EE 565 Machine Learning
Project 5 Report

Mehrdad Ghyabi

***Abstract*—This is a summarized report of the project five of the machine learning course. The main focus of this project is on the Self-Organizing Maps (SOM), histograms and density estimation, and Gaussian Mixture Models (GMM) and classification using those models.**

***Index Terms*— Self-Organizing Map (SOM), Contextual Maps, Kohonen Maps, Trait Query, Histogram, Density Analysis, Gaussian Mixture Models (GMM), Probabilistic Classification**

I. Introduction

This report is consisted of six sections. After a summarized introduction in section I, the performance of SOM on a given database is investigated in section II. In section III, the same algorithm is implemented on another dataset and results are presented. In both sections II and III, the same toolbox has been used to train the SOMs. In section IV, density estimation and expectation maximation are performed on a give dataset using different number of bins. Next, in the section V, ability of GMM algorithm to classify data drawn from double moon distributions is investigated.

II. Problem 1: Contextual Map Using SOM

The input in this part is a given dataset of 17 different animals. Each sample is a vector of 13 different traits which denoted with either zero or one.

To solve this problem different toolboxes were investigated. Two toolboxes in Python, namely "sompy" and "MiniSom" were used but results from neither were satisfying. The problem was inadequate documentation and accuracy. Consequently, SOM Toolbox 2.1 [1] was used in MATLAB environment to train the SOM, however, since the plotting part was already coded in python, the trained maps were exported from MATLAB to Python and resulting images were plotted in Python environment.

A. Toolbox

SOM Toolbox 2.1 is a sophisticated toolbox with many functions and features. As a result, only the functions that were used to solve this problem are going to be explained here. For more information please refer to the creators' report [1].

Before feeding the model with the training dataset, a hot-coded label matrix was appended to the feature matrix, hence, the input matrix has 17 rows and 30 columns. The input data structure was created using "som_data_struct". This function creates a structure including the training dataset as well as label names and feature names. It makes it much easier to interpret the trained map if training data is used in this format.

Next, the map was randomly initialized using the "som_randinit" function. In this step the size of the map (13 by 13) and a hexagonal lattice were assigned to the model.

To train the model function "som_seqtrain" was used. In this step the training data structure and initial map were fed to the function alongside with other necessary parameters. A "bubble" neighborhood function with radius 3 was selected after many trial and errors. Learning rate was set to 0.3 and number of epochs was set to 500.

The U-Matrix was exported using function "som_umat" and neuron weights were exported from SOM object by using syntax "SOM.codebook". As it was mentioned earlier the plotting part was done in the Python environment.

B. Contextual Map

The resulting contextual map is presented in Figure 1. As it can be seen in the figure, there is a meaningful separation between birds and other animals. There is also a separation between predators and non-predators. Similar animals are near each other like horse and zebra or goose and duck or wolf and dog.

C. Trait Maps

Using trained weights, neurons were activated by vectors representing single traits. By comparing activation results, the map was divided into "small", "medium", and "large" regions (Figure 2), "swim", "run", and "fly" regions (Figure 3), and "feather" and "hair" regions (Figure 4). The presented figures show acceptable accuracy with respect to the contextual map.

III. Problem 2: Another Contextual Map Using a SOM

In this part, the same process from problem 1 is repeated on another dataset. Since this data set is from a project with a strict privacy policy, I can not disclose any name or other information that leads to violation of those policies.

As a part of NMSU 2025 strategic plan, one of the colleges of NMSU has to increase its research productivity by 25 percent in five years. To achieve that, all the faculty members working in the college were interviewed. The collected data was transformed into a feature matrix which was used to train a SOM. The feature matrix is presented in Table 1. All the names have been converted to numbers. In Table 1, FM is the faculty

member, D1, D2, and, D3 means the departments they work for, T and TT show their tenure status, and years shows the time they have been working for NMSU. AOE and Productivity determine if they are happy with their allocation of effort and research productivity. P19 and P17 show their publications in the past year and past three years respectively. Grant determines if they have ever been PI of an external grant and writes show how promising their situation is in terms of publications.

The feature matrix was normalized before training phase.

TABLE I
INFORMATION ABOUT NUMBER OF ITERATIONS

FM	D1	D2	D3	T	TT	Years	AOE	Productivity	P19	P17	Grant	Writes
1	1	0	0	0	1	4	1	1	0	0	1	1
2	0	0	1	0	1	1	0	0	0	0	0	0
3	1	0	0	1	0	17	1	1	0	0	1	-1
4	0	0	1	0	1	6	0	0	1	1	0	-1
5	1	0	0	1	0	4	0	0	1	1	1	1
6	0	1	0	0	1	4	1	1	1	1	0	1
7	0	1	0	0	1	3	0	0	1	1	1	1
8	0	0	1	0	1	1	0	0	0	0	0	0
9	1	0	0	1	0	18	0	1	0	0	1	-1
10	1	0	0	1	0	10	0	0	0	1	1	1
11	0	0	1	0	1	3	0	1	1	0	0	-1
12	0	1	0	0	1	0	1	1	0	1	0	0
13	0	0	1	0	1	5	0	0	1	1	1	-1
14	0	1	0	0	1	0	0	0	0	1	0	0
15	1	0	0	1	0	7	1	1	1	1	1	1
16	1	0	0	0	1	4	1	1	0	0	1	-1
17	0	0	1	0	1	2	0	1	1	0	1	-1
18	1	0	0	1	0	11	0	0	1	0	1	0
19	1	0	0	0	1	1	0	0	0	0	0	0
20	0	0	1	1	0	11	0	1	1	1	1	-1
21	0	1	0	1	0	14	0	0	0	0	1	0
22	0	0	1	0	1	3	0	1	1	1	1	1
23	1	0	0	0	1	1	0	0	0	0	0	0
24	1	0	0	0	1	0	1	1	1	1	0	1
25	0	0	1	1	0	10	1	1	1	0	1	-1
26	1	0	0	1	0	6	1	1	0	1	1	0
27	0	1	0	0	1	3	0	0	0	1	1	1
28	1	0	0	1	0	13	1	1	0	0	1	-1

A. Contextual Map

The resulting contextual map is presented in Figure 5. People with similar features are clustered in this map. There is a separation between senior and junior faculty members. Also, people with different levels of research productivity are well separated.

B. Trait Maps

Two trait maps based on department (Figure 6) and employment status (Figure 7) are presented. There is an accurate separation in both cases. The SOM results are going to be very helpful in decision making process.

IV. PROBLEM 3: DENSITY ESTIMATION

In this section, density estimation as well as expectation maximation (EM) are done using a histogram function.

A. Histogram Function

The given algorithm was implemented into a function to calculate histogram of a given data set. The number of bins is the second input of the aforementioned function.

B. Density estimation and EM

Figure 8 shows the histogram of the given dataset with 20 bins. Figure 9 shows the distribution resulting from EM consisting three components. Figure 10 and Figure 11 show the same information only in this case the number of bins is increased to 200. Again, number of bins was increased to 2000 and results are presented in Figure 12 and Figure 13.

Investigating these 6 figures reveals that, as the number of bins is increased the accuracy of the EM algorithm is improved.

V. PROBLEM 4: CLASSIFICATION USING DENSITY ESTIMATION

In this part of the project ability of GMM to classify 1000 datapoints drawn from a double moon distribution with parameters $r = 1$ and $w = 0.6$ and $d = -0.5$ and effect of number of components on it is investigated.

A. Toolbox description

To solve this problem function “mixture.GMM” from “sklearn” library was chosen to develop codes on Python platform. Several model parameters can be defined in this library. The defined parameters are number of components, covariance type (diag or full), threshold, tolerance, minimum covariance, number of iterations, and parameters to update (any combination of weights, means and covariances). After that step the desired distribution is fit to the model using “fit”

function.

B. Implementation

One separate GMMs with one component are fit to each half-moon of the training dataset and the result is presented in Figure 14 for the upper moon and Figure 15 for the lower moon. In this case the resulting distribution can not capture the shape of half-moons.

C. Random draw

In this part, 3000 random point are drawn from the GMM trained in the previous part. The resulting dataset is illustrated as blue dots in the Figure 16. As it was anticipated, the resulting dataset does not look like the training dataset, hence, number of components (K=1) is not a good choice.

D. Accuracy

A test set consisting 1000 datapoints drawn from a double moon distribution with the same parameters was made. Likelihoods from each GMM were used to classify datapoints in the test set. The resulting accuracy was 0.889. In Figure 17 the correctly classified points from the upper moon are shown as blue "+", the correctly classified points from the lower moon are plotted as green "x" and misclassified points are shown as red "*". This figure shows the GMMs are not complex enough to separate datapoints from different classes with a high accuracy.

E. Accuracy

Parts B, C, and D were repeated using different number of components in the trained GMMs. Figure 18 and Figure 19 show the trained GMM with 2 components for upper and lower moon respectively. Figure 20 shows 3000 points drawn from the trained GMMs. It is showing the general form of the double moon distribution but is not accurate. The accuracy of classification of a randomly drawn test set is increased to 0.989. The accuracy of model is illustrated in Figure 21.

Figure 22 and Figure 23 show the trained GMM with 3 components for upper and lower moon respectively. Figure 24 shows 3000 points drawn from the trained GMMs. It is showing the shape of the double moon distribution but there is still noise. The accuracy of classification of a randomly drawn test set is increased to 1.0. The accuracy of model is illustrated in Figure 25.

Figure 26 and Figure 27 show the trained GMM with 5 components for upper and lower moon respectively. Figure 28 shows 3000 points drawn from the trained GMMs. It is showing the shape of the double moon distribution with just a few noisy datapoints noise. The accuracy of classification of a randomly drawn test set is 1.0. The accuracy of model is illustrated in Figure 29.

VI. Conclusions

SOM is a powerful tool for data analysis, high dimensional data understanding and decision making.

For an EM to capture an accurate shape of the histogram of a given dataset, there should be enough bins. The greater number of bins, the more accurate the estimation is

GMM is a powerful and fast tool for classifying tool. The most important parameter while training GMM is the number of components. It is shown that for a training set drawn from a double moon distribution, three components would be enough.

Refrences

[1] http://www.cis.hut.fi/somtoolbox/

