Applied Machine Learning
Report on Practice Assignment

Team 34

Fall Semester 2017

1 Introduction

2 Datasets

The initial dataset was composed of 4 classes with 12 images per class. Figure 1 shows the composition of the different classes, that were chosen based on their varied geometrical and chromatic features: Watches are dark and have a round shape, bananas are long and yellow, chairs have angular features and pens are long and thin. This representative features are summarized in Table 1.

(a) Banana (b) Chair (c) Pen (d) Watch

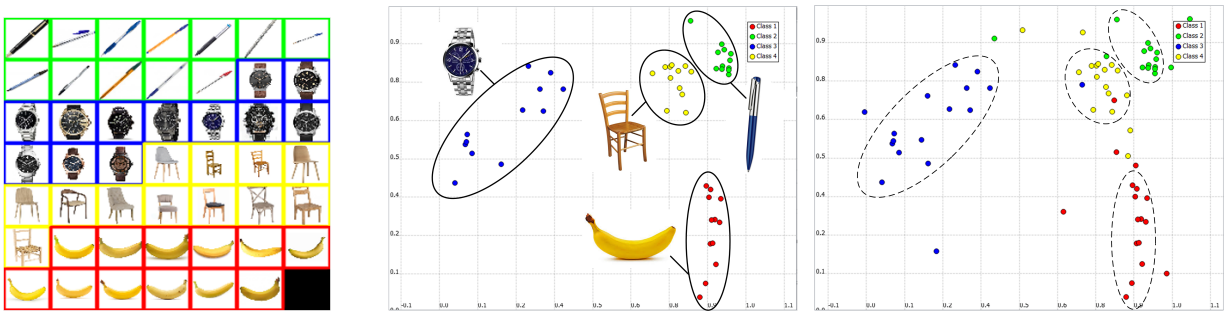
Figure 1: Four different classes

The pictures are normalized when imported in MLDemos to have the same size. The initial dataset (see Figure 2a) is stored in a single $336 \cdot 336$ image constituted of a 7 by 7 matrix of images. Thus, each image has a size of $48 \cdot 48$ pixels and a number of features of $48 \times 48 \times 3$.

Classes	**Shape**	**Colour**	**Number of images**	**Size in pixels**
Banana	Long	Yellow	12	$48 \cdot 48$
Pens	Long and thin	Multicoloured	12	$48 \cdot 48$
Watches	Round	Dark	12	$48 \cdot 48$
Chairs	Angular	Light	12	$48 \cdot 48$

Table 1: Dataset characteristics

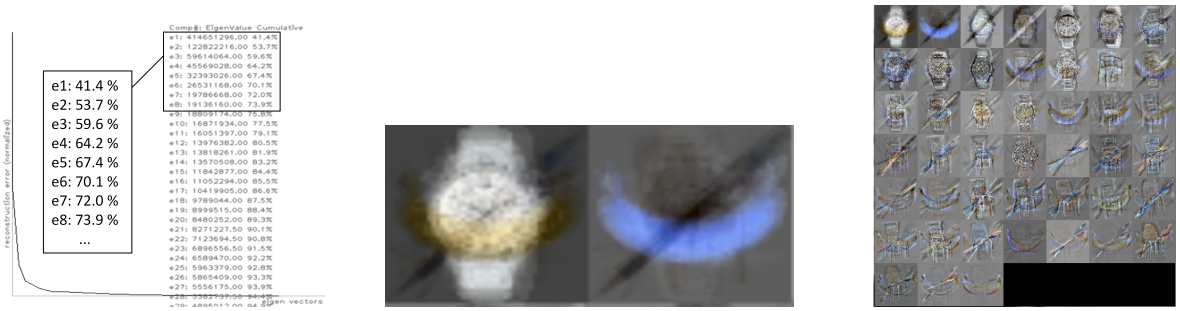
An effort was made to have a linear separable dataset in the beginning for learning purposes. In order to obtain that, the images were purposely chosen to have similar orientations in space. For the second part of this report the same dataset was used, but some noisy datapoints were added (4 per class). This new dataset can be seen in Figure 2c.



