracy table

Note that in table 3.3, only with dim=40, we have already quite close results among other methods. Since ORL is a relatively easy data set for classification pur-

Figure 3.3. YaleB sample images.

pose,the light condition and the human gesture are almost identical. The results demonstrate another important key point mentioned, the main strength of our methods is that it can deal with the occlusion,varying expression and illumination.

The last data set we would look at is the YaleB database from Yale University computer vision lab,the sample images are below 3.3, the mat file from [2],This dataset now has 38 individuals and around 64 near frontal images under different illuminations per individual, each image has been resized into 16 x 16. We again randomly divide the whole data set into training set and test set, each set consists of about 32 images for each class, around 340 in total. The parameter setting is dimension=60, the coefficient λ for initial Ł$_1$ regularity term is 1000, the coefficient β for controlling the solution difference is 100, the general error tolerance = 10^{-4} except for Nie's is 10^{-2} as otherwise it would diverge, max number of iteration is still 200. Table 3.4 gives the accuracy table based on the average:

If we increase the dimension to 100, please refer to table 3.5,then again our methods stand out.

I want to draw the conclusion in this section that, via comparing the classification results on three data sets, it can be seen that our methods are among the best of the combinations of popular unsupervised dimension reduction and classification. In addition to that, it verifies the necessary of having the constrain term.

Table 3.4. Classification accuracy for dataset YaleB when dim=60.

unconstrained L_1	0.912
constrained L_2	0.886
unconstrained L_1	0.740
$MY method$	0.916
$LPP + MY$	0.75
$LPP + KNN$	0.845
$LPP + SVM(linear)$	0.645
$LPP + SVM(RBF)$	0.831
$PCA + MY$	0.851
$PCA + KNN$	0.471
$PCA + SVM(linear)$	0.723
$PCA + SVM(RBF)$	0.737

Table 3.5. Classification accuracy for dataset YaleB when dim=100.

unconstrained L_1	0.936
constrained L_2	0.916
unconstrained L_1	0.892
$PCA + MY$	0.872
$PCA + KNN$	0.502
$PCA + SVM(linear)$	0.895
$PCA + SVM(RBF)$	0.891

3.2 More investigations

In this section, I want to take a closer look at the method I proposed, a few graphs would be included to help the analysis, the dataset I used would. First, here comes the equation again:

$$\min_{w,\alpha_i} \sum_i \frac{1}{2}(||w^T A_{-i}\alpha_i - w^T x_i||_2^2 + \beta||\alpha_i - \alpha_i^0||_1) \tag{3.1}$$

First, I want to discuss the role of λ in my method, as it is obviously closely related to the initial α_i^0, which in turn affects our method performance.

Table 3.6. Accuracy for different λ values.

0.0001	0.01	0	1	100	10000
0.8176	0.8176	0.8206	0.8324	0.9	0.8235

From the table 3.6, we can see that the different λ values would have significant impact on the classification performance. It is also interesting to note that the accuracy when $\lambda = 0$ is higher than those extreme cases when λ is too large or too small.

Second, I want to discuss the role of β in our method, as it is the coefficient to control the difference of the initial solution and the final solution. Now λ is fixed, which is 1000, please refer to table 3.7. It can be observed that when β is 100, my method reaches the best accuracy.

Table 3.7. Accuracy for different β values.

0.0001	0.01	0	1	100	10000
0.8618	0.8647	0.8588	0.8588	0.8676	0.8588

Next thing, I want to consider the subspace dimension effect on YaleB dataset, intuitively the higher the dimension is, the more information might be kept there, it could help promote the accuracy. Here $\lambda = 1000$, $\beta = 100$. I used PCA as reference group, as it also clearly depends on the dimension. It can be observed that when at the low dimension, PCA displayed its advantage, while around dim=60, my method start to be better, the performance for PCA almost remains stable after dim=100; while my method ends up with accuracy around 0.85 when dim=150. However, I want to caution that I am not saying my method is better than PCA in the general sense, it really depends on the classifier. Keep in mind that my dimension reduction method really depends on the classifier, in this experiment, Ma's method, meanwhile,

