MACHINE LEARNING FROM DATA
Fall 2018

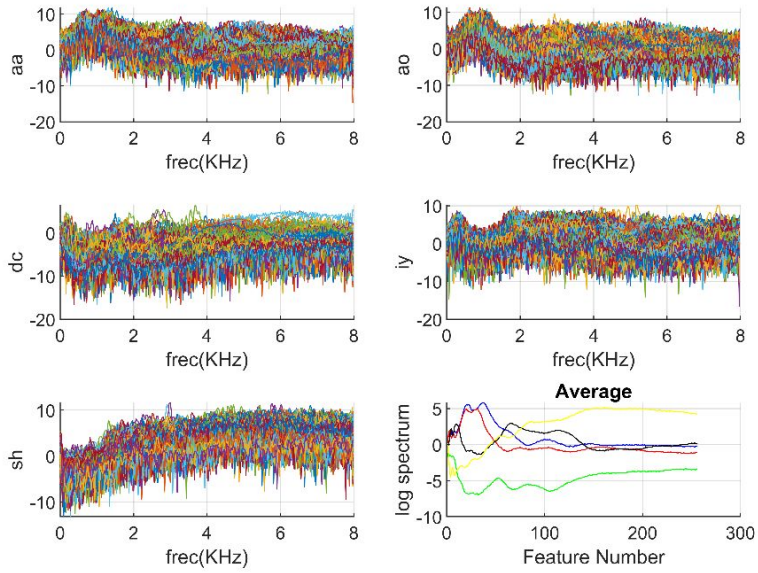
Names: Santagiustina Francesco, Simonetto Adriano
Group:

1. Instructions

- Answer the questions
- Save the report and upload the file

2. Questions

Q1. Include the plots of the phoneme spectra.



Q2. Include the error probabilities for the training and test sets obtained with the linear classifier (LC) and the quadratic classifier (QC), using all the features. Discuss the results.

The probabilities of error are as follows:

P_e	Training set	Test set
Linear	0.0736	0.0885
Quadratic	0.0499	0.1055

As it can be expected, the quadratic classifier outperforms the linear one on the training set. Less intuitively it behaves worse in the train set, which is probably a sign of overfitting: we chose too many features and the parameters became too accustomed with the training set, determining a bad generalization on the test set.

Q3. Include the confusion matrices for the test set obtained with the linear classifier (LC) and the quadratic classifier (QC), using all the features. Discuss the results.

$$CM_l = \begin{bmatrix} 134 & 47 & 0 & 0 & 0 \\ 43 & 223 & 0 & 1 & 0 \\ 0 & 0 & 188 & 5 & 4 \\ 0 & 0 & 1 & 302 & 0 \\ 0 & 0 & 0 & 0 & 227 \end{bmatrix} \quad CM_q = \begin{bmatrix} 127 & 53 & 0 & 0 & 1 \\ 51 & 211 & 1 & 1 & 3 \\ 0 & 0 & 191 & 5 & 1 \\ 0 & 0 & 9 & 291 & 3 \\ 0 & 0 & 0 & 0 & 227 \end{bmatrix}$$

According to both confusion matrices, classes 3, 4 and 5 (corresponding respectively to 'dcl', 'iy' and 'sh') seem to be quite well separated from all the others, with only a handful of misclassifications. On the other hand discriminating between classes 1 and 2 ('aa' and 'ao') seems to be a much harder task as a considerable amount of the elements ends up in the wrong class. More precisely it appears that a misclassification of the kind 1-2 affects almost a third of all elements of class 1, while one of the kind 2-1 happens more or less one fifth of the times. That said, the two classes still maintain a good separation from the other three.

