Discriminative K-SVD for Dictionary Learning in Face Recognition

Piyush Nolastname, Ravi Teja Chikkam, Arizona State University

1 Introduction

Face recognition has been a problem of interest for quite some time now, owing to the sheer number of areas where it can be applied, as well as the challenges it poses. Face recognition, if mastered, can be applied in the field of authentication, personalization and security checks. As for the challenges, even the slightest of the changes in the image, ranging from variation in illumination, to occlusion, can throw off any algorithm. This makes it a difficult problem to solve. Second challenge in solving this problem is that conventional solutions for face recognition are not scalable and not accurate enough. Most of the conventional solutions , such as Eigenface[1] and FisherFace[2] involve two stages: feature extraction and classification. Feature extraction is something which can not be transferred easily across different domains, making it a difficult solution to implement.
To get around these issues, a technique called sparse representation using overcomplete dictionaries was applied with great success. Sparse representations have become increasingly popular over the last few years in the field of image processing in applications such as image denoising, image compression and face recognition. Sparse representations are employed to search for the most compact representation of a signal in terms of a linear combination of atoms in an overcomplete dictionary. Searching for the sparse representation of a signal over an overcomplete dictionary is achieved by optimizing an objective function that includes the signal reconstruction error and the measure of sparsity. It is achieved using the K-SVD algorithm[8], which is a generalization of K-Means algorithm. It is an iterative approach that alternatively updates the sparse representation vector based on the current dictionary, and the dictionary (atom-by-atom), to better fit the data. The update of the dictionary and sparse representation is intermingled to obtain a faster convergence. The dictionary obtained from the above process has only representational capabilities, but no discriminative capabilities. To make the dictionary representative of the sample subspace, so that samples from different classes can be distinguished from each other, the objective function can be modified to include the term measuring a linear classifier performance, so that the final dictionary learned is suitable for classification tasks such as face recognition as well.
However, the dictionary, sparse representation, and the classifier parameters are not learned simultaneously at the same time, hence resulting in only an approximate solution and most importantly, it runs the risk of falling into local minima of the sub problems. Also, many samples are required for training which is detrimental to learning a sparse solver.
The D-KSVD algorithm avoids the pitfall of falling into local minima by setting the objective function in a way that allows us to solve for the global solution for the Dictionary, Classifier and the Sparse representation simultaneously.
We will be using KSVD-BOX[1] which implements the K-SVD algorithm using Orthogonal Matching Pursuit heuristic[2] and implement the D-KSVD algorithm. We would also be training and testing the implemented algorithm using a synthetic data set the extended Yale Extended Face Database as well [3].

2 Related Work

There have been many works that approached the problem from the perspective of reducing the l2 norm distance without much success. Then came the feature extraction -classification such as eigenface and Fisherface. Robust face recognition utilizes sparse based representation to perform the classification task (SRC algorithm)[5]. However, in the SRC algorithm the training images are stacked together to form the dictionary in contrast to the dictionary learning that occurs in KSVD. Hence, SRC algorithm would require large number and carefully chosen training images. Attempts have been made to embed discriminatory power into the KSVD algorithm. A modification of one such algorithm is being considered as a baseline algorithm[6] against which the predictive power of D-KSVD is compared. The baseline algorithm fails at trying to solve the optimization problem. It does so iteratively by solving for W, sparse representation and D two at a time which results in sub-optimal solutions. LC-KSVD[7] (Label Consistent KSVD) was developed after the D-KSVD algorithm which includes an extra label consistent constraint which minimizes "discriminative sparse code error" along with the representational error from KSVD and classification error from D-KSVD. LC-KSVD has advantages over D-KSVD in providing better discriminative abilities across multiple classes because of the presence of label consistant constraint.

3 Problem Statement

It has been observed that image faces in varying illumination conditions lie in low dimensional spaces. Sparse based algorithms like the SRC algorithm, makes use of this property and tries to learn the sparse coding of the test image based on an overcomplete dictionary.
The face recognition problem can be formulated in the following way using the sparse based representation:
Given samples of i-th class:

$$[v_{i,1}, v_{i,2}, \ldots, v_{i,n_i}] \in R^{m*n_i},$$

any test image $y \varepsilon R^m$ will lie in the subspace spanned by the training examples:

$$y = a_{i,1} * v_{i,1} + a_{i,2} * v_{i,2} + \cdots + a_{i,n_i} * v_{i,n_i}$$

where a_{ij} is some scalar.