(a) Pictures in dataset 1 (b) Projected datapoints of dataset 1 (c) Projected datapoints of "Hacked" dataset 2

Figure 2: Comparison of the datasets used for the report

3 Dimensionality Reduction



(a) Cumulative eigenvalues (b) First two eigenvectors (c) All eigenvectors

Figure 3: First PCA

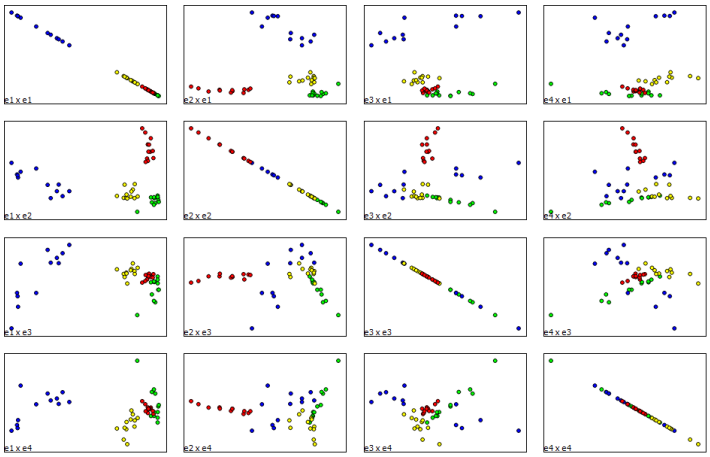


Figure 4: PCA projection matrix on eigenvectors 1 to 4

PCA was applied to the dataset 1. Figure 3a shows the cumulative reconstruction error: it can be noticed that 73.9% of the dataset can be reconstructed with the first 8 eigenvectors. It can be seen in Figure 4 that the projection onto e1-e2 was the optimal projection for clustering purposes as expected: indeed they represent most of the variance of the data. The first eigenvector corresponds to a white watch, a yellow banana and the shape of a pen in the background. On the other hand the second eigenvector shows a blue banana, a pen and a shape of a chair in the background. As such, eigenvector 1 will separate well the watches class from the rest, while the second one will separate better the banana class from the rest. The two principal eigenvectors are shown in Figure 3b while the first 45 eigenvectors can be seen in Figure 3c.

4 Clustering - qualitative assessment

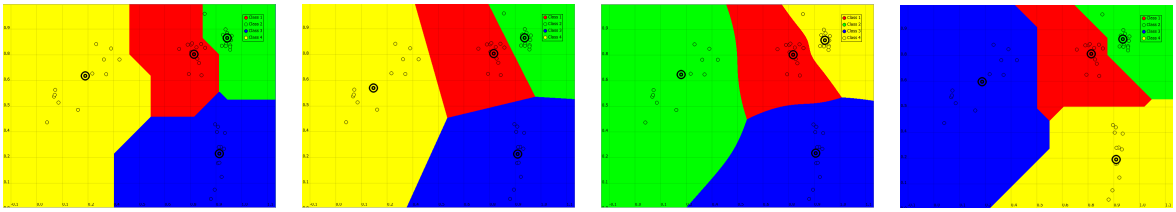
In order to assess the functionalities of unsupervised clustering methods, the dataset 1 was used (see Figure 2a). In particular, the algorithms k-Means, Soft k-Means and DBScan were applied.

4.1 K-means

The hyperparameters for the K-means method are:

- the number of clusters;
- the metric (L1, L2, Linf, Lp).

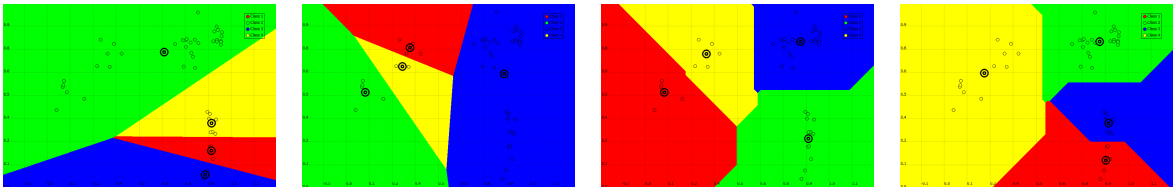
Since clustering does not look at the label of each point, the separation of data points is done according to the position of the points in the PCA projection. For dataset 1, the classes corresponding to chairs and pens are quite close in the PCA projection (see Figure 2b). As such, when the cluster number is smaller than the total number of classes as in Figure 7a, chairs and pens are left in the same class.



