Practice 3 – Classification Methods

Pattern Recognition

1. Introduction

In the last two lab sessions you have seen different strategies for a first approach to pattern recognition and to dimensionality reduction, losing as little information as possible from the original data.

In this practice, we are going to apply some concepts and techniques already seen in the previous sessions, together with new ones to build a model that learns from the data. Given a new sample, this model should be able to classify it by assigning one of the different emotions.

2. Database and code

In this practice you will work with the same databases as in the previous ones. Most of the files used in this this practice have already been used in the previous ones. Here we present the two new scripts that you are going to work with:

p3Classification.m: It is the main script. It loads the data, and divide it in groups to perform crossvalidation. It also calls the function applyMethods.m to apply the desired methods.

ApplyMethods.m: In this function you will have to code the dimensionality reduction techniques and the models to make the predictions.

3. Work to do

In this lab you will use what we learned in the previous section to reduce the dimensionality to make the dataset more treatable, and you will build models to perform the classification task. As in the previous lab, you will have to deliver a **report** explaining what have you done, why have you done it, and comment the results obtained.

In concrete, you should use PCA, LDA and kernel PCA, to reduce the dimensionality of the data, and SVM, kernel SVM, and the Mahalanobis distance to classify the data. You have to test different combinations of *dimensionality reduction technique* and *classification* method, and compare their accuracies. To do so, you can follow the following steps:

- Use SVM to classify the data . You have to take into account that SVM is a binary classifier, so you have to implement an strategy to classify the n classes. You can search on the internet which methods are used in this cases and implement one of them.

- Do the same as in the previous exercise but reducing the dimensionality of the data (with PCA and LDA) to see if it actually improves the performance of the SVM. Try also to use the Mahalanobis distance as a classification method.

- Repeat the last exercise but using kernel PCA. Try different kernels.

- Finally, try to classify the data using kernel SVM. You can previously reduce the dimensionality of the data using PCA and LDA.

To make it easier to show and comment your results, build a table like the next one and fill it with the obtained accuracies:

	Raw data	PCA (reducing to N dimensions)	LDA (reducing to N dimensions)	Kernel PCA (gaussian kernel)	Kernel PCA (polinomial kernel)	. . .
Mahalanobis						
SVM						
Kernel SVM (gaussian kernel)						
Kernel SVM (polinomial kernel)						
. . .						

BONUS: Think a procedure to automatically find the best combination of *dimensionality reduction technique* and *classification method* and their corresponding parameters such as the number of reduced dimensions or the kernel parameters. Explain it.