Optimization in Face Recognition

Ankit Raj (12125), L.S. Vishnu Rao (12376)
Department of Electrical Engineering
EE609A: Convex Optimization in SP/COM
Mentor: Prof. Ketan Rajawat, IIT Kanpur

***Abstract*—Face Recognition is a challenging and well researched problem in the field of Computer Vision and Image Processing. A face recognition system should be capable of identifying a person from an image or a video frame from a video source. Face Recognition is framed as a classification problem with faces representing the classes. About 80% of the faces belonging to a particular class are used for training that class. Rest 20 % are used for testing. We have used Model based representation of an image and used Linear Representaion based Algorithms for classfiication task. Face Recognition has got various applications in the fields like Security, Image collection, Identity Verification. Challenges of any face recognition system is to work irrespective of occlusion, different types of illumination, blurring and shading. We have made a comparative study of the three algorithms that use different optimization frameworks for the classification task.**

***Keywords*—Face Recognition, Optimization, Sparse Representation, Linear Representation Ensemble, l_1 minimization, Regression, Relaxation**

I. Introduction

In this term paper, we have investigated the application of convex optimization in the field of "Face Recognition". Image is generally represented either as Model based representation or as Feature based representation. In the feature based representation, an image is represented as collection of features. These features are obtained at the keypoints (interest points) of the image. In the feature based face recognition system, features corresponding to keypoints of face (nose, ear, eye, mouth) are found out and relative position of them with that of the training images is obtained using these features. If features are matched (distance less than threshold), then the image belong to the same subject as the training image. We are using Model based representation for this task. Linear Representation based Algorithms, which form a subclass of Model based representation, have gained popularity in the Face Recognition in the recent decade. Sparse Representation Classification (SRC) and Linear Regresssion Classification (LRC) are two popular variants in this class of algorithms. The LRC algorithm [1] solves the L2 norm minimization problem over the entire training set. The class which gives the least reconstruction error is the obtained class of the face. The SRC algorithm [2] solves a second-order cone problem over the entire training set with constraint that image has sparse representation over redundant dictionary. The above two algorithms use the assumption that the entire image lies in the subspace spanned by one of the class of training samples. However, this assumption isn't true for some of the images, particularly for those which are occluded and have partial defects. Of late, Linear Representation Ensemble (LRE) algorithm [3] has addressed this issue by assigning classes to the patches and not the entire image, using Probabilistic Patch Representations (PPR). Thus, classification of entire image is augmentation of the classification of the patches. This algorithm considers the linear subspace assumption on patches and not the entire image. This algorithm includes finding weights vector obtained by minimization of convex cost function on the training samples.

II. Method 1: Sparse Representation Classification [2]

A. Model

Consider a discrete time signal x, which can be represented as $N \times 1$ column vector in R^N. The signal x can be represented as linear combination of $N \times 1$ orthonormal basis vectors ψ_i as the orthonormal basis vectors ψ_i span the entire R^N space. But, if we increase the number of vectors in ψ_i such that the matrix becomes more redundant, the representation of signal in terms of basis vectors will be of size $> N$ but with most of the elements zero. This representation of signal in terms of combination of few basis vectors is called sparse representation. The signal x is K-sparse if it is a linear combination of only K basis vectors. Hence, instead of getting N coefficients for representation (using N basis vectors), the sparse signal can be expressed with lesser non-zero coefficients ($K < N$). This can be seen clearly in Fig. 1 . Not all signals do have sparse representation but the natural signals (like images) are inherently redundant. We have used this property fo images for face recognition task.

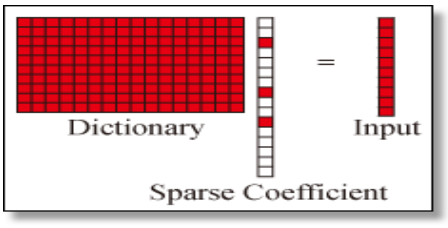

Fig. 1. Sparse Representation

For a redundant dictionary of size $M \times N$ with ($N > M$), a signal can be represented with few non-zero (in Fig. 1, it is three) coefficients, rest all are zero.