(a) Metric L1 (b) Metric L2 (c) Metric Lp with p=3 (d) Metric Linf

Figure 5: Dataset 1 with correct K-Means clustering, different metrics and K=4

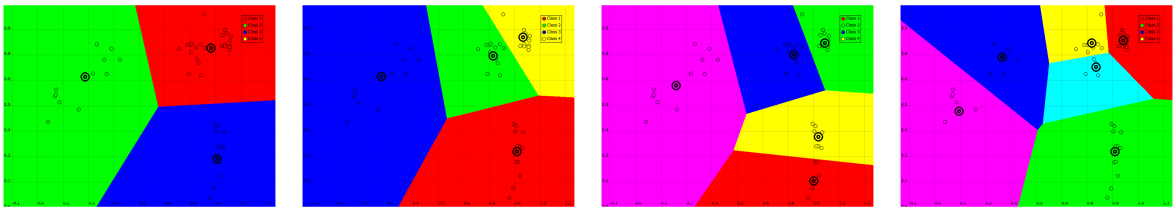
In Figure 5, correct results are shown for K-Means applied for K=4 and different metrics. Although all metrics were able to correctly separate the classes, some errors were typically shown, as can be seen in Figure 6. These errors can be mainly due to the random initialisation of K-Means and the fact that the classes do not have similar distributions, which is the condition K-Mean works best for.



(a) Metric L2 (b) Metric L2 (c) Metric Linf (d) Metric Linf

Figure 6: Dataset 1 with incorrect K-Means clustering, metrics L2 and Linf and K=4

Finally, K-Means behaviour was tested for different values of K (see Figure 7).



(a) K = 3 (b) K = 4 (c) K = 5 (d) K = 6

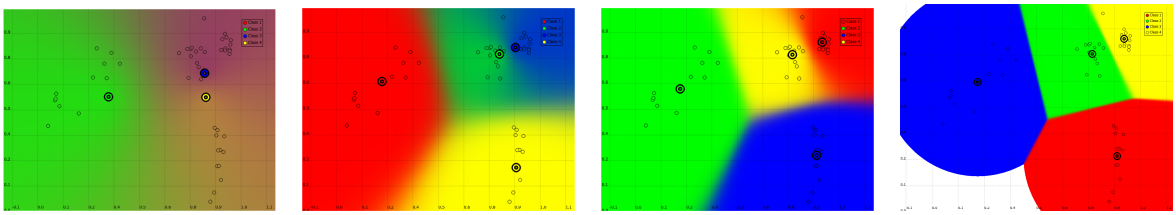
Figure 7: Dataset 1 with K-Means clustering, Euclidean metrics and different value of K

4.2 Soft K-Means

The hyperparameters for the soft K-means method are:

- the number of clusters;
- the stiffness β.

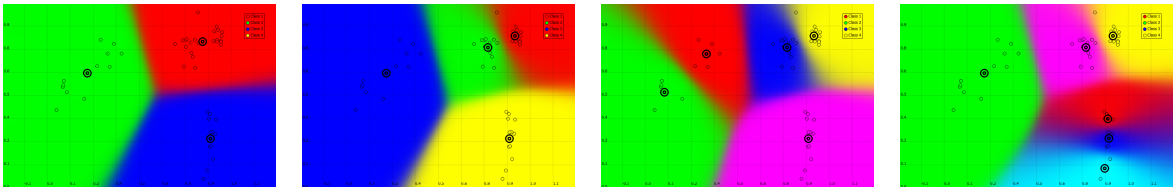
In Figure 8, results for K=4 and different values of β are shown. The higher the value of β, the more similar soft K-Means becomes to hard K-Means. When the stiffness is too low, the classes are difficult to distinguish. From $\beta = 30$, the 4 classes are clearly distinguished and correctly clustered.



