Human-centered Assistive Robotics Technische Universität München Prof. Dr.-Ing. Dongheui Lee	**MACHINE LEARNING IN ROBOTICS** Assignment 1

Exercise 2: *Handwritten digits classification using Baysian classifier*

MNIST database contains images of handwritten digits. It has a training set of $60,000$ examples and a test set of $10,000$ examples. The digits have been centered and size-normalized to 28×28 pixels. The files "train-images.idx3-ubyte" and "train-labels.idx1-ubyte" contain the training images and the corresponding labels. The labels are from 0 to 9 which are the identity of the images. The files "t10k-images.idx3-ubyte" and "t10k-labels.idx1-ubyte" contain the test images and their corresponding labels, which you will use for the evaluation of the learned classifier. Load the data using the provided functions $loadMNISTImages$ and $loadMNISTLabels$.

$images = loadMNISTImages('train-images-idx3-ubyte');$
$labels = loadMNISTLabels('train-labels-idx1-ubyte');$

Reading the training images will yield a matrix of size $784 \times 60,000$ where $60,000$ is the number of images while reading the labels file will produce a vector of size $60,000 \times 1$. The i^{th} image in the dataset can be easily visualized by $imshow(reshape(images(:,i),28,28))$. Now you will model the distribution of each digit using a multivariate Gaussian distribution.

Since the data is already in high dimensional space, you will first project the data into a lower dimensional space, to avoid the singularity problem associated with density estimation in high dimensional space. You will use Principal component analysis (PCA) to find linear orthogonal basis that preserve the maximum variance in the data. To find the PCA basis:

1. Make the training data zero mean, by subtracting mean of the images from all training images.
2. Use matlab command cov to calculate the covariance matrix of the zero-mean data.
3. Calculate eigenvalues and eigenvectors of the covariance matrix by using matlab command eig.
4. Now the d principal components of the data are the d eigenvectors with highest eigenvalues.
5. Project the data on these basis

Now you have the reduced dimensional data. As mentioned earlier, we will model each class by using a multivariate Gaussian distribution. Now calculate the mean and covariance of each digit class separately.

Figure 1: Some sample images from MNIST dataset.

You can use the matlab functions $mean$ and cov for learning the means and covariances.

For a novel test input:

1. Subtract the mean vector of the training data from it.
2. After mean subtraction, project it on the already learned basis.
3. Calculate the likelihood value of the projected data for each class.

   $$p(x|class_j) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_j)' \Sigma_j^{-1}(x-\mu_j)}$$

   where $|\ |$ and $'$ represent the determinant and transpose respectively. You can use the matlab function $mvnpdf$ for calculating the likelihood value. The model parameters μ_j and Σ_j are the model parameters which you learned for the j^{th} class.
4. Assosiate the input to the class yielding highest likelihood value.

Now apply the maximum likelihood classifier on the **test data** by changing the value of d from 1 to 60. Attach the plot of classification error when varying d from 1 to 60. Report the optimal value of d producing smallest percentage of misclassified images and its corresponding classification error. If more than one d value give the lowest error then report the one with smallest d value. You can generate the confusion matrix with matlab command $confusionmat$ and plot it with $helperDisplayConfusionMatrix$. Also include the confusion matrix for optimal d value in the report. For the sake of verification, the classification error for $d = 15$ is 7.03. You can compare your results with some well know classifiers for this dataset at http://yann.lecun.com/exdb/mnist/.

Note: For getting a better insight about PCA, you can plot mean image and top few eigenvectors of a class. Additionally, you can also experiment with reconstructing an image from top few eigenvectors. This part is not evaluated, it is just for your better understanding of PCA. Please do not include these results in the submission.

Solution 2: *Handwritten digits classification using Baysian classifier*

Please check the attached code (main.m). The comments of each cell describe what each code does.

After successfully running a code, the optimal value of number of principal components would be 48, and a confusion matrix as below will be obtained.

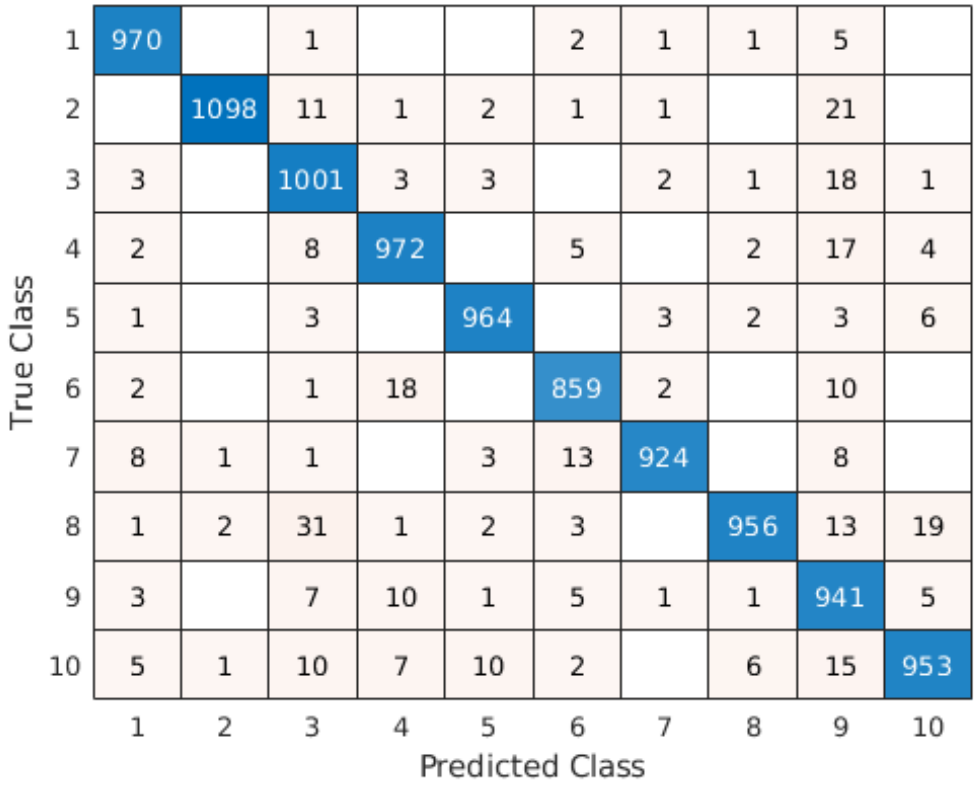

Figure 2: Obtained Confusion Matrix.