If we group the samples above, it called a dictionary.

$$D = [A_1, A_2, \ldots, A_k] = [v_{1,1}, v_{1,2}, \ldots, v_{k,n_k}] \quad (2)$$

where k represents the number of classes.

Now any test image can be represented as a combination of the atoms of the dictionary:

$$y = a_{1,1}*v_{1,1}+a_{1,2}*v_{1,2}+\cdots+a_{k,n_k}*v_{k,n_k} = D*x_0 \quad (3)$$

where

$$x_0 = [0, \ldots, 0, a_{i,1}, a_{i,2}, \ldots, a_{i,n_i}, 0, \ldots, 0] \in R^n$$

is the coefficient vector, which consists of all zeros, except for the values $a_{i,n}$ which are non zero values, representing the ith class.

Now any image for any sample can be represented by the equation:

$$Y = D\alpha,$$

where
α *is the sparse coefficient vector and D is the dictionary*.

Limitations of SRC Algorithm

There are a few limitations of this approach:

1. The number of training examples, or the dictionary size, required for the algorithm to work is huge.
2. To get around the problem of large number of training images, images must be selected manually in a way so they span a wide subspace in terms of representation.

Adding Discriminative Capabilities to K-SVD

The limitation of larger sized dictionary of SRC algorithm can be overcome by learning the dictionary using the k-SVD algorithm instead of using the images as atoms of the dictionary.

The K-SVD algorithm solves the following equation:

$$< D, \alpha >= \underset{D,\alpha}{\operatorname{argmin}} \|Y - D * \alpha\|_2 \ subject\ to\ \|\alpha\|_0 \leq T$$

where Y is the matrix of all input images and T is the sparsity constraint.

However, the dictionary learnt through this algorithm, however, representative, still lacks the discriminative power. It is because of the fact that the above equation only considers the sparsity constraint and nothing else. Hence, it is not fit for classification yet.

There have been works [3][4] that attempt to add discriminative power to the dictionary by adding classifier parameter terms to the equation.

$$< \theta >= \underset{\theta}{\operatorname{argmin}} \sum_i C(h_i * f(\alpha_i, \theta)) + \lambda_1 * \|\theta\|_2$$

The above equation is not difficult to solve and can only e solved using gradient descent methods, which is not optimal.

Another work [18] attempted to solve for the D and the W using the following equation:

$$< W, b >= \underset{W,b}{\operatorname{argmin}} \|H - W * \alpha - b\|_2 + \beta' \|W\|_2$$

The above equation should work, however, the way it is solved makes the solution suboptimal.

The W and the D values are solved iteratively using the following method:

1. Initialize D and α using KSVD.
2. Calculate W using the above equation keeping D and α fixed.
3. Calculate α when D and W are fixed.
4. Calculate D when α and W are fixed.
5. Repeat steps 2-4 until convergence is reached.

Limitations of discriminative approaches

There are two limitations of the aforementioned algorithm:

1. Since the D, W and α are solved iteratively, it runs the risk of running into local minima.
2. Since the optimization is tried to be achieved

without solving for the 3 values together, it takes a long time to converge to the solution.

4 D-KSVD ALGORITHM

The D-KSVD algorithm avoid the aforementioned limitations by solving for the D and W simultaneously, instead of iteratively.
Hence, D-KSVD algorithm avoids getting into local minima.

5. ALGORITHM

Training:

$$< D, W, \alpha >= \underset{D,W,\alpha}{\operatorname{argmin}}$$
$$\|\begin{pmatrix} Y \\ \sqrt{\gamma} * H \end{pmatrix} - \begin{pmatrix} D \\ \sqrt{\gamma} * W \end{pmatrix} * \alpha\|_2$$
$$subject\ to\ \|\alpha\|_0 \leq T$$

1. Compute the label matrix H and multiply with sqrt of gamma parameter.
2. Vertically concatenate modified H matrix with the Y matrix obtained from applying Randomface.
3. Choose the value for parameter alpha (sparsity factor), dictionary size etc.
4. Perform KSVD(can be implemented using KSVD function from KSVD-BOX).
5. Split the matrix obtained from KSVD output to generate dictionary matrix D and classifier W.
6. Re-normalize D and W to obtain D' and W'.