(a) $\beta = 5$ (b) $\beta = 15$ (c) $\beta = 30$ (d) $\beta = 200$

Figure 8: Dataset 1 with soft K-Means clustering, K = 4 and different value of β

It can be noticed that the results for soft K-Means when varying the K don't differ considerably from what obtained with hard K-Means, as shown in Figure 9.

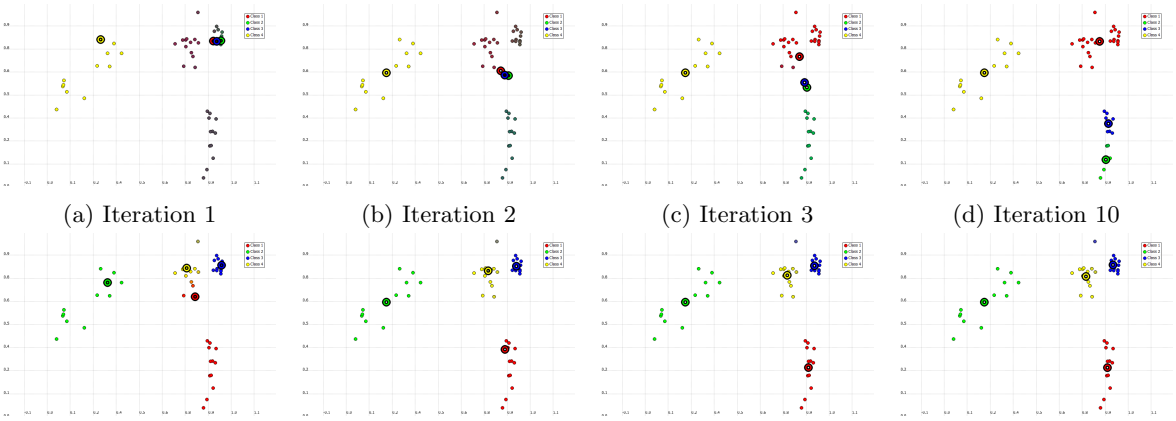


(a) K=3 (b) K=4 (c) K=5 (d) K=6

Figure 9: Dataset 1 with soft K-Means clustering,$\beta = 30$ and different values of K

Initialization

The methods above apply a random initialisation of the cluster centers. The effect of this random procedure can be observed in Figure 10. In particular, Figures 10a to 10d show an initialization that results in an incorrect clustering, while Figures 10e to 10h show a correct clustering in the end.



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 10

(e) Iteration 1 (f) Iteration 2 (g) Iteration 3 (h) Iteration 6

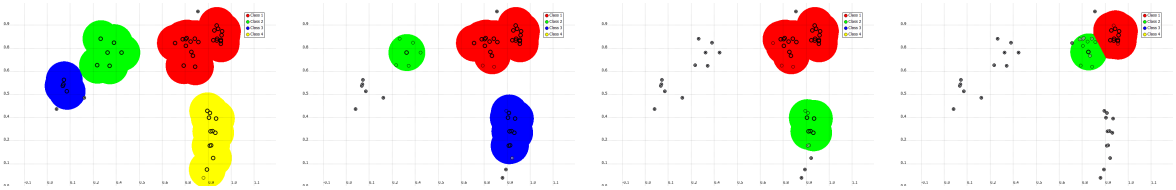
Figure 10: Dataset 1 with soft K-Means clustering for K=4 and $\beta = 30$, showing effects of random initialization

4.3 DBSCAN

The hyperparameters for the DBScan method are:

- the minimum number of points in a cluster MinPts;
- the size of neighborhood ϵ.

