Machine Learning I
Practice Session II – Clustering Solutions

1 Goals

The goal of this practice session is two-fold: (a) to build a better understanding of how the hyper-parameters of K-means, soft K-means, DBSCAN and GMMs shape the clusters. (b) to learn how to determine the optimal hyper-parameters for these methods by using BIC, AIC and the F1-measure.

The document is divided into two parts:

- Part 1 – How-to: Instructions on how to perform clustering with K-means, soft K-means, DBSCAN, GMM and how to derive the evaluation metrics (log-likelihood, BIC, AIC and F1-measure).
- Part 2 – Tasks and Questions: Set of tasks to be performed during the practical and questions you must answer.

In this second practical, we will use two datasets provided in the previous practice session:

1. Wine cultivar
2. Activity recognition

You can download and find a description of the datasets here.

2 How-to:

2.1 Use the MLDemos interface for clustering

In order to use the clustering algorithms in MLDemos, select the second box from the left in the algorithms panel (Box 1 in Fig. 1)

For this practical, you will need to use the following set of options (Fig. 1) :

1. **Clustering Tab**: Let all clustering options appear.
2. **Cluster**: It launches clustering on the set of data you have drawn or uploaded, using the clustering algorithm you have selected (Box 8) and its hyper-parameters (Box 9).
3. **Clear**: Clears the created model
4. **One iteration:** (available only for K-means and soft K-means). It executes one iteration of K-means or soft k-means algorithms. This allows you to follow the progresses of the clustering.

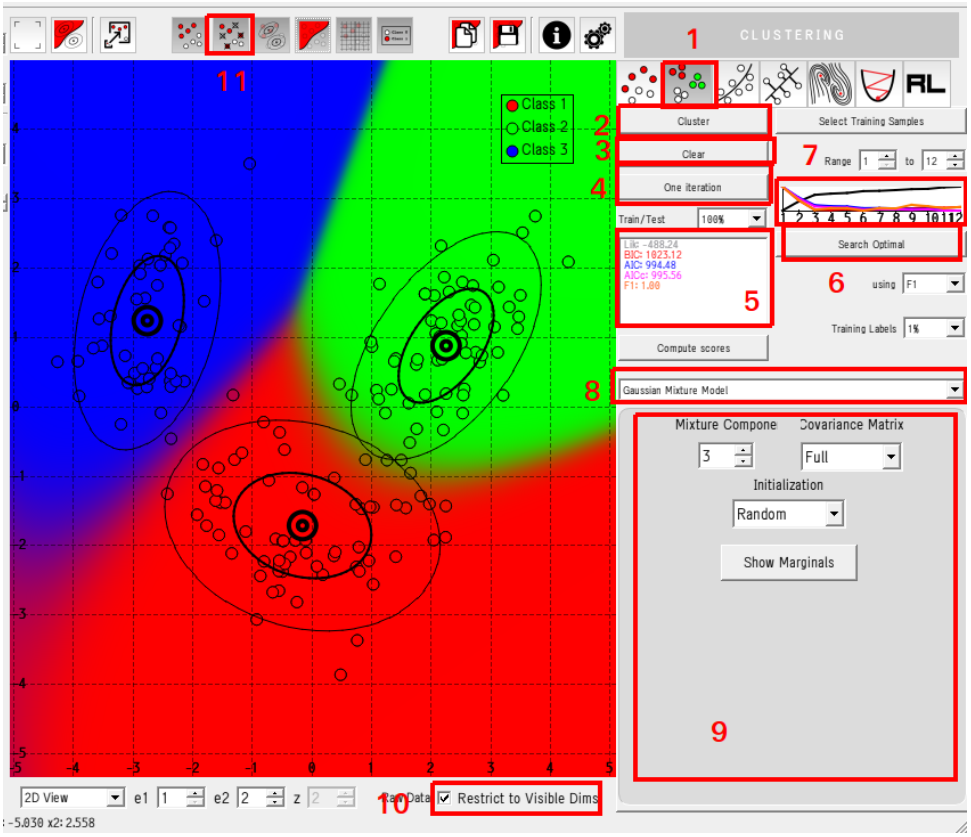


Figure 1: The MLDemos clustering interface

5. **Evaluation block** displays the values of clustering metrics (Log-likelihood / RSS, BIC, AIC, F1-measure) once clustering has converged.

6. **Search optimal:** Performs grid search for the optimal number of clusters according to each criterion (Log-lik, BIC, AIC, AICc, F1-measure)

7. **Curves of evaluation metrics:** Illustrates the curves created by the search optimal function

8. **Clustering algorithm:** Select the desired algorithm to cluster your data

9. **Hyper-parameters block:** You can set the hyper-parameters of the clustering algorithm. Also, for the case of k-means you can select the soft clustering variant from here.

