MLCV Coursework 1 Reports

First Author
Institution1
Institution1 address
firstauthor@i1.org

Second Author
Institution2
First line of institution2 address
secondauthor@i2.org

Abstract

The ABSTRACT is to be in fully-justified italicized text, at the top of the left-hand column, below the author and affiliation information. Use the word "Abstract" as the title, in 12-point Times, boldface type, centered relative to the column, initially capitalized. The abstract is to be in 10-point, single-spaced type. Leave two blank lines after the Abstract, then begin the main text. Look at previous CVPR abstracts to get a feel for style and length.

1. Computationally Efficient Eigenfaces

Eigen faces are basis set of training image dataset. By using them, we can reconstruct or recognize faces. To make eigen faces of given images, we have to apply PCA to covariance matrix of images. This operation is quite time and space consuming while PCA needs cubic of feature dimension time for its computation. To solve this problem, we employed low-dimensional PCA, which can be used when number of training dataset is quite smaller than dimension of feature.

1.1. Comparison between PCA and low-dimensional PCA

As result, number of nonzero eigenvalues in covariance matrix $S = \frac{1}{N}AA^T$ is same with number of nonzero eigenvalues in matrix $\frac{1}{N}A^TA$. Moreover, as we can see in Figure 1 nonzero eigenvalues of each matrix are same. Also, if we multipy A to eigenvectors of low-dimensional matrix, they become same with eigenvectors of covariance matrix. However, while their resulting values and vectors are identical, their time complexities are different. Let feature dimension D and number of training data N. Then, computing covariance matrix and finding eigenvalues and eigenvectors of covariance matrix S takes $O(ND^2 + D^3)$. Low-dimensional PCA takes $O(2N^2D + N^3)$ to compute low-dimensional matrix, find eigenvectors, eigenvalues and multiply A to eigenvectors. So, if D is much bigger than N, low-dimensional PCA takes less computation time. However, if

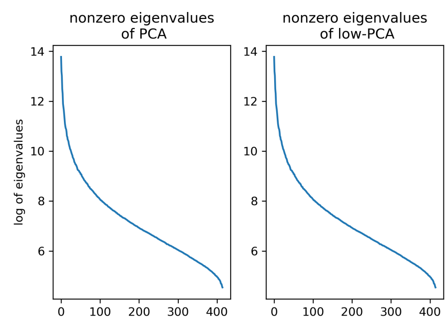


Figure 1: Plot of nonzero eigenvalues of PCA and low-dimensional PCA (low-PCA). Nonzero eigenvalues of each matrix are identical.

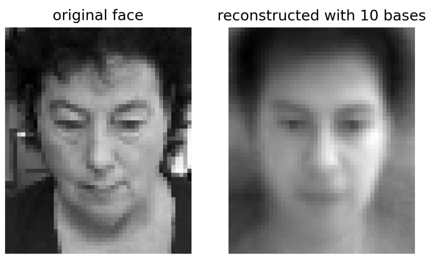
N is compatible with D, additional matrix multiplying with A might make low-dimensional PCA takes more time than original PCA. In our case, while N(416) is relatively smaller than D(2576), low-dimensional PCA takes less time to compute eigenfaces. Table 1 shows and compare training time of PCA and low-dimensional PCA.

1.2. Face Reconstruction

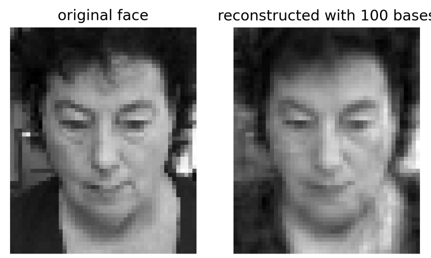
With eigenfaces optained from above section, we can reconstruct any given faces. Specifically, we optained basis of training face data and any face can be projected to our face space. Then, we can express those faces by linear combination of several eigenfaces. There are 415 eigenvectors with nonzero eigenvalues. Indeed, we can reconstruct all faces in training data exactly with 415 eigenfaces. Figure 2a 2b 2c shows using more eigenfaces can make more smillar faces and using 415 eigenfaces could reconstruct almost same face. However, several faces in test data cannot be reconstructed exactly, because test face could not be included in train face space. By comparing Figure 2c and Figure 2d, we can see that Figure 2c is reconstructed correctly with 415 bases, while Figure 2d failed with same number of bases.

Method	Training Time(sec)	Reconstruction Error	Recognition Accuracy(%)
PCA	9.36	-	-
low-PCA	0.79	11.93	62.5
PCA-LDA	17.50	X	92.3
subset PCA	0.14	22.84	61.5
Incremental PCA	0.54	18.46	62.5

Table 1: Results. Reconstruction error is measured by rmse with $m_{pca} = 50$. Recognition Accuracy is measured with $m_{pca} = 50$ for low-PCA, subset PCA and incremental PCA. $m_{pca} = 100, m_{lda} = 50$ is used for PCA-LDA

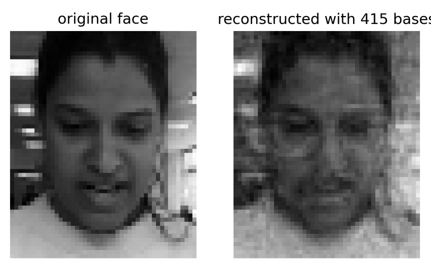


(a) Reconstructed face with 10 bases



(b) Reconstructed face with 100 bases

(c) Reconstructed face with 415 bases



(d) Failed image

Figure 2: Plots of reconstructed faces.

2. PCA-LDA for Face Recognition

LDA increase variance of features between other classes and decrease variance between same classes. In this section, we explore the effect of apllying LDA over PCA on face recognition accuracy in many aspects. Classifier used in this section is KNeighborsClassifier of python scipy library, which employed default settings.

2.1. Recognition accuracy

Let's first talk about recognition accuracy with PCA only method. As we can see in Figure 3, accuracy of 62% is achieved with 50 bases and saturated. On the other hand, recognition accuracy of PCA-LDA method is quite interesting. Using higher m_{pca} increases accuracy of recognition, because it can represent each face more accurately. Peak accuracy is achieved with $m_{pca} = 364$ and $m_{lda} = 31(100\%)$, and it suddenly fall down with higher m_{lda}. It could be caused by curse of dimension. Rank of within-class scatter matrix is $N - n_{class} = 416 - 52 = 364$ and rank of between-class scatter matrix is $n_{class} - 1 = 52 - 1 = 51$. While LDA has atmost rank of $(W_{PCA}^T S_W W_{PCA})^{-1}(W_{PCA}^T S_B W_{PCA})$nonzero eigenvectors, it has 51 nonzero eigenvectors. It means using m_{lda} larger than 51 cannnot increase variance of between-class features and decrease variance of within-class features. While variance is fixed, increasing dimension complicates classificatin because nn classification is based on

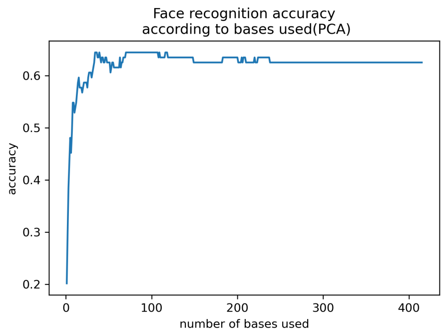


Figure 3: Accuracy of face recognition with PCA

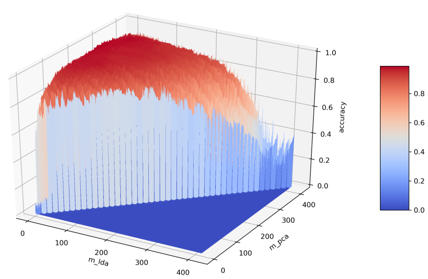


Figure 4: Accuracy of face recognition with PCA-LDA

eucledian distance, but too high dimension makes distance too large, eventhough they are in same class.

2.2. Example of success and fail faces

There are few examples of success and fail cases. Figure 5 shows successfully predicted faces. While test face in first row of Figure 5 seems sharper than other faces in same class, our classifier successfully predict its class. In failure cases(Figure 6), test face have simillar hairstyle with faces in wrong predicted class. It could be hard to classify them with other high accuracy classifiers.

2.3. Time and memory

Table 1 briefly compare computation time of PCA and PCA-LDA methods with fixed parameters. As explained in Section 1.1, low-PCA takes $O(2N^2D + N^3)$ time. However, LDA takes $O(NDmin(N, D) + \min(N, D)^3)$ time complexity and $O(ND + N\min(N, D) + D\min(N, D))$ memory. PCA-LDA has strong tradeoff between accuracy and time-memory comsumption.

3. Incremental PCA

In the reality, there could be case that we could not get full training data at once and given with some time gap. With normal PCA or low-dimensional PCA, we have to

Figure 5: Predict successed faces. First column shows test faces and second to fourth columns shows train faces included in class of successfully predicted.

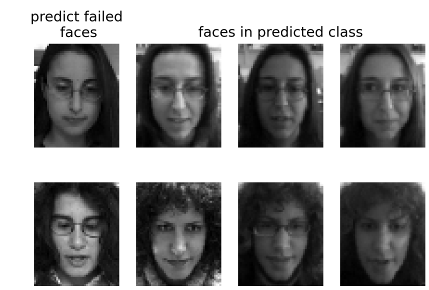


Figure 6: Predict failed faces. First column shows test faces and second to fourth columns shows train faces included in class of wrong predicted

train model from the bottom if training data is transformed. Incremental PCA is useful strategy for such data transformation. We also can reduce training time with Incremental PCA.

3.1. Comparison with other PCAs

There are normal PCA, low-dimensional PCA, PCA-LDA, PCA with one subset and incremental PCA. In our case, we want to deal with situation of gradual data incoming. So there are three PCA we can use in this situation: low-PCA, PCA with first subset and incremental PCA. In Table 1, we now can compare these PCA strategies. Using one subset PCA is fast, but it have big reconstruction error and low recognition accuracy. This is because small data brings low rank face space and low accuracy reconstruction of faces in test dataset. Low-dimensional PCA has highest reconstruction error and lowest recognition accuracy. Also, it seems time consumption is very compatitable with subset PCA and incremental PCA. However, its fact only with much low number of dataset than face dimension. As we can see in comparison for PCA and one subset PCA, training time cubicly increases. But incremental PCA does not. It have reasonable reconstruction error and recognition accuracy. Also, it's training time does not increase cubicly, and it is sum of all 4 subset PCA and time for merge them: it means, in real world, we need very few time to merge new data with existing PCA model.

3.2. Discussion in Incremental PCA

Many authors misunderstand the concept of anonymizing for blind