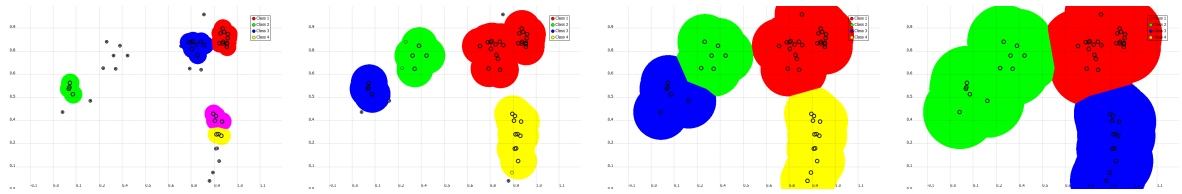
It can be noticed that when the number of Minpoints increases (see Figure 11), the smaller clusters are lost, and more and more points are considered outliers as the ϵ doesn't change.



(a) $Minpoints = 2$ (b) $Minpoints = 4$ (c) $Minpoints = 6$ (d) $Minpoints = 10$

Figure 11: Dataset 1 with DBSCAN clustering, $\epsilon = 0.1$ and different value of $Minpoints$

The smaller the ϵ,



(a) $\epsilon = 0.05$ (b) $\epsilon = 0.1$ (c) $\epsilon = 0.15$ (d) $\epsilon = 0.2$

Figure 12: Dataset 1 with DBSCAN clustering, $Minpoints = 2$ and different value of ϵ

4.4 Type of metric

4.5 Initialization

5 Classification

5.1 Methods comparison

The purpose of this section is to compare the GMM and SVM method with the "hacked" dataset (i.e. **Dataset 2**). Several factors can influence the performance of the GMM and C-SVM classification methods. Since there are many factors to take into account, a comparison plan is fixed in figure (13).The first parameters to set are the test/ratio train and the number of folds for crossvalidation because these parameters directly influences the results of classifications of each method.

Train/test ratio

We noticed that if we take a small ratio train/test there is not enough point to train the algorithm.If the ratio is too high, there is not enough test point and the variance becomes too high. A good trade-off is actually the ratio 2/3 training, 1/3 testing.

Number of folds sensitivity

In MLStudio, the crossvalidation method is based on a random separation of the data between Train and test before repeating the classification and averaging the results. With this type of cross validation a sufficiently large number of folds is recommended to obtain consistent results with lower variance. However, too much fold is not useful because after a while the results do not improve but the cost of calculation increase. A good trade-off for our dataset was 30 folds because it was at this point that the results started to stabilize between several cross validation process.