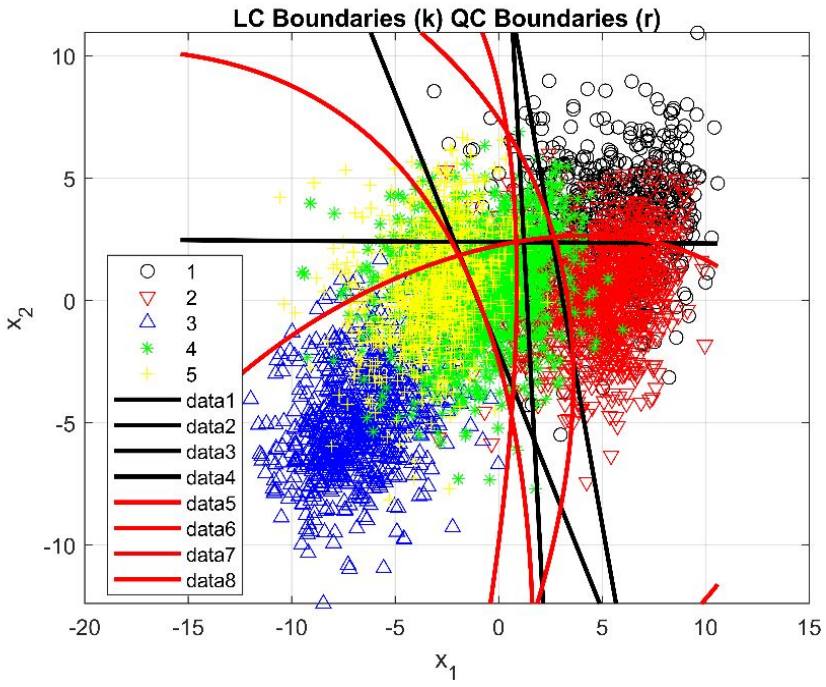
Q4. Which features would you choose? Show the error probabilities for the training and test sets obtained with the linear and the quadratic classifier. Compare with the previous case (using all features) and discuss the results.

We chose features 19 and 50 as 50 seems to be one of the best choices if we want to discriminate between classes 1 and 2, and same for 19 but for the three others. (In order to confirm that this was actually the case we made another attempt using 'bad' features as 1 and 2. What we saw was that all the error probabilities reached values that were twice the ones in the table below.)

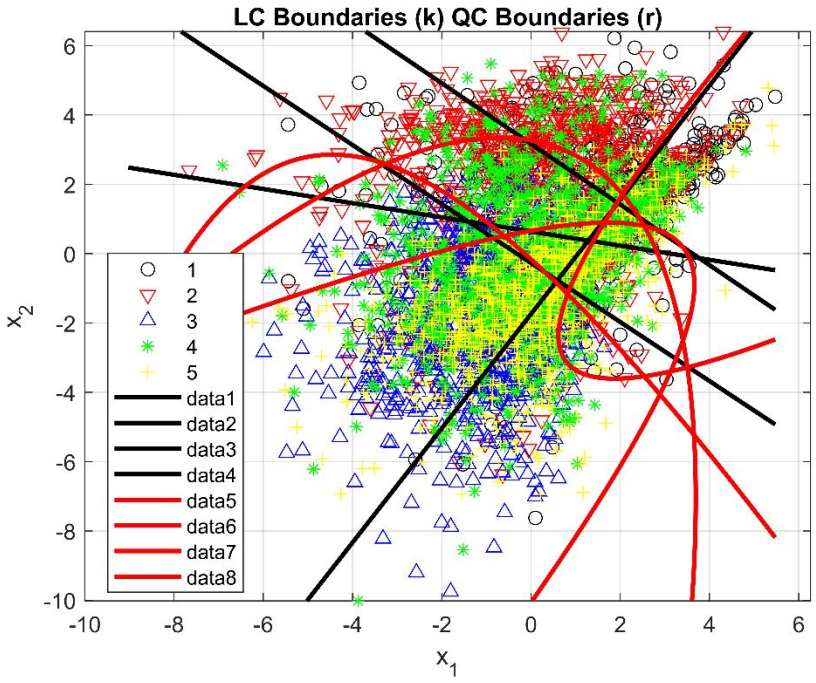
P_e	Training set	Test set
Linear	0.3107	0.3166
Quadratic	0.3024	0.3013

The first thing that we can notice is that all the error probabilities are much higher than the previous case. Of course this is not surprising as we are adopting a small subset of the total set of feature employed before for the classification and we are therefore getting a worse result.
Another point is that this time, in both cases, the errors that we get in the test set are much similar to those in the training set. This is a really reasonable result as the number of features employed is 2, and the amount of data we are using is big enough to make the model generalize well.
At last we can also see that even though the quadratic classifier still shows a better performance than the one of the linear, this time we have a less relevant improvement. Again this can be linked to the fact that we are using only two features, which are not enough to get a much better performance from the quadratic classifier.