10. **Restrict to visible dims:** If selected, the algorithm is applied only on the dimensions that currently appear in 2D view. Otherwise, the algorithm is applied on all the dimensions that the samples currently have.

11. **Display learned output:** When selected, it colors the points based on the color code of the cluster that they belong to. This can be turned off to get a visualization of samples that clustered together but belong to different classes .

2.2 Perform Clustering

This example demonstrates how to use the clustering interface in order to cluster the *wine* dataset with K-means and perform grid-search to find the optimal number of clusters K.

- Load the *Wine* dataset, perform PCA and reduce its dimensionality by keeping the eigenvectors that explain 90% of the variance
- Using the *Cluster tab*, select k-means with three clusters and click the *cluster* button. This will create a visualization of the clustering model on the 2D view and update the evaluation metrics (Box 5). Please note that the *lik* in Box 5 refers to the log-likelihood and thus it can be negative. You can turn the *Display learned output* off to get a better visualization of samples that clustered together but belong to different classes.
- In order to perform grid search for the optimal number of k, click the search optimal button. This will plot the BIC, AIC and F-measure curves above. The legend for the color-codes in Box 7 is given in Box 5.

3 Questions

Q1: Load the *Wine* dataset, perform PCA and keep the first two eigenvectors. Select the K-means algorithm with K=3 and the euclidean distance as similarity metric. Use the *One iteration button* (Box 4 in Fig 1)and try to predict where the means of the clusters will be located at the next step of the algorithm.

Q2: On the same data, apply the k-Means, soft k-Means, DBSCAN and GMM algorithms by varying their hyper-parameters. Answer the following questions for each algorithm:

- K-means hyper-parameters: k and similarity metric
  - What is the optimal value for k, in terms of class separation?
  - How having k larger than the optimal affects the result?
  - What is the impact of different distance metrics to the result?
  - Find the optimal set of hyper-parameters in terms of class-separation
- soft K-Means hyper-parameters: k and beta
  - How having k larger than the optimal affects the result? Is the effect of k the same as k-means?
  - Vary the beta parameter using high and low values. How and why the value of beta affects the result?
  - Find the optimal set of hyper-parameters in terms of class-separation
- DBSCAN hyper-parameters: *Max distance* (ϵ), *Min samples* and similarity metric
  - Set *Min samples=4*, the euclidean distance as similarity metric and vary the *Max distance* (ϵ). How *Max distance* (ϵ) affects the result?
  - Do the same as above but this time keep both the *Max distance* (ϵ) and similarity measurement fixed and vary the *Min samples* hyper-parameter. How the *Min samples* hyper-parameter affects the result?
  - Test different similarity measurements and mention how they affect the result.
  - Find the optimal set of hyper-parameters in terms of class-separation
- GMM hyper-parameters: Number of components and type of covariance matrix
  - Vary the number of GMM components and mention how they affect the result.
  - Select different types of covariance matrix and mention how they affect the result.
  - Find the optimal set of hyper-parameters in terms of class-separation

– Which of the above algorithms are sensitive to initialization and output different solutions at each run and why?

Q3: On the same data, use the k-means algorithm, vary the number of clusters (from 1 to 10) and create a line plot of the log-likelihood, BIC and AIC w.r.t to the number of clusters. Use the results illustrated in Box 5. of Fig 1 to get the values of log-likelihood, BIC and AIC and create a simple plot at Excel. Answer the following questions:

– How and why the evaluation metrics change with respect to the number of clusters?
– What causes the appearance of a "plateau" at the line plot of AIC and why?
– Why BIC is always larger than AIC?
– What is the optimal k based on the line plot of the evaluation metrics?

You can be helped by righting down the BIC and AIC formulas

Q4: On the same data, use the GMM algorithm with K=3 and k-means initialization. Decide which type of covariance matrix would you use based on the distribution of the data. Then compare the performance of different types of matrices using the F1-measure and BIC measurements as they appear in Box 5. of Fig 1. Which type of covariance matrix would you choose taking into account the F1-measure and which taking into account BIC? Is the choice the same for both criteria and why?