B. Optimization framework

Collection of basis vectors (redundant) which give sparse representation for signals is called as Dictionary with their columns as atoms. We would be using these terminologies in

[1]Source: http://ranger.uta.edu/~huang/R_Cervigram.htm

our report. To obtain a sparse representation of a signal, given good dictionary is equivalent to solving following optimization problem:

$$\alpha = \arg\min_{\alpha} \|\alpha\|_0 \quad s.t. \quad D\alpha = x \tag{1}$$

wherer α is the sparse representation of the given signal x with dictionary, D. Minimizing l_0 norm is equal to minimizing the number of non-zero elements of α which is required for sparse representation and signal reconstruction should also be done properly. For signal reconstruction, the constraint is $D\alpha = x$. Minimizing l_0 norm is a non-convex problem and it can't be solved analytically or using standard convex optimization toolbox. Different algorithms to solve Equation 1 exist. Most commonly used algorithms are greedy ones which finds best atom at every iteration. Iteration stops once the difference between $D\alpha$ and x is quite less or α_0 norm has crossed certain threshold. Popular algorithms are Orthogonal Matching Pursuit [4], Simultaneous OMP [5]. For our case, we have relaxed the l_0 norm so that the optimization problem becomes convex. The nearest convex norm to l_0 norm is l_1 norm so, α_0 in Eqn. 1 is replaced by α_1 to obtain:

$$\alpha = \underset{\alpha}{\operatorname{argmin}} \|\alpha\|_1 \quad s.t. \quad D\alpha = x \tag{2}$$

We solved the above optimization problem for dictionary trained on each class (face). The class (represented by dictionary) which gives the minimum error (residue) is the detected class (face).

C. Dictionary Learning

Dictionary is a very crucial component of sparse representation as only good dictionary enables formation of sparse representation of the given signal. The property satisfied by an ideal dictionary are Restricted Isometric Property [6]. Other parameters that decide the effectiveness of dictionary are Mutual Coherence [7] and Spark [8].
For our case, we have used directly the pixel values of the image concatenated as a column. Each grayscale image is of the size $a \times b$ and it is represented as the vector belonging to R^{ab} space. We have tested the algorithms in relatively smaller dataset. In the Olivetti Dataset [9] that we used, there are 10 images per class for each of the 40 classes. We used first 8 images per class for the dictionary. So, we have 8 atoms per dictionary and we have 40 such dictionaries (per class). Rest of the 2 images per class are used for testing the algorithm. We could have used certain feature (for e.g. Local Binary Patterns) of an image to obtain the dictionary.

D. Algorithm

Steps of the algorithm:

1) Obtain dictionary for each class using the pixel values of the grayscale image as described in section 3. The combined dictionary is of the form:

$$D = [D_1, D_2, D_3, ..., D_{40}] \tag{3}$$

where each $D_i s$ has atoms corresponding to pixel values (concatenated to form a column) of the first 8 images belonging to the i^{th} class.
2) Concatenate the pixel values of each test image to get the signal vector.
3) The equation 2 has been modified as the number of training images (atoms per dictionary for each class) is quite less. So, the constraint $D\alpha = x$ has been further relaxed and the optimization changes to:

$$\alpha = \arg\min_{\alpha} \|\alpha\|_1 + \|D\alpha - x\|_2 \tag{4}$$

We have used CVX toolbox to solve the optimization problem.
4) Solve Equation 4 for each test image where x is the signal vector to get corresponding sparse representation, α using each of $D_i s$ for $1 < i < 40$. Hence, there are 40 sparse representations, $\alpha_i s$ for each test images.
5) Now, obtain residue for each class using the following equation:

$$\epsilon_i = x - D_i \alpha_i \tag{5}$$

6) The class with the minimum residue is the detected class:

$$k = \arg\min_{i} \|\epsilon_i\|_2 \tag{6}$$

7) Do this for all the test images.

III. Method 2: Linear Regression based Classification [1]

Patter recognition problem has been formulated as a Linear Regression Problem. A Linear Regression Problem follows a fundamental concept that signal lies in a particular subspace. A linear model (space) for each class has been developed which computes the sub-space nearest to the test image. The model(class) which gives the least reconstruction error is the detected class. Intuitively, if the basis vectors for each are correct, each class should best span or give the nearest sub-space to the test image of that particular class. This intuition we are trying to exploit for the face recognition problem.

A. Model

Linear Regression uses the simple principle of representing a signal using some basis vectors. For perfect representation, signal must lie in the space spanned by the basis vectors. In most of the cases, there won't be perfect reconstruction and then it is desirable to get the nearest the nearest sub-space spanned by the basis vectors to the input vector.
For a given basis matrix $B = [b_1, b_2, b_3, ..., b_N]$ with each b_i belongs to R^M space. We want to get a test vector, y as some linear combination of the basis vectors of the matrix B. The combination obtained, x is the linear regression representation of the input signal, y. In mathematical form, we are trying to solve the following equation:

$$find \quad x \quad s.t. \quad y = Bx \tag{7}$$

B. Optimization framework

To solve Equation 7, input signal y must lie in the sub-space spanned by B, basis matrix. It would be highly optimistic to believe this would happen in a very high-dimentsional space of the image signal. Each image of size $a \times b$ is being represented as a signal in R^{ab} space. The best thing that can be done to get linear representation is to get the sub-space spanned by the vectors of B nearest to the test signal. Mathematically, it can be written as:

$$x = \underset{x}{\operatorname{argmin}} \|y - Bx\|_2 \tag{8}$$

C. Algorithm

Steps of the algorithm:

1) Obtain the basis matrix B_i for every class (i.e. $1 < i < 40$). The vectors of B_i represent the space spanned by each of the class. For each i, represent each element of B_i is the image of size $a \times b$ represented as R^{ab} size vector. 8 training images per class are used to comprise the B_i matrix. So, the size of B_i matrix is $ab \times 8$.
2) Each test image of size $a \times b$ is represented as R^{ab} size vector (y). 40 (Number of classes) optimization problems need to solved to get linear regression representation $output_t$

$$output_i = \underset{x}{\operatorname{argmin}} \|y - B_i x\|_2 \tag{9}$$

3) Now, compute reconstruction error obtained because of the nearest sub-space spanning for each class.

$$\epsilon_i = y - B_i output_i \tag{10}$$

4) Idea is that the matrix corresponding to the class of the test image should give the least reconstruction error.

$$k = \underset{i}{\operatorname{argmin}} \epsilon_i \tag{11}$$

k represents the detected class.
5) Do this for all the test images and get accuracy based on the detected class and the actual class.

IV. Method 3: Linear Representation Ensemble based Classification [3]

A. Model

In the above two methods it is observed that a given image is represented in a vector format and this vector is attempted to be evaluated as a linear combination of other vectors present in the dictionary. These methods have one key disadvantage. Since the images usually considered in the study are of larger dimensions, the vector representing them should also be of bigger dimensions. Evaluating these big dimension vectors as a linear combination of basis vectors present in the dictionary would demand the size of the dictionary to be larger. Thus there is a constraint either in the dimension of the vector representing the image or the size of the dictionary which evenutally leads to the ineffectiveness of the above methods.

A novel idea to tackle this problem of dimensionality is to represent a patch of the image in vector format rather than reperesenting the entire image. The small size of the patch would help in using a reduced dimension of the dictionary. Moreover, the outcomes of all the patches would have to be combined in an ensemble learning fashion and thereby improving the accuracies. If we combine (ensemble) the results of many such patches, we expect better performance. We have tested the algorithm by varying the number of patches and seen its effects.

B. Algorithm Overview

Rectangular patches are generated on all images including the training and the testing ones. These patches have to be from different locations of the image and have to be of varying sizes, however, they must be consistent over the images. Consider that there are N patches per image and k classes in the study. The pixels within a patch of the image are put in a vector format thereby resulting in N vectors per image. These vectors can be of varying dimensions corresponding to the dimension of the patch from which they are formed.
The N patch vectors belonging to an image are transformed into N probability vectors of fixed dimensions, the method described in the following section. The dimension of these probability vectors would be equal to k and each element of the vector would denote the probability that the patch would belong to a particular class i of k. The N patch probability vectors are weighted summed using a vector α resulting in a single k dimensional vector, ξ, per image. The argument corresponding to the maximum index is taken as the class of the image.

C. Optimization framework

There are two optimisations involved in this method, the first one to form probabilistic patch representations (PRRs) from normal patch represention and the second one in determining the vector α which is used in combining the PRRs.

1) Determining PRRs: Having generated N patch vectors per image, these vectors have to be transformed into probability vectors. We employ the same method of dictionary based linear representation as used in the previous methods but now apply it for the patches. If X_k^t is the dictionary corresponding to the k^{th} class and t^{th} patch. If y^t is the t^{th} patch of an image y, then

$$y^t \approx X_k^t \beta_{t,k} \tag{12}$$

The optimisation would result in the following, Tr denotes transpose operator,

$$\beta_{t,k}^* = (X_k^t(Tr)X_k^t)^{-1} X_k^t(Tr) y_t \tag{13}$$

The reconstruction error for would be obtained for every patch and different classes would be obtained by

$$r_{t,k} = \|y_t - X_k^t \beta_{t,k}^*\|_2 \tag{14}$$

The probability vector denoted by $b_{t,k}$ would be evluated from the recontruction error as

$$b_{t,k} = \frac{e^{(-r_{t,k}^2/\delta)}}{\sum_{j=1}^{K} e^{(-r_{t,j}^2/\delta)}} \tag{15}$$

Intutively, we are trying to transform reconstruction errors into probability distributions by application of soft-max operator. The $b_{t,k}$ vectors can be stacked in the form of a matrix B, the number of rows of which would correspond with the number of patches and the number of columns would be the number of classes.

2) Determining $\alpha's$: The vector ξ described in the previous section can be obtained using the following equation

$$\xi = B\alpha \tag{16}$$

α can be obtained by minimising the cost function, N denotes the total number of training images,

$$Cost = \sum_{i=1}^{N} exp(-z_i) \tag{17}$$

z_i represents the confidence that ξ selects the correct label for the image. If ξ_p denotes the p^{th} index in ξ, then z_i can be found by

$$z_i = \frac{\sum_{j \neq l_i}^{K} \xi_{l_i} - \xi_j}{K-1} \tag{18}$$

The above equation after couple of manipulations will be equal to

$$z_i = \sum_{t=1}^{T} \alpha_t (b_{t,l_i} - \frac{1}{k}) \tag{19}$$

The final optimisation problem would then be

$$min \quad \sum_{i=1}^{N} exp(-\sum_{t=1}^{T} \alpha_t (b_{t,l_i} - \frac{1}{k})) \tag{20}$$

such that $\alpha \geq 0 \quad and \qquad \|\alpha\|_1 \leq \lambda$ where λ is threshold.
It has to be noted that a common dictionary has to be used for both training and testing optimisation. The training patch vectors can be collectively used as a common dictionary. Also note that leave-one out scheme is to be employed while performing optimisation on the training dataset. This is because the training patch vectors are already present in the dictionary resulting in $0-1$ probabilities and hence would not incorporate necessary discriminative information.

V. Results

A. Dataset Description

We have tested the three algorithms on the AT&T database [9]. The AT&T database is maintained at the AT&T Laboratories, Cambridge University; it consists of 40 faces(classes) with 10 images per subject. The database incorporates facial gestures, such as smiling or nonsmiling, open or closed eyes, and alterations like glasses or without glasses. It also characterizes a maximum of 20 degree rotation of the face with some scale variations of about 10 percent. 10 images corresponding to a particular subject (class) is shown in Fig. 2.

8 images per subject have been used for training in all the three algorithms. The testing has been done on the remaining 2 images per subject. here, each subject refers to a particular face type (class).

Fig. 2. 10 images from a subject in AT&T database

B. Comparative Performance of different algorithms on the dataset

The results of all the algorithms are put in the table I. We observe that the linear regression algorithm is performing slightly better than the linear representation algorithm. It was found during the code implementation that LRE though being a better algorithm theory-wise, consumed huge amount of time for execution. The execution time was proportional to the number of patches considered for the experiment. Due to the lack of computational resources, the first optimisation in the LRE algorithm was explicilty calculated and Equation 13 was used instead of cvx optimisation. Performance of the algorithm was hence dependent on the inverse operation present in Matlab. Sparse Representaion based algorithm's performance degrades because of the relaxation of l_0 norm minimization to l_1 norm minimization. This relaxation doesn't ensure that the representation is certainly sparse and the reconstruction error is also more than the linear regression based algorithm.
Figure 3 shows the variation of accuracies of the LRE algorithm with the number of patches. We observe that accuracy decreases as we move beyond 350 patches. This is because the matrix to be inverted gets bigger in size with increasing patch number and inv operation of matlab doesn't perform well on large matrices.