Q5. Include the scatter plot and decision boundaries obtained between class ‘aa’ and all the other (four) classes. Discuss the results.



As we can see in the plot above, class the two chosen features manage to discriminate pretty well class three from the rest. They do instead a much worse job with classes 4 and 5 that this time end up mixed up together, and same can be said about classes 1 and 2 (that however showed the same behaviour even when all features were present).
As for the boundaries we can for example observe the ones between classes one and 2 to see that the two classes are so mixed together that employing a quadratic classifier instead of a linear one does not do much of a difference. A better (not that much) performance is instead achieved by the quadratic classifier between classes 1 and 4.
The plot below is the scatter plot obtained for features 1 and 2 where we can see how ‘bad’ features behave in the same environment: the features are not discriminative enough and all classes appear mixed with the others.



Q6: Complete the table with the training and test errors for the linear (LC) and the quadratic (QC) classifiers when using three, two and one feature, and SNR=10dB. In this case PCA is used for feature selection. Discuss the results. Analyze the scatter plots in two dimensions and in one dimension.

	3 features		2 features		1 feature	
	Test	Train	Test	Train	Test	Train
LC	0.0080	0.0027	0.0080	0.0027	0.0347	0.0280
QC	0.0027	0.0013	0.0053	0.0020	0.0213	0.0187

As expected, as the number of features grows, the error on the training set tends to diminish. The same happens on the test set as the amount of features considered does not get too high.
Moreover as usual the error tends to be higher on the test set than on the training set.
In practice however there aren't many differences between choosing three or two features in this case as the SNR is really high.
Confronting the 1-D and 2-D scatter plots we can see why the error is higher in the 1-D case: in one dimension classes 2 and 3 tend to overlap, a behaviour that does not appear when using two features.

Q7: Complete the table with the training and test errors for the linear (LC) and the quadratic (QC) classifiers when using three, two and one feature, and SNR=10dB. In this case MDA is used for feature selection. Discuss the results. Analyze the scatter plots in two dimensions and in one dimension.

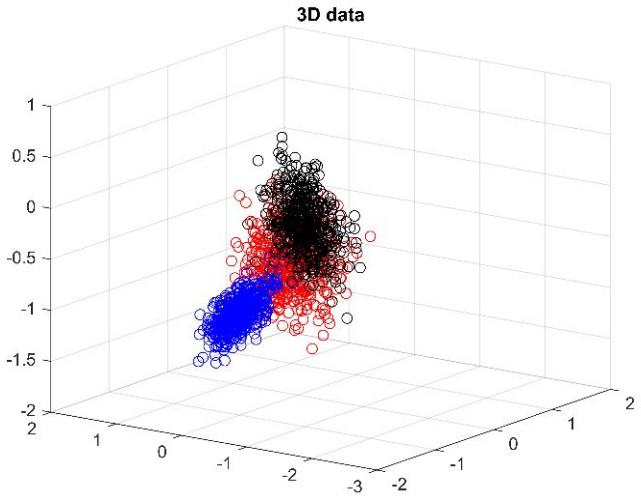
	3 features		2 features		1 feature	
	Test	Train	Test	Train	Test	Train
LC	0.0080	0.0027	0.0080	0.0027	0.072	0.054
QC	0.0027	0.0013	0.0053	0.0020	0.067	0.057

The conclusions that we can get for the error probabilities are almost the same to those of Q6. The only difference that we can spot is that the 1-D probabilities result a bit higher than those in the previous case. The SNR however is really high and it is hard to reach any meaningful conclusion from the facts.