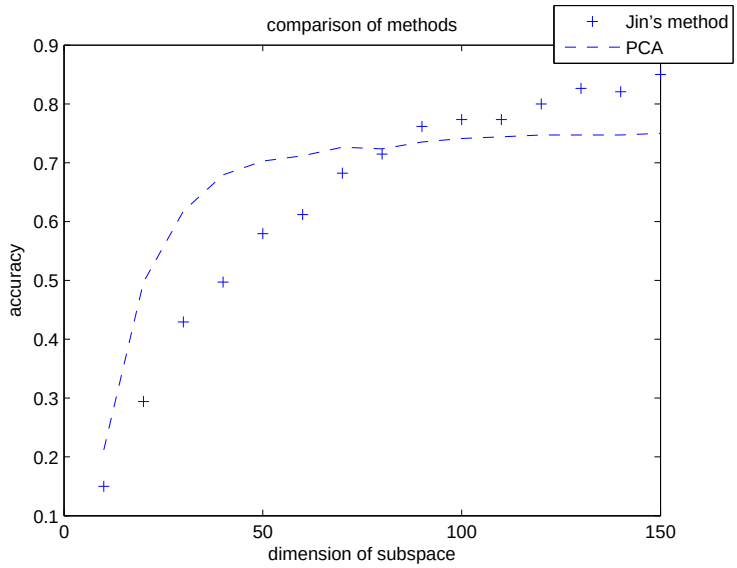

Figure 3.4. PCA vs Jin's method for dim from 10 to 150 on YaleB dataset.

PCA is a very general method that could almost be applied to most applications when dimension reduction is desirable.

It is also interesting to observe the number of iterations my method takes for different values of β, as it is important to make sure the method converges within the specified tolerance, here, the maximum number of iterations is 200, the specified error tolerance is 10^{-4}.

Table 3.8. The number of iterations for different β values.

0.0001	0.01	0	1	100	10000
132	35	35	35	200	200

The next thing I want to investigate is that for fixed α values, the graph of sum of α difference for different β values. From graph 3.6,it can be observed that when $\beta = 10$, the chain seems have experienced a quite stable burn in period before it fell within the specified tolerance, which is the desirable property I want as to make sure convergence.

What is next? I also need to check the value of target function, we are gradually rotating w to find the optimal subspace representation. Then what is the angle between w values for different parameter β? Here is the matrix angle table for $\beta = [0.0001, 0.01, 0, 1, 100, 10000]$, note that for matrix 3.2, each entry corresponding to the angle between different β values. for example, 0.1922 is the angle between the ws when $\beta = 1$ and $\beta = 100$.

0	0.0098	0.0388	0.0262	0.2018	0.2018
0.0098	0	0.00394	0.0254	0.2006	0.2006
0.0388	0.00394	0	0.0362	0.2088	0.2088
0.0262	0.0254	0.0362	0	0.1922	0.1921
0.2018	0.2006	0.2088	0.1922	0	0.0202
0.2018	0.2006	0.2088	0.1921	0.0202	0

It would also be critical to observe the converge criteria, which is $\frac{\max |w_{t+1} w_{t+1}^T - w_t w_t^T|}{||w_{t+1} w_{t+1}^T||_F}$, for different values of beta, it takes different numbers of iterations to fall within the loop. It can be seen that for extreme large or small values, the convergence evaluation statistic displays no pattern, always keep fluctuating. While for $beta = 10$, after reasonable number of iterations burning period, it seems well fell into the specified tolerance.

Note that the ratio converges does not guarantee the convergence of w,we also need to make sure the denominator,at least,does not increase during the iteration, because otherwise the ratio could converge even if the numerator does not converge at all. Please see figure 3.7 below,