Solutions

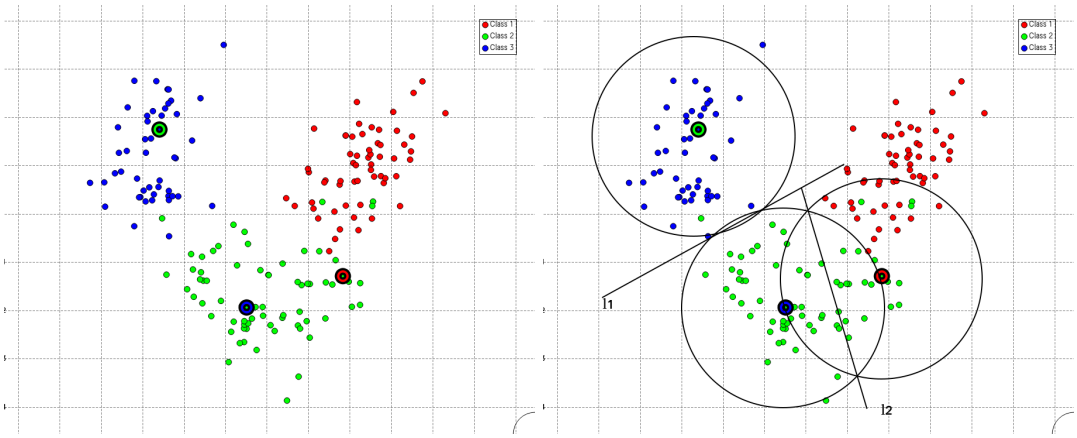
Q1: Load the *Wine* dataset, perform PCA and reduce its dimensionality by keeping eigenvectors that explain 90% of the variance. Select the K-means algorithm with k=3 and the euclidean distance as similarity metric. Using the *One iteration button* try to predict where the means of the clusters will be located at the next step of the algorithm.

S1: In order to find where the mean of a cluster will move at the next step of the algorithm, we need to find the samples for which each mean is responsible. Given an initialization as in Fig. 2a and that euclidean distance is used, the boundaries of the blue cluster with the other two, are defined by the lines $l1$ and $l2$ in Fig. 2b. In order to derive the lines, we need to create k circles with the same radius centered at each mean. The two intersection points of each pair of circles define the linear boundaries of the cluster. Please note that the boundaries are linear because of the euclidean distance which is used as similarity metric. Therefore, the mean of the green cluster cluster will be moved slightly up and left.

Q2: On the same data, apply the k-Means, soft k-Means, DBSCAN and GMM algorithms by varying their hyper-parameters. Specifically:

* K-means: Number of clusters and similarity metric
* soft K-Means: Number of clusters and beta
* DBSCAN: the *Max distance* (ϵ), *Min samples* and similarity metric
* GMM: Number of components and type of covariance matrix

Observe how each hyper-parameter affects the learned model and find the optimal – in terms of class separation – set of hyper-parameters for each method (do not use the search optimal button, find the optimal qualitatively). Which of those algorithms can output different model at each run and why.



(a) Initialization of means

(b) Linear boundaries (l_1, l_2) between the blue cluster and the other two clusters

Figure 2: Geometrical derivation of a cluster's boundaries using euclidean distance as similarity metric

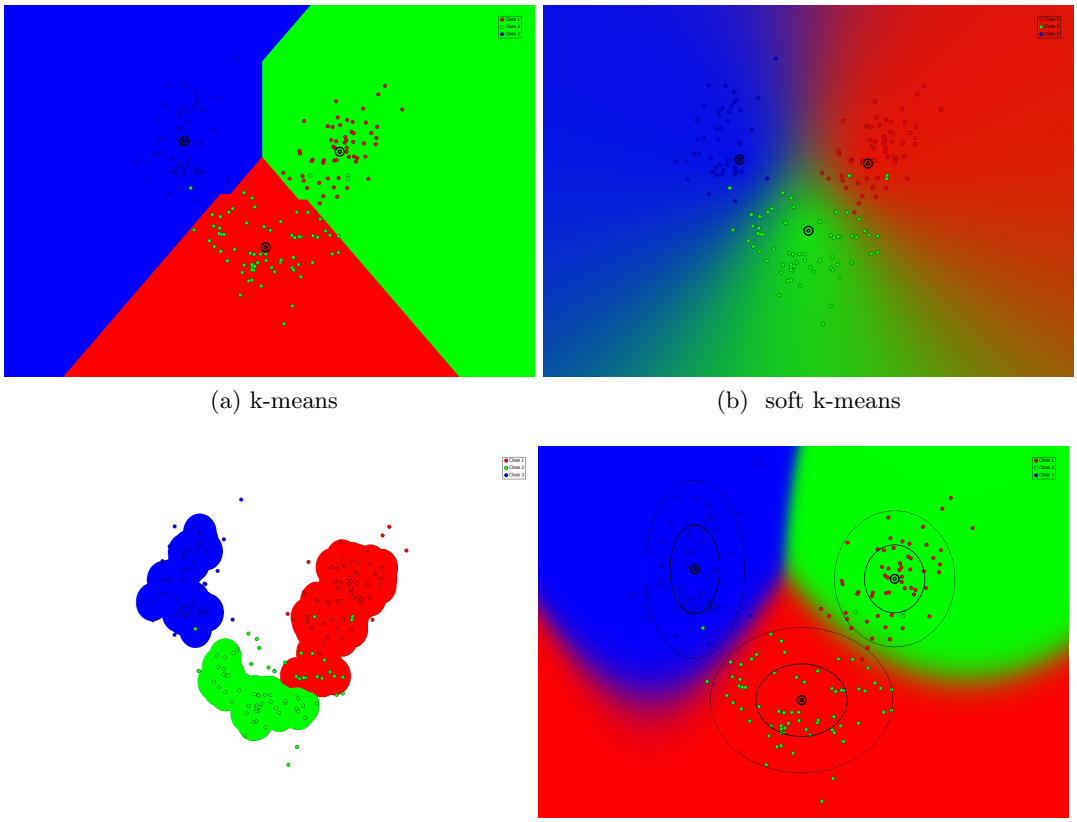
S2:

- k-means: The k parameter controls the number of clusters while the distance metric controls the linearity of the clusters' boundaries. A good solution can be found for k = 3 and L-inf norm as distance metric. (Fig. 3a)
- soft k-means: k has the same impact as k-means. The beta parameter affects the scope of responsibility for each cluster. Choosing a very low value of beta will result to clusters with very similar responsibility over the samples. A good solution can be found for k = 3 and beta = 0.9. (Fig. 3b)
- DBSCAN: In DBSCAN there exist two hyper-parameters, the *Max distance* (ϵ) and the *Min samples*. The *Max distance* (ϵ) defines the max distance that two samples (or clusters) are allowed to have in order to be placed in the same cluster. Increasing the *Max distance* will cause a decrease at the number of created clusters because clusters or points that are further are merged together. The *Min samples* define the minimum amount of points that a cluster is allowed to have and controls the labeling of samples as noise. Another hyper-parameter that one can consider is the distance metric which affects the shape of the clusters' boundaries. Good hyper-parameters are $\epsilon = 0.530$ and *Min samples* $= 5$ (Fig. 3c)
- GMMs: The number of mixtures controls the amount of clusters. The type of the covariance matrix affects the shapes of the boundaries. A good solution can derive for k=3 and diagonal covariance. (Fig. 3d)

The GMM, k-means and soft-kmeans depend on the random initialization since the optimization algorithm (EM) can converge to local minimum. The DBSCAN algorithm does not depend on any kind of initialization and returns always the same solution.

Q3: On the same data, use the k-means algorithm, vary the number of clusters and create a plot of the evaluation metrics (log-likelihhod, BIC and AIC in Box 5. Fig .1) w.r.t to the number of clusters (you can create a simple plot from Excell). Explain how and why the evaluation metrics change with respect to the number of clusters. You can be helped by writing down the BIC and AIC formulas

S3: The plot of evaluation metrics w.r.t. number of clusters is illustrated in Fig. 4 Log-



(a) k-means (b) soft k-means

(c) DBSCAN (d) GMM

Figure 3: Clustering models on the wine dataset

likelihood[1] tends to increase as we increase the amount of clusters. This increase is due to the fact that a large amount of clusters models the data better compared to a lower amount because the distances between the samples and the means are getting smaller as we increase the number of clusters. Nevertheless, the rate of the increase gets lower as we add more components than three. Regarding BIC and AIC, they get smaller as we increase the amount of clusters, until $k = 3$ where they reach a plateau. BIC tends to increase more compared to AIC after that point. This happens because BIC penalizes more the model's complexity by including both the number of samples and the number of optimization parameters at the penalty term. On the other hand, AIC only includes the number of optimization parameters. That's why BIC is always larger than AIC.