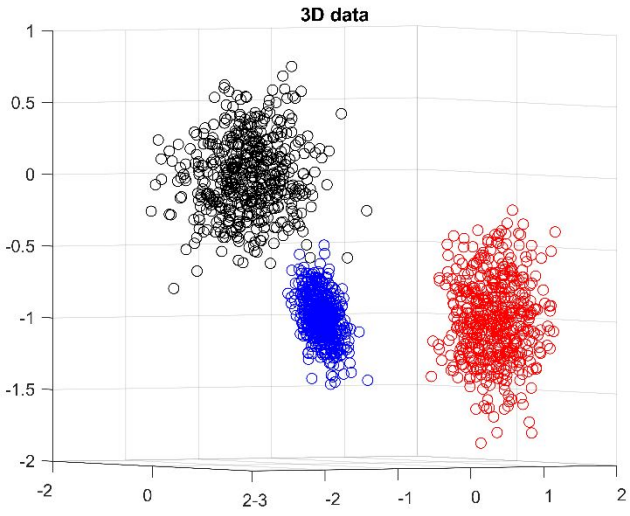
The same can be said for the scatter plots where in the 1-D case classes 2 and 3 still show some overlap, while this does not happen in the 2-D case.

Q8: By rotating the figure observe that there are 2D projections where projected clusters are well separated while there are other projections where projected clusters overlap. Copy a pair of examples illustrating this point

Overlap



Good separation



Q9: Complete the table with the training and test errors for the linear (LC) and the quadratic (QC) classifiers when using three, two and one feature, and SNR= 0 dB. Use PCA for feature selection. Discuss the results. Analyze the scatter plots in two dimensions and in one dimension.

	3 features		2 features		1 feature	
	Test	Train	Test	Train	Test	Train
LC	0.1680	0.1627	0.1680	0.1667	0.2187	0.2133
QC	0.0667	0.0680	0.1653	0.1467	0.2240	0.1933

The error probabilities are of course higher than those of the previous case since we decreased the SNR. This time we can see an improvement in the error probabilities also when going from 2 to 3 features, but this happens only for the quadratic classifier. As the number of features grows, the performance of the quadratic tends to improve more than that of the linear one.
The error probability in test and training sets remains similar, which shows as before how we get a good generalization with a small number of features.
Observing the 1-D scatter plot, we notice that as before classes 2 and 3 tend to overlap, but this time the same also happens for classes 1 and 2. In 2-D the separability is still better than in the 1-D case but also in this case the separability of the classes suffers from the drop in SNR.

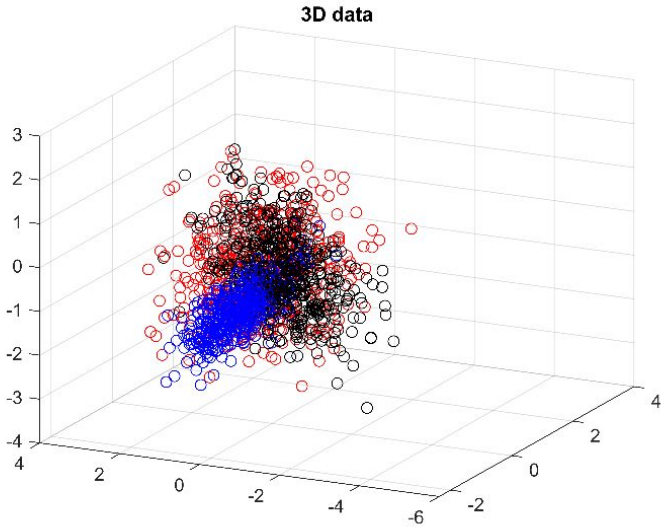
Q10: Complete the table with the training and test errors for the linear (LC) and the quadratic (QC) classifiers when using three, two and one feature, and SNR= 0 dB. Use MDA for feature selection. Discuss the results. Analyze the scatter plots in two dimensions and in one dimension.