Classification:

$$l = W' * \alpha'$$

1. Compute the sparse representation for the test images using orthogonal matching pursuit. This can be implemented using omp function from the OMP-BOX.
2. Multiple the obtained sparse coefficients with the re-normalized classifier W'.
3. Classify the test image to the class that corresponds to the index with highest non-zero value.

6. ASSUMPTIONS

There were no assumptions involved with the experiment or the solution that we could find. The equations and the parameters used for this technique had proper mathematical backing.

7. EXPERIMENT SETUP

Extended Yale B Data

We test the algorithm on the extended Yale database first. The Extended Yale B dataset consists of 38 classes. Each class consists of about 64 images. The size of the image is 192x168 pixels.

We use randomly chosen 32 images from each class for training and the rest of the images as the test images.
We project the images using Randomface[5] into lower dimensions and then get the learned dictionary and the classifier using the KSVDBox. We used the following parameters as input to the KSVD function:

7. T Data (Sparsity factor) = 30
8. Dictionary Size = 570
9. Number of iterations = 50
10. sqrt gamme → = 2

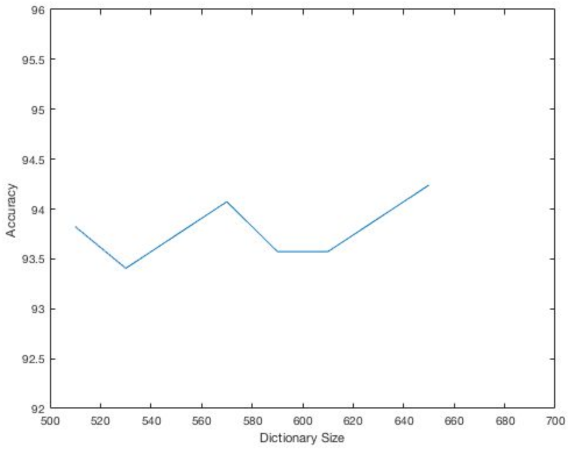
We normalize the dictionary and the classifier obatined from the output of KSVD. For classification, we iterate over the test images, get their sparse representations using the OMP box and then get the labels using the classification equation mentioned in section 5. Now we label the sample image using the max value from the label vector.

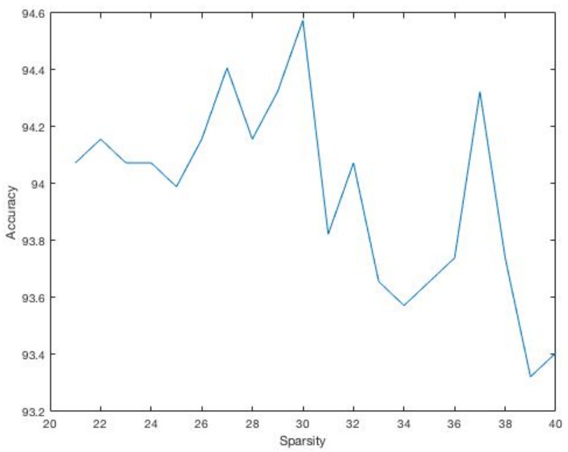
Synthetic Data

Our initial attempt at creating synthetic data, involved creating a vector of size 100 from a given value of mean and variance, adding noise to that vector and replicating the vector to create a collections of signals for single class, generated from the same distribution. To generate the samples for the second class, we use a different value of mean and variance and create the collection of signals as matix. However, we notice that the signals, belonging to the same class, are almost similar, unlike the distribution of the image data.
So, as our next attempt to create synthetic data, we create two arrays for two classes, having mean values of size 10 each. The first array, representing means values for class 1 contains float values ranging from -3 to 0. The second array, representing class 2, contains the float values ranging from the 0-3. We create an array of size 10 containing variance values, which is common for the two classes.
Now we create two matrices, for each class, containing 300 signals, by using the randomized sampled values from the each of the arrays containing mean and variance values. Each signal has a dimension of 100. We also add noise to the resulting matrices. Now we have the clusters representing the two classes. We pass these clusters to the KSVD box to get the learned dictionary and the classifier. We keep the dimensions of the dictionary as 150. We iterate for 50 times and the sparsity factor is 30 for this case. acceptable as there is a lot of overlap between our clusters, which results in some misclassifications.