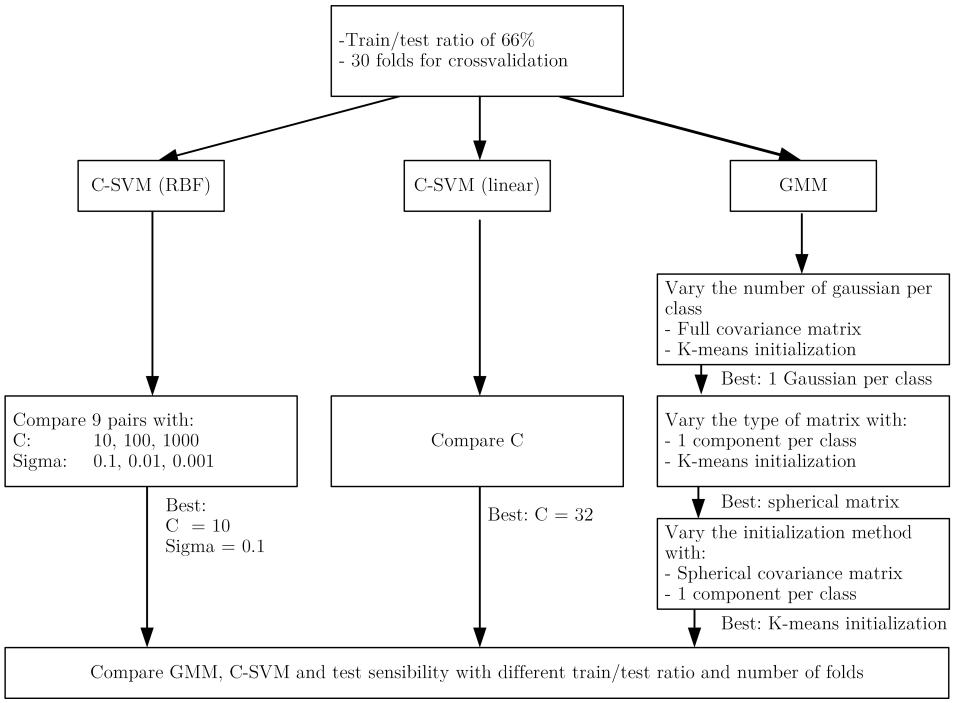


Figure 13: Comparison method block diagram

5.2 GMM classification

For the GMM classification the hyperparameters were found in three stages. First, the K-Means initialization mode is set because it is the mode expected to be more efficient than the random and uniform method. A full covariance matrix is set to allow as much freedom as possible. In the figure 14 a comparison with the number of components per class based on the F-Measure allows to choose the number of GMMs per class.

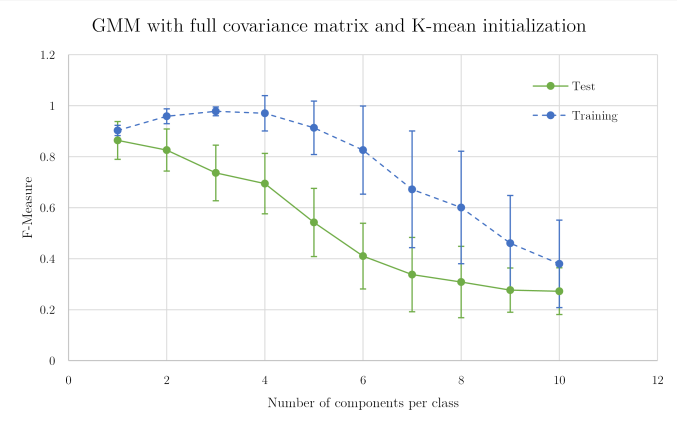
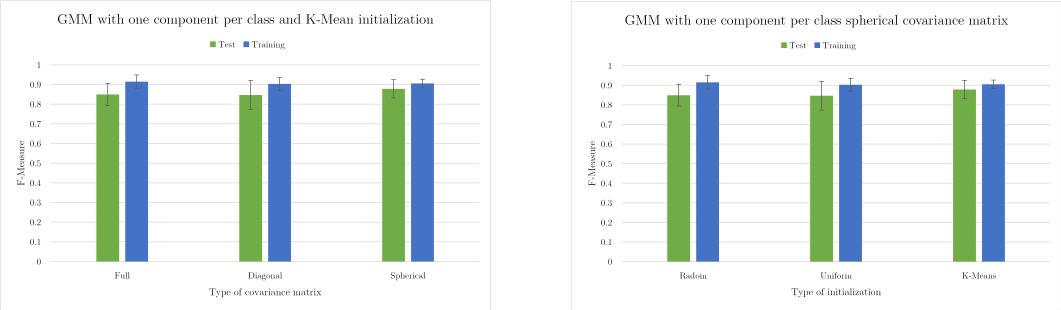


Figure 14: GMM: comparison for different number of components per class

This graph shows that only one GMM per class is the most favorable for classifying the dataset. The training set shows better results with more GMMs per class but the classification of the testing set gets worse and the variance is higher.

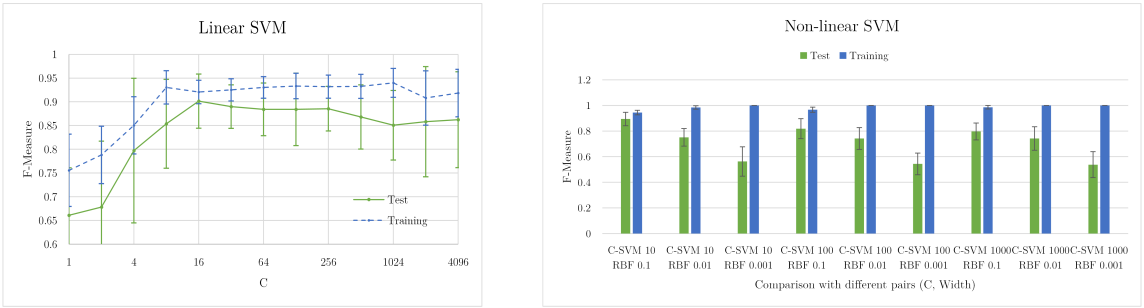


(a) GMM: comparison for different type of matrix (b) GMM: comparison for different initialization