	3 features		2 features		1 feature	
	Test	Train	Test	Train	Test	Train
LC	0.1680	0.1627	0.1680	0.1627	0.2747	0.2593
QC	0.0667	0.0680	0.1440	0.1413	0.2640	0.2607

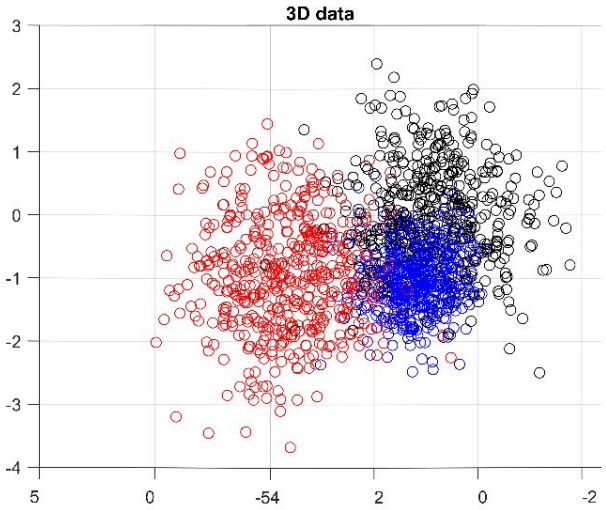
The considerations that we can make are the same of those we made for the PCA. Again we can see a higher error in the 1-D case with respect to PCA while for 2-D and 3-D the results are similar. This is probably due to the chosen seed.
As for the scatter plots, the conclusions are the same to those in Q9.

Q11: observe by rotating the figure that there are 2D projections where projected clusters are well separated while there are other projections where projected clusters overlap. Copy a pair of examples illustrating this point

Overlap



Good separation



Q12. Find and write the three vectors corresponding to the class means. Give also the value of the seed used in your experiments.

The three mean vectors are defined at line 9 of lab2_gengauss_al.m as :

$$\mathbf{m_1} = \mathbf{d} \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{m_2} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \mathbf{m_3} = \mathbf{d} \times \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$$

where *d* = 1 is the distance between means of adjacent classes (intersymbol distance). For our experiments we used 7 as seed.

Q13. Which is the rank of the matrix Sb? How many features can we use with MDA?

At a first glimpse the columns of Sb seem all proportional, but if in mds_ml.m we add a line to store Sb (like: save('Sb.mat','Sb');) and compute its rank with the function rank() we obtain 2. This coincide with the theory which tells us that in MDA we can use a maximum number of features equal to the number of classes minus one. However in this particular case we must note the fact that the three means are aligned. Also, each column of Sb is obtained as a linear combination of the vectors $d_i = m_i - m$ but in this case the vectors would all belong to the same line (and are thus equal up to a multiplicative scalar factor), if it wasn't for the little difference between the mean of the distribution and the actual sample average of the realizations. This situation leads the two non-null eigenvalues of Sb to be of very different magnitudes: the main eigenvalue associated to the direction of the means alignment and another one, over two thousands times smaller, which takes into account of the variability of the realization.

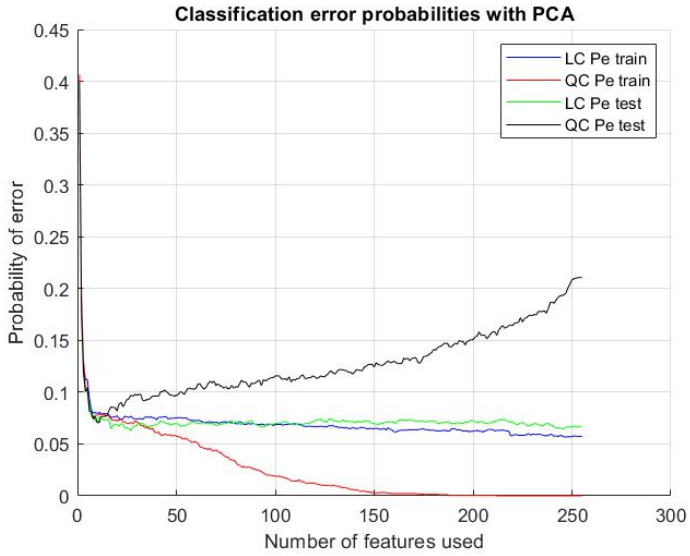
Q14. Complete a table with the training and test errors for the linear (LC) and the quadratic (QC) classifiers when using three, two and one feature, and SNR= -5 dB. Use PCA and MDA for feature selection. Discuss the results. In which cases is MDA clearly better than PCA?