Also AIC appears to reach a plateau and does not increase despite that it uses a penalty term. This happens because the improvement of the likelihood for each additional cluster is almost equal to the increase of the penalty term. This makes AIC to reach a plateau for $k > 3$

$$BIC = -2\ln(L) + \ln(M)\, B \tag{1}$$

$$AIC = -2\ln(L) + 2B \tag{2}$$

[1]Here the likelihood is obtained as presented in GMMs lecture (probabilistic perspective of k-means)

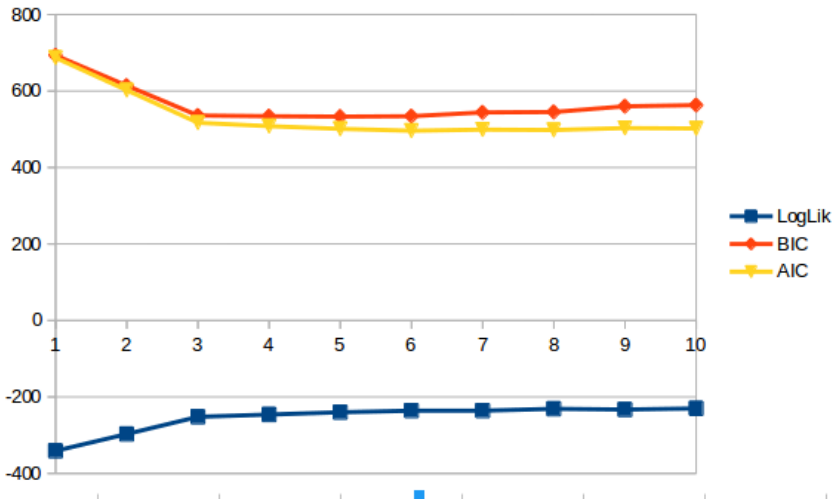
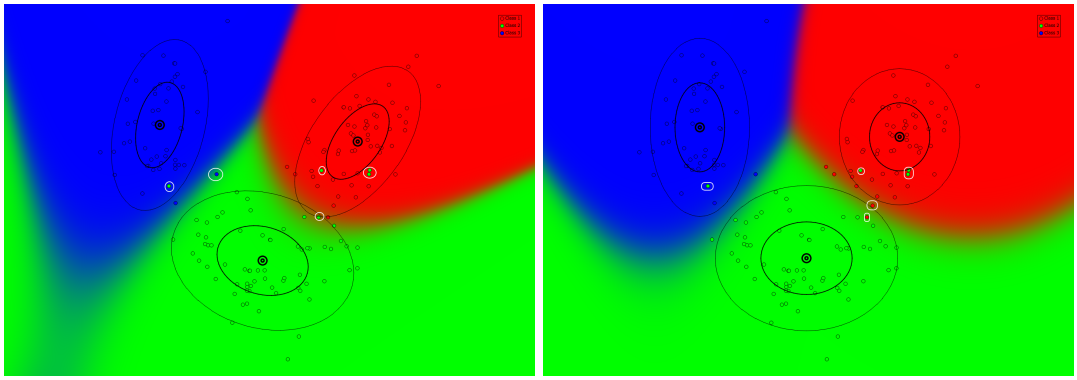


Figure 4: Grid search of k-means hyper-parameter for the wine dataset

Q4: On the same data, use the GMM algorithm with k=3 and k-means initialization. Decide which type of covariance matrix would you use based on the distribution of the data. Then compare the performance of different types of matrices using the F1-measure and BIC. Which type of covariance matrix would you choose taking into account the F1-measure and which taking into account BIC? Is the choice the same for both criteria and why?

S4: A reasonable selection – based on the 2D plots of the first two projections – is a full covariance matrix because there exists intra-cluster correlations which vary for each cluster Fig. 2a. Evaluating different types of matrices using the F1-measure we obtain the results illustrated in Table 1. Based on those results the diagonal and Full covariance matrices perform the same. This happens because both types result to the same amount of miss-clustered samples which affects the F1 measure since it is a supervised metric (white circles in Fig. 5). Since both models perform the same and F1 measure does not penalize the number of optimized parameters, we will choose the diagonal covariance (less computationally expensive model) since it is parameterized by a fewer amount of parameters compared to the full covariance matrix.



(a) GMM – Full covariance matrix (b) GMM – Diagonal covariance

Figure 5: Miss-clustered samples using GMMs with Full and Diagonal covariance matrices

Based on the BIC, the optimal type is the Full matrix because it provides the minimum value. This happens because the improvement of the likelihood – as we switch from diagonal to full matrices– is larger than the complexity penalty ($\ln(M)\,B$). It is expected to get different results from those two metrics because they evaluate the quality of the model based on different criteria. The F1-measure is semi-supervised (requires labels of the samples) and therefore it penalizes miss-classifications while the BIC is unsupervised and provides a trade-off between fit – in terms of likelihood – and complexity.

Table 1: F1-measure and BIC for GMMs with three mixtures and different types of covariance matrices

	Spherical	Diagonal	Full
F1-measure	0.94	0.95	0.95
BIC	1092	1060	1023