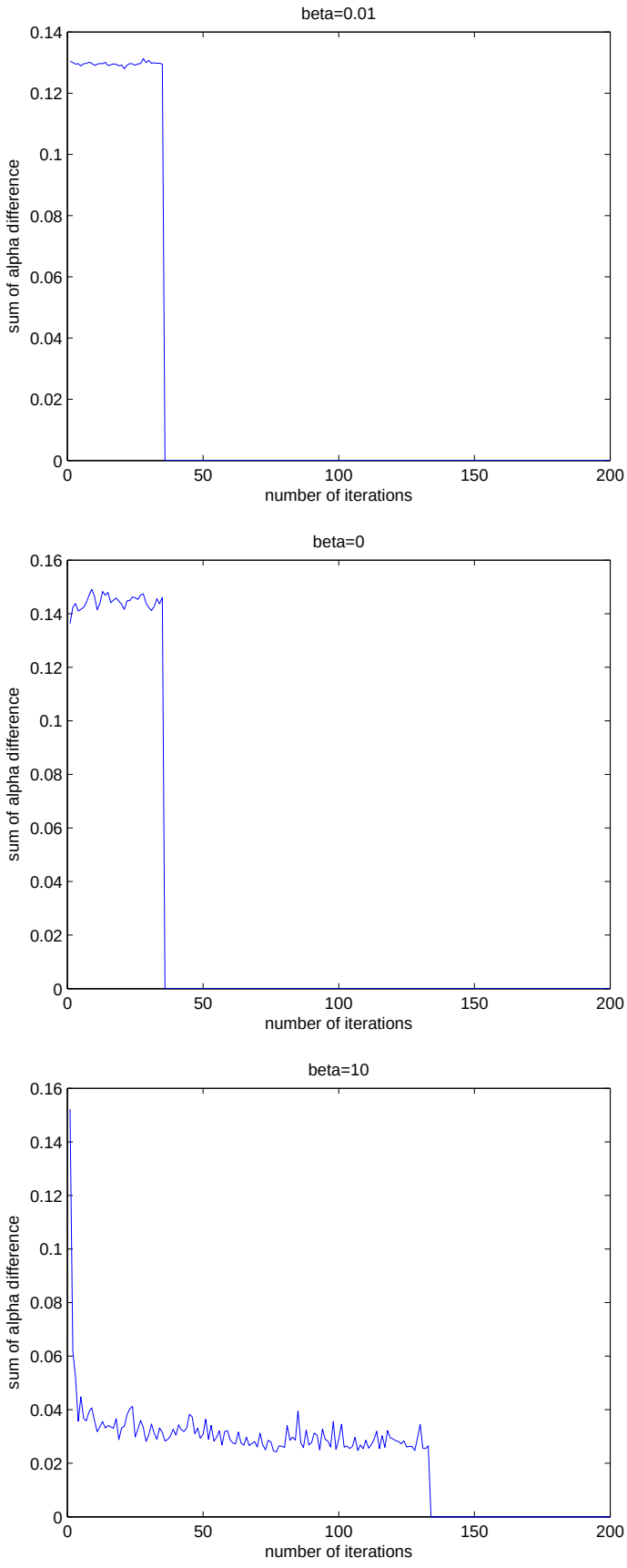

Figure 3.5. Sum of α difference when $\beta = 0.01, 0, 10$.

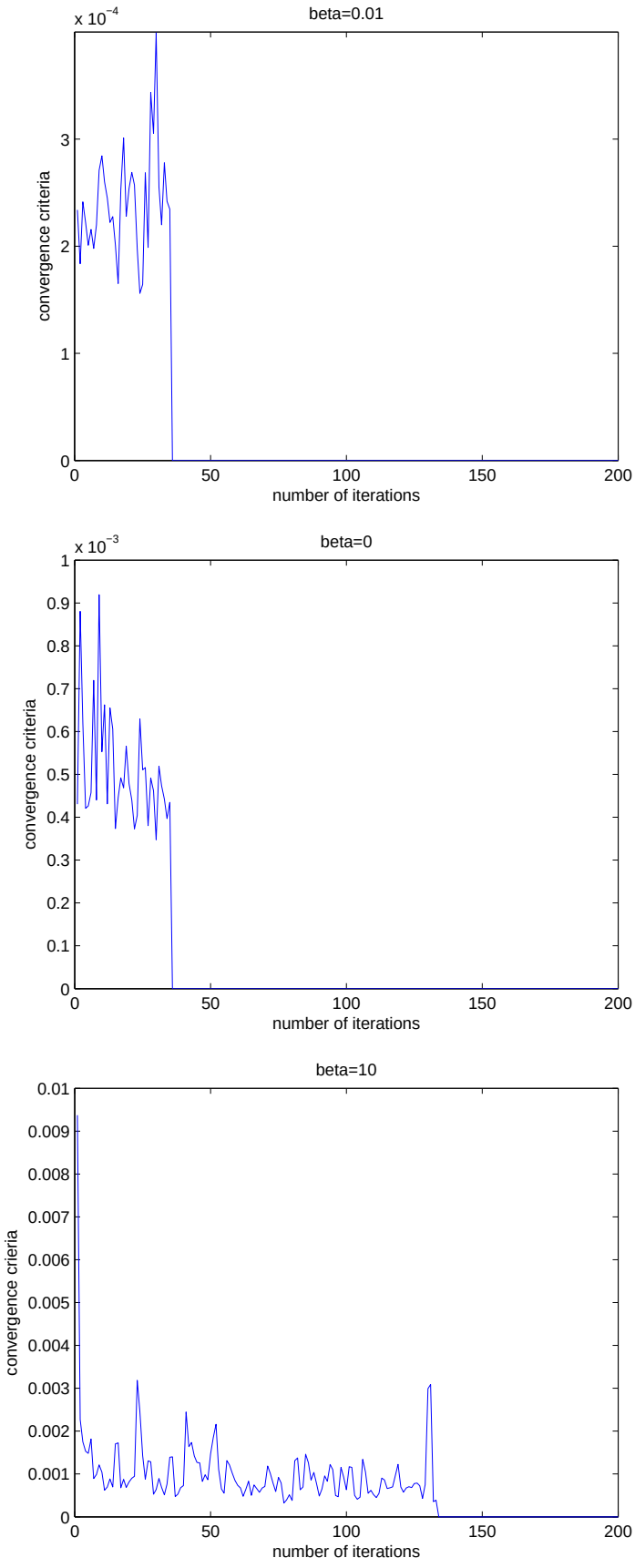

Figure 3.6. Convergence static value when $\beta = 0.01, 0, 10$.