PCA	3 features		2 features		1 feature	
	Test	Train	Test	Train	Test	Train
LC	0	0	0.6853	0.6713	0.6507	0.6613
QC	0	0	0.6880	0.6493	0.6213	0.6713

MDA	3 features		2 features		1 feature	
	Test	Train	Test	Train	Test	Train
LC	0	0	0	0	0	0
QC	0	0	0	0	0	0

These two tables clearly show us that MDA is better to reduce dimensionality in this case. Indeed it allows us to correctly classify all the measurements of both training and test set using 3, 2 and also only 1 feature. On the contrary PCA correctly classify the measurements only when the dimensionality isn't reduced (3 features), but for 2 and 1 features the error probability is close to the one of a random guess (Pe = ⅔). We can explain this result in the following way : the distribution of the measurement is such that, for each class the scatter plot forms an ellipsoid with two long axes perpendicular to a short one in the direction of the other classes**.** Roughly speaking we have three shifted parallel discs and we can guess that the global scatter matrix used by PCA the contribute of the intra-class scatter matrix (Sc) is more important than the contribute of the inter-class scatter matrix (Sb), so it identifies as directions of maximal variation the two associated to to the intra class variance. So once the dimension of the eigenvector associated to the lowest eigenvalue is dropped by passing from 3 to 2 features we don't have anymore the useful information associated to the different initial mean of each class and can't classify the measurements correctly because they are totally overlapped.

Q15. Show the error curves for the linear and the quadratic classifier on the training and on the test set. Copy your code in Annex1



Q16. Discuss which dimension is the most adequate for the linear classifier and which is the best one for the quadratic classifier. Remember that it is important not to overfit on the training data (the test error should not be much larger than the training error).

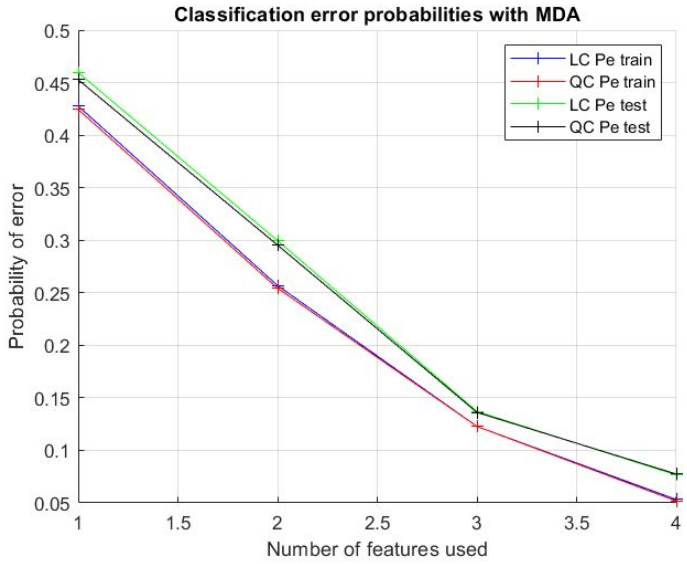
Using PCA we obtain the best results in the test dataset by using 27 features with the linear classifier and 10 features with the quadratic one. But we can see that the test error probability of the LC is almost

constant after having reached a sufficient number of features, so we could keep only 17 features to decrease the dimensionality of the problem without significantly affecting the performances. Better results on the training set for a higher number of used features probably depends on the overfitting of the model to the particular subset of data (the quadratic model being more flexible, overfits more).

Q17. Which is the maximum number of features dmax?

In MDA we can use reduce the number of dimensions to a maximum value of $N_{class} - 1 = 5 - 1 = 4$ features.