The results for the algorithms:

Algorithm used	Accuracy
Sparse Representation based classification	85 %
Linear Representation based classification	96.25 %
Linear Representation Ensemble with 250 patches	95 %

TABLE I. Accuracy of the three methods

Variation of LRE with number of patches:

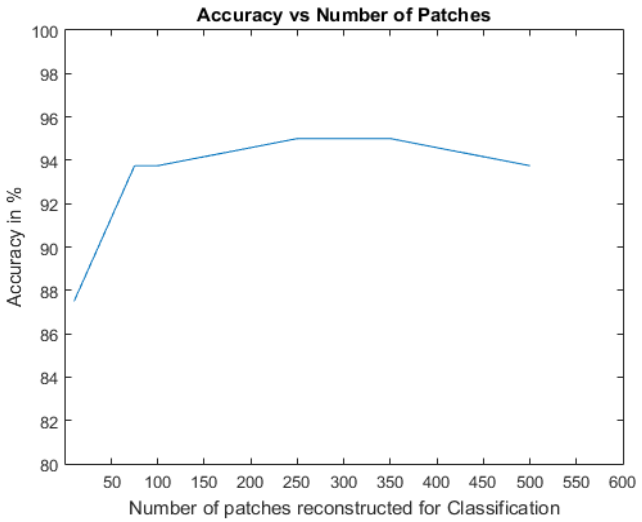

Fig. 3. Accuracy vs Number of Patches

VI. Conclusions

Linear Representation based algorithms are based on the fact that the test image is very close to the space spanned by the training images of the class same as the test image. Images being natural signals, are inherently redundant and can be expressed as sparse representation using atoms of good dictionary. Originally, there is l_0 norm minimization in sparse representation as minimizing l_0 is equivalent to keeping number of non-zero elements few. Relaxation of l_0 norm as l_1 norm has affected the accuracy and for the AT&T database, we have obtained an accuracy of 85 %.

Very good performance of Linear Regression based classification (LRC) indicates that images corresponding to a particular class in the used database are very near to each other in R^{ab} space if each image is of size $a \times b$. Another reason for very good performance of the LRC can be attributed to the highly differentiative classes in the dataset. The space spanned by vectors of one subject (class) is not close to any of the other class. Hence, corresponding to a test image, only the vectors from the corresponding class (for most of the cases) are able to represent the test image. The obtained accuracy for the LRC algorithm on the dataset is 96.25 %.

Linear Representation Ensemble algorithm works on the principle of dividing the entire images into multiple patches and employing linear representation algorithms patch-wise. This idea helps in working with smaller dimensional vectors which in addition with ensemble learning approach would likely be improving the face recognition accuracies. In our experiment, we observed that though LRE algorithm is a sound algorithm, it demands lot of computational resources due to multiple patch consideration.

Acknowledgment

First of all, we would like to thank the authors of the three papers for their wonderful work which motivated us to explore optimization in face recognition application. We would like to thank Mr. Javed for his valuable inputs throughout the project. We would like to thank Prof. Ketan Rajawat to help us out whenever we got stuck at something.

References

[1] R. T. Imran Naseem and M. Bennamoun, "Linear regression for face recognition," in *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 2010.

[2] A. G. John Wright, Allen Y. Yang and S. S. Sastry, "Robust face recognition via sparse representation," in *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 2009.

[3] H. Li, F. Shen, C. Shen, Y. Yang, and Y. Gao, "Face recognition using linear representation ensembles," *Pattern Recognition*, 2016.

[4] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.

[5] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. part i: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

[6] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9, pp. 589–592, 2008.

[7] M. Elad, "Optimized projections for compressed sensing," *Signal Processing, IEEE Transactions on*, vol. 55, no. 12, pp. 5695–5702, 2007.

[8] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *Information Theory, IEEE Transactions on*, vol. 50, no. 10, pp. 2231–2242, 2004.

[9] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*. IEEE, 1994, pp. 138–142.