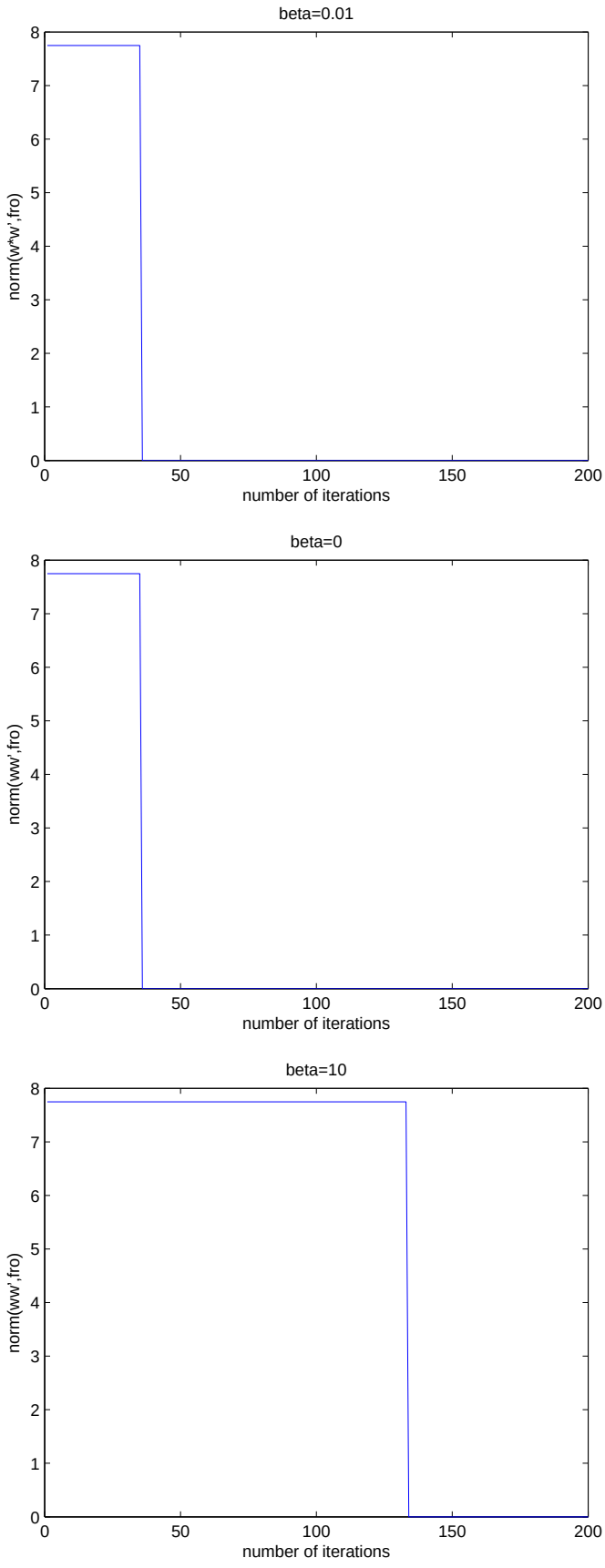

Figure 3.7. The w norm when $\beta = 0.01, 0, 10$.

Table 3.9. Accuracy for dataset YaleB when image is 32x32

unconstrained L_1	0.952
constrained L_2	0.937
$MY's method$	0.918

I would also like to include a few graphs about the optimal objective function values, this is a very important statistic when we tried to decide the regularity parameter as we would like to balance the residual error and the sparsity. I already included the sparsity regularity term graph in the previous part. As you can imagine, if I impose too much weight on the regularity term, it could significantly compromise the performance of my method, since the linear representation would become trivial, however, if the regularity term is negotiable, then we could not get the desirable sparsity.

The graph we get would be indeed inaccurate, since SLEP is package that seeks global solution, after the objective function value decreased to global minimum very fast, it start to go up and down. I contacted the author of the package SLEP, he replied:"in practice, when close to the optimal, my function cannot give a consistently decreasing objective function value, which is in accord with the convergence analysis."