Q18. Show the error curves for the linear and the quadratic classifier on the training and on the test set. Copy your code in Annex2



Q19. Compare results and discuss the use of PCA and MDA for the Phoneme dataset

With PCA the minimum error probabilities obtained on the test set are 0.0630 for the LC and 0.0706 for the QC which doesn't introduce any improvement. The MDA obtains its better results with 4 features : an error probability of 0.0766 for the LC and of 0.0774 for the QC on the test set. Even if the best PCA LC test error probability is slightly lower than the MDA one we need to keep in mind that they are not using the same number of features, the test error probabilities of PCA with 4 features are 0.1013 for the LC and 0.1004 for the QC, which is substantially worse than MDA performances.

Annex 1. Matlab code for Q15 (PCA)

```
clear
close all

disp(' ')
disp('Feature selection by PCA dimensionality reduction')


%% Options / Initalitation
V_coor=1:256;                % 256 to take all features set 1:256
```

```
N_feat=length(V_coor);
% class name: Labels:
% 1(aa);2(ao);3(dcl);4(iy);5(sh);
N_classes=5;
N_fft=256;                                    %256   (8KHz)   128   (4KHz),   64   (2KHz),
32(1khZ)
%% Database load
load BD_phoneme

%% MEAN IS REMOVED FROM DATABASE
X=X-ones(length(Labels),1)*mean(X);

%% Database partition
P_train=0.7;
Index_train=[];
Index_test=[];
for i_class=1:N_classes
       index=find(Labels==i_class);
       N_i_class=length(index);
       [I_train,I_test] = dividerand(N_i_class,P_train,1-P_train);
       Index_train=[Index_train;index(I_train)];
       Index_test=[Index_test;index(I_test)];
end
% Train Selection
X_train=X(Index_train,:);
Labels_train=Labels(Index_train);
% Test Selection and mixing
X_test=X(Index_test,:);
Labels_test=Labels(Index_test);
clear Index_train Index_test index i_class N_i_class I_train I_test

%% Feature selection loop
all_LC_Pe_train = [];
all_QC_Pe_train = [];
all_LC_Pe_test = [];
all_QC_Pe_test = [];
all_X_train = X_train;
all_X_test = X_test;
for d= 1: (N_feat-1)
%d
W_fc=pca(all_X_train);
W_fc=W_fc(:,1:d);
X_train=all_X_train*W_fc;
X_test=all_X_test*W_fc;


%% Create a default (linear) discriminant analysis classifier:
linclass = fitcdiscr(X_train,Labels_train,'prior','empirical');
Linear_out = predict(linclass,X_train);
Linear_Pe_train=sum(Labels_train ~= Linear_out)/length(Labels_train);
%fprintf(1,' error Linear train = %g   \n', Linear_Pe_train)
Linear_out = predict(linclass,X_test);
Linear_Pe_test=sum(Labels_test ~= Linear_out)/length(Labels_test);
%fprintf(1,' error Linear test = %g   \n', Linear_Pe_test)

%% Create a quadratic discriminant analysis classifier:
quaclass                                                                              =
fitcdiscr(X_train,Labels_train,'discrimType','quadratic','prior','empirical');
Quadratic_out= predict(quaclass,X_train);
Quadratic_Pe_train=sum(Labels_train ~= Quadratic_out)/length(Labels_train);
%fprintf(1,' error Quadratic train = %g   \n', Quadratic_Pe_train)
```

```
Quadratic_out= predict(quaclass,X_test);
Quadratic_Pe_test=sum(Labels_test ~= Quadratic_out)/length(Labels_test);
%fprintf(1,' error Quadratic test = %g   \n', Quadratic_Pe_test)


%% Store error probabilities for current d'
all_LC_Pe_train = [all_LC_Pe_train; Linear_Pe_train];
all_QC_Pe_train = [all_QC_Pe_train ; Quadratic_Pe_train];
all_LC_Pe_test = [all_LC_Pe_test; Linear_Pe_test];
all_QC_Pe_test = [all_QC_Pe_test ; Quadratic_Pe_test ];

end
%% Plotting error curves

figure();
hold on
plot(all_LC_Pe_train, 'b');
plot(all_QC_Pe_train, 'r');
plot(all_LC_Pe_test, 'g');
plot(all_QC_Pe_test, 'k');
legend({'LC Pe train','QC Pe train','LC Pe test','QC Pe test'});
hold off
grid
zoom on
xlabel('Number of features used')
ylabel('Probability of error')
title('Classification error probabilities with PCA');
```