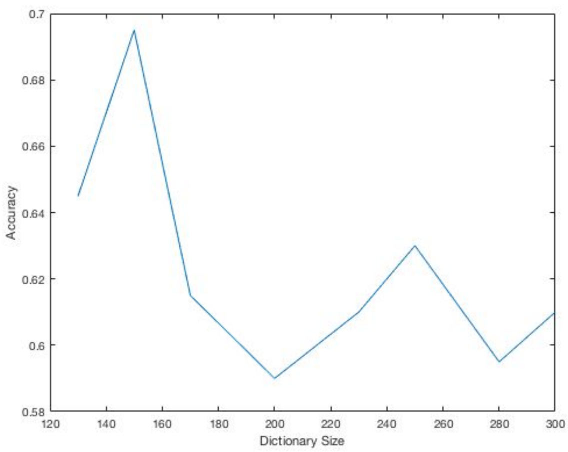
8. Results

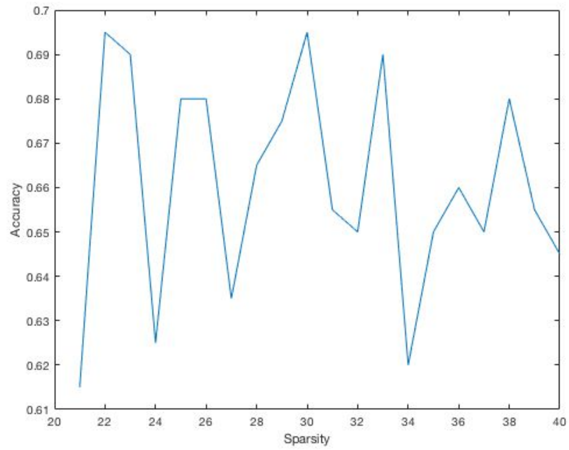
For the extended Yale dataset, we get an accuracy of 94%. We test the accuracy for different values of dictionary sizes and sparsity factors. The figure 1 shows the accuracy vs dictionary size graph and figure 2 shows accuracy vs sparsity factor graph.





For synthetic data, we get an accuracy of 69.50%. Below are the graphs of accuracy vs dictionary size along with sparse factor.





9. Problems during experiment

One problem that we faced while running the experiments was that while testing for synthetic data, the D-KSVD algorithm was easily able to classify the signals generated and get 100% accuracy. We tested by creating signals with different distributions and varying noise, with no success. On a closer look at the Yale image data, we found that the means and variance values of different signals belonging to even same class were significantly different from each other. So we had to keep this in mind while creating synthetic data.

10. Conclusion and Future scope

Problem of face recognition was solved by leveraging the ability of KSVD to learn an over-complete dictionary along with a sparse representation. As KSVD lacks discriminative ability a discriminative term has been added to the objective function. In contrast to the baseline algorithm D-KSVD solves for the dictionary, sparse representation and discriminative term simultaneously and thus avoid the pitfall of getting struck at local minima of the sub problems. D-KSVD has been implemented in matlab using KSVD-BOX, OMP-BOX and experiments were conducted to check its accuracy. Cropped version of extended YaleB dataset was used to evaluate the accuracy of D-KSVD in solving the problem of face recognition. Experiment was also conducted on synthetic dataset to again verify the classification power of D-KSVD. The results from these experiments establish the strength of D-KSVD in solving the classification task.

There is still scope for improvement, with regards to the requirements for training images, which if reduced, can make the algorithm very effective as well for ubiquitous usage. Future work can also focus on improving the run time of the algorithm as one of the major challenges of face recognition is scalability.

References

[1] M. Turk and A. Pentland. Eigenfaces for recognition. Journal of cognitive neuroscience, 3(1):71–86, 1991.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linearprojection. IEEE Transactions on pattern analysis and machine intelligence, 19(7):711–720, 1997

[3] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008, pages 1–8, 2008.

[4] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. Adv. NIPS, 21, 2009.

[5] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 210–227, 2009.

[6] D. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition. In IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008, pages 1–8, 2008.

[7] Z.Jiang, Z.Lin, L.David. Learning A Discriminative Dictionary for Sparse Coding via Label Consistent K-SVD.

[8] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on signal processing, 54(11):4311, 2006.

[9] https://www.umiacs.umd.edu/~zhuolin/projectlcksvd.html