3.3 Applications

One scenario that showed our method power is the case when the image size is very large, for the YaleB dataset we get, it is of size 32 by 32 for each image, if we did not do any image resize and apply our methods directly, it can be observed that our methods still maintain a high classification accuracy rate using now relatively low dimension. Here are the experiments result(based on five times average):

It can be seen that with increase image size, our methods both displayed satisfying results. It indeed has more significance than the appearance. Although in the experiments, we all used face images, which can be resized to the desirable size

for classification purpose, in real life high dimensional data, the resize operations are usually not applicable. This experiment shows that we can directly handle this kind of data with low dimension with comparable result with MY's method. As Ma mentioned in [15],his method takes several seconds on a 3G HZ machine for each image, the computation cost becomes too expensive for large scale high dimensional data, our methods showed a nice "Spend time now and Save time later property", with the ample training samples, we can significantly reduce the testing images size, which would reduce the total computation time. So this is a method of potential applications to large size data.

CHAPTER 4

CONCLUSION AND FUTURE WORK

In this paper, we have contended both theoretically and experimentally that finding an optimal subspace is critical for high-performance classification of high-dimensional data such as face images,also we showed the importance of having constraint of the new solution and the initial solution in the original space. We showed the effectiveness of our methods by comparing with other methods on three different data sets, also demonstrated the convergence of my L_1 method via analyzing the intermediate variables. Also, with sparsity properly harnessed and subspace representation well chosen, the choice of subspace dimension remains low for even high dimensional data, one can achieve very surprising classification result with only relatively low dimension.

An implicit assumption that the sample size greater than the image size, therefore we have to resize the image, otherwise, when we tried to minimize the objective function 2.4, we would find the the eigenvectors that corresponding to the eigenvalue 0, that is becomes the covariance matrix is not of full rank. To avoid this problem, we can exclude those trivial eigenvectors,and start to pick the eigenvectors from first non-zero eigenvalues. Although this is feasible, in our experiment, as long as the classification accuracy seems fine for the resized image, we all make the image size less than the sample size.

Although we have made a lot of efforts trying to reveal the relationship between the parameters in our method. There are still a lot of stuff not clear yet that need further investigations. Since this method is quite expensive in computation, it is quite difficult for us to tune the optimal parameters setting in the real sense. What is more, although using L_1 term as regularity term is desirable, in general it has no analytical closed form solution, when use alternative optimization method is applied in a non-

convex region, there is no guarantee we can find the global optimal solution.Another obvious future work is trying to make sure of label information, in other words, trying to enclose the label information in our objective function, changing from unsupervised method to supervised method, it is expected to boost the accuracy.

REFERENCES

[1] Baback Moghaddam and Alex Pentland. Probabilistic visual learning for object detection. pages 786–793, 1995.

[2] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(3):328–340, 2005.

[3] Peter N. Belhumeur, Joao P. Hespanha, and David Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 711–720, July 1997.

[4] Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[5] Matthew Turk and Alex Pentland. Eigenfaces for recognition.

[6] A.P. Pentland M.A. Turk. Face recognition using eigenfaces.

[7] Alex Pentland, Baback Moghaddam, and Thad Starner. View-based and modular eigenspaces for face recognition. In *IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION PATTERN RECOGNITION*, 1994.

[8] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.

[9] Marian Stewart Bartlett, Javier R. Movellan, and Terrence J. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13:1450–1464, 2002.

[10] Chengjun Liu and Harry Wechsler. Comparative assessment of independent component analysis (ica) for face recognition. In *International Conference on Audio and Video Based Biometric Person Authentication*, pages 22–24, 1999.

[11] Baback Moghaddam. Principal manifolds and bayesian subspaces for visual recognition. In *INTERNATIONAL CONFERENCE ON COMPUTER VISION, CORFU*, pages 1131–1136, 1999.

[12] Peter N. Belhumeur, Joao P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*

[13] Aleix M. Martinez, Aleix M. Mart'inez, and Avinash C. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:228–233, 2001.

[14] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey, 2000.

[15] John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):210–227, 2009.

[16] Chris Dick, Fred Harris, and Michael Rice. Synchronization in software radios-carrier and timing recovery using fpgas.

[17] Emmanuel C, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, 2004.

[18] J. Liu, S. Ji, and J. Ye. *Sparse Learning with Efficient Projections*, 2009.

BIOGRAPHICAL STATEMENT

Jin Huang was born in Suzhou, China, in 1981. He received his B.S. degree from Dalian University of Technology, China, in 2004. His M.S. degrees in financial math from The University of Lousiana at Lafayette in 2006 and M.S. degree in statistics in 2009 from Bowling Green state University,Ohio.