Annex 2. Matlab code for Q18 (MDA)

```
clear
close all

disp(' ')
disp('Feature selection by MDA dimensionality reduction')


%% Options / Initalitation
V_coor=1:256;                % 256 to take all features set 1:256

N_feat=length(V_coor);
% class name: Labels:
% 1(aa);2(ao);3(dcl);4(iy);5(sh);
N_classes=5;
N_fft=256;                                    %256   (8KHz)   128   (4KHz),   64   (2KHz),
32(1khZ)
%% Database load
load BD_phoneme

%% MEAN IS REMOVED FROM DATABASE
X=X-ones(length(Labels),1)*mean(X);

%% Database partition
P_train=0.7;
Index_train=[];
Index_test=[];
for i_class=1:N_classes
```

```
        index=find(Labels==i_class);
        N_i_class=length(index);
        [I_train,I_test] = dividerand(N_i_class,P_train,1-P_train);
        Index_train=[Index_train;index(I_train)];
        Index_test=[Index_test;index(I_test)];
end
% Train Selection
X_train=X(Index_train,:);
Labels_train=Labels(Index_train);
% Test Selection and mixing
X_test=X(Index_test,:);
Labels_test=Labels(Index_test);
clear Index_train Index_test index i_class N_i_class I_train I_test

%% Feature selection loop
all_LC_Pe_train = [];
all_QC_Pe_train = [];
all_LC_Pe_test = [];
all_QC_Pe_test = [];
all_X_train = X_train;
all_X_test = X_test;
for d= 1: (N_classes-1)
%d
W_fc=mda_ml(all_X_train,Labels_train,N_classes);
W_fc=W_fc(:,1:d);
X_train=all_X_train*W_fc;
X_test=all_X_test*W_fc;


%% Create a default (linear) discriminant analysis classifier:
linclass = fitcdiscr(X_train,Labels_train,'prior','empirical');
Linear_out = predict(linclass,X_train);
Linear_Pe_train=sum(Labels_train ~= Linear_out)/length(Labels_train);
%fprintf(1,' error Linear train = %g   \n', Linear_Pe_train)
Linear_out = predict(linclass,X_test);
Linear_Pe_test=sum(Labels_test ~= Linear_out)/length(Labels_test);
%fprintf(1,' error Linear test = %g   \n', Linear_Pe_test)

%% Create a quadratic discriminant analysis classifier:
quaclass                                                                                    =
fitcdiscr(X_train,Labels_train,'discrimType','quadratic','prior','empirical');
Quadratic_out= predict(quaclass,X_train);
Quadratic_Pe_train=sum(Labels_train ~= Quadratic_out)/length(Labels_train);
%fprintf(1,' error Quadratic train = %g   \n', Quadratic_Pe_train)
Quadratic_out= predict(quaclass,X_test);
Quadratic_Pe_test=sum(Labels_test ~= Quadratic_out)/length(Labels_test);
%fprintf(1,' error Quadratic test = %g   \n', Quadratic_Pe_test)


%% Store error probabilities for current d'
all_LC_Pe_train = [all_LC_Pe_train; Linear_Pe_train];
all_QC_Pe_train = [all_QC_Pe_train ; Quadratic_Pe_train];
all_LC_Pe_test = [all_LC_Pe_test; Linear_Pe_test];
all_QC_Pe_test = [all_QC_Pe_test ; Quadratic_Pe_test ];

end
%% Plotting error curves

figure();
hold on
plot(all_LC_Pe_train, 'b*');
plot(all_QC_Pe_train, 'r*');
```

```
plot(all_LC_Pe_test, 'g*');
plot(all_QC_Pe_test, 'k*');
legend({'LC Pe train','QC Pe train','LC Pe test','QC Pe test'});
hold off
grid
zoom on
xlabel('Number of features used')
ylabel('Probability of error')
title('Classification error probabilities with MDA');
```