Figure 15: F-measure comparison for different type of matrix and initialization

By following the procedure explained in the block diagram 13 the plots 15b allow us to justify our choices. A spherical covariance matrix and a K-Means initialization method are chosen to favor the smallest variance on the training set. Indeed a large variance on the testing set indicates an overfiting problem of the training set.

5.3 C-SVM classification



(a) Linear C-SVM: comparison for different penalty factor C (b) C-SVM with RBF kernel: comparison for different pairs of hyperparameters

Figure 16: F-measure comparison for different type of matrix and initialization

The only hyperparameter that must be choosen for the linear SVM classification method is the weights the penalty term (i.e. the "C"). The goal is to find the best tradeoff between minimizing the classification errors and maximizing the margin with the helps of the plot 16a. To have a lower variance on the testing set classification a C of 32 is chosen. Lower variance is preferable even though the average of the F-Measure is a little smaller to minimize the overfitting of the training set. For the non-linear SVM classification some values of *sigma* and C are chosen and all combinations

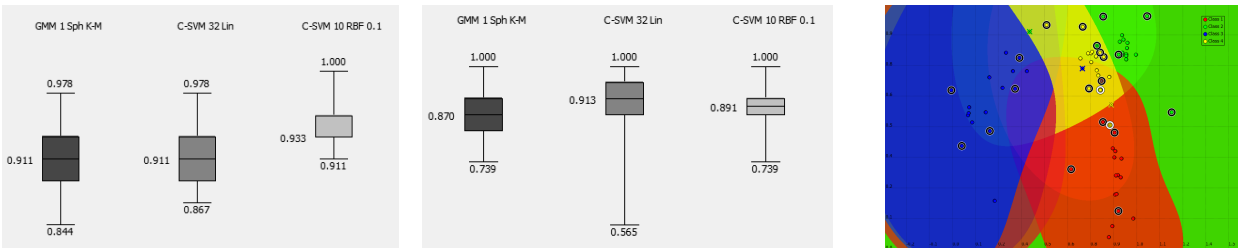
sigma
0.1, 0.
0,001
choice

are compared in the figure 16b.This figure shows that the best combination is $C = 10$ and $\sigma = 0.1$ with the highest F-measure and the lowest variance on training and testing set.

5.4 C-SVM and GMM comparison

The SVM method with an RBF kernel is the most appropriate for classifying dataset 2.

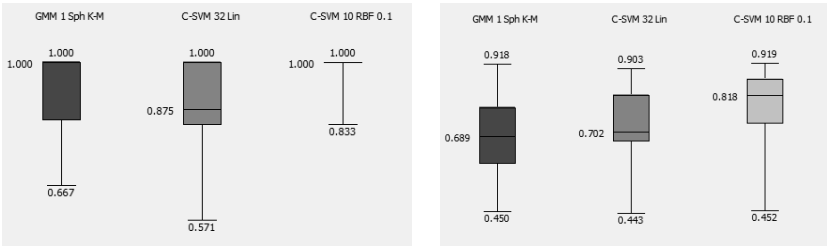
Expliq
un peu
plus



(a) Training set (b) Test set (c) C-SVM ($\sigma = 0.1$ and $C = 10$)

Figure 17: F-measure box-plot comparison with a train/test ratio of 66% and 30 folds for cross validation.

5.5 Test of sensitivity



(a) Training set (b) Testing set

Figure 18: F-measure box-plot comparison with a train/test ratio of 66% and 30 folds for cross validation.

5.6 Overall discussion and conclusion