ENEE633 Project01 Reports

Neutral vs. facial expression

In this experiment, 400 images from 200 subjects are supposed to be classified to 2 classes. To evaluate the performance more precisely, datasets are shuffled and split but the images belong to the same subject still falls into one subset. In other words, subjects for training are randomly selected from 200 subjects and their corresponding pictures are goes into training set. Remaining pictures goes into test set. With different parameter setting and models, experiments are repeated for 10 times to get the average accuracy.

PAC parameter lamda is selected according to the following experiment. The method is PCA(lamda)+Bayes. From the following table, the optimal choice of parameter lamda is 0.8 since its accuracy is highest. For the following experiments, the lamda is fixed to be 0.8.

lamda	**0.5**	**0.6**	**0.7**	**0.75**	**0.8**	**0.85**	**0.9**	**0.95**
Dimension	9	16	25	33	43	56	79	117
Accuracy	0.8240	0.8720	0.8610	0.8780	0.8820	0.8560	0.8460	0.8430

The lamda influences the number of dimensions when projecting the data. A larger lamda means a higher dimension space we project to. Usually in PCA, with a higher dimension space, in other words, a larger dataset, we will have more information but in this case, accuracy does not always increase with lamda. This interesting finding indicates it is not always good to choose a high dimensional project space, let alone the cons of dealing with larger dataset.

To find the impact of k in k-NN classifier, different value of k is applied. The classifier is PCA(0.8)+k-NN(k). A common choice of k is sqrt(# boservations per class). Values around the common choice are tested. From the table below, k value has very limited impact to the accuracy.

k	**13**	**14**	**15**	**16**	**17**	**18**
Accuracy	0.8550	0.8600	0.8450	0.8620	0.8670	0.8630

When the k is small, a more complicated model will help to increase the training accuracy but the model is not robust and prone to overfit. That's why the test accuracy doesn't increase with a small k. We fix the k value to be 17 in the following experiment in this section since 17 gives the best result.

To compare the Bayes, k-NN's combination with PCA and LDA, we do the following experiments.

Classifier		Accuracy (75% training 25%testing)
Bayes	LDA	0.8510
	PCA (0.8)	0.882
K-NN(17)	LDA	0.8650
	PCA (0.8)	0.8670

The table above shows the accuracy of four combinations of classifiers with fix parameter in PCA and k-NN. PCA does a slightly better job than LDA. Usually PCA is better in cases where number of samples per class is less. In this case, the number of classes is only 2. LDA works better in large dataset especially having multiple classes. Although in the lecture we analyze the Bayes is better than k-NN, but from the table above, k-NN is as good as Bayes. They even achieves the exact same average accuracy!

In Kernel-SVM, experiments are done to find the impact of σ^2 in RBF kernel, the impact of r in polynomial kernel, and the impact of C in SVM.

σ^2	**1**	**10**	**100**	**1000**
Accuracy	0.485	0.525	0.860	0.780

From the table above, we find the optimal σ^2 is around 100. Please note c is fixed to 0.01.

r	**1**	**0.1**	**0.075**	**0.05**
Accuracy	0.665	0.527	0.863	0.783

From the table above, we find the optimal r is about 0.075.

c	**0.001**	**0.01**	**0.1**	**1**
Accuracy	0.705	0.853	0.708	0.719

The optimal c from above is 0.01. C has an optimal value because a larger or smaller value is not the best. Larger c allows less training error and smaller c allows more by changing the size of margin hyperplane. Large plane causes more mismatch in testing while small plane might result in overfitting.

T	**10**	**20**	**25**	**30**
Accuracy	0.6440	0.6900	0.6920	0.6160

The optimal T from table above is 25. mu is fixed to 0.8.

mu	0.5	0.6	0.7	0.8	0.9
Accuracy	0.5	0.705	0.7425	0.6960	0.6940

The optimal mu is 0.8, while T is fixed 25. In the experiment above, accuracy increases as mu increases because the mu decides the portion of training data in weak SVM. Larger mu makes more data to be in training set to make weak SVM stronger.

In the experiments above, the training data percentage is fixed 75%. We will change the percentage below to see what happens.

Training percentage	50	60	70	80	90
Accuracy	0.863	0.852	0.871	0.897	0.905

From the table above, as the training percentage increases, the accuracy increases. It's probably because more training data are feed in the classifier to make it more accurate.

Subject label Classification

In this experiment, dataset data.mat are used and split into training and testing dataset. Each individual subject has three cases which are neutral, expression and illumination. These three cases could be used as training or testing data. In our experiment, one of them is used as testing and two remaining are training dataset. K-fold (k=3) approach are applied to evaluate the accuracy.

Training data	testing data	PCA(0.8)+kNN(1)	PCA(0.8)+Bayes	LDA+Bayes	LDA+kNN(1)
Remainings	illumination	0.6	0.6	0.605	0.665
	Expression	0.655	0.62	0.62	0.725
	Neutral	0.645	0.525	0.715	0.835
Average		0.633	0.582	0.647	0.742

From the table above, we see the LDA+kNN(1) does the best job. kNN is also better than Bayes in this case. Also the different combinations of training and testing data have an impact on the accuracy. Doing multiple experiments and taking average is very necessary in this case.

Besides, we also did some experiments using other datasets. Training and testing dataset are not randomly split. Training percentage is 70% while the testing is 30%.

Dataset	PCA(0.8)+kNN(1)	PCA(0.8)+Bayes	LDA+Bayes	LDA+kNN(1)
Illumination	0.846	0.975	1	1
pose	0.808	0.672	0.756	0.743

From the table above, we find LDA is better than PCA in these two datasets. kNN and Bayes really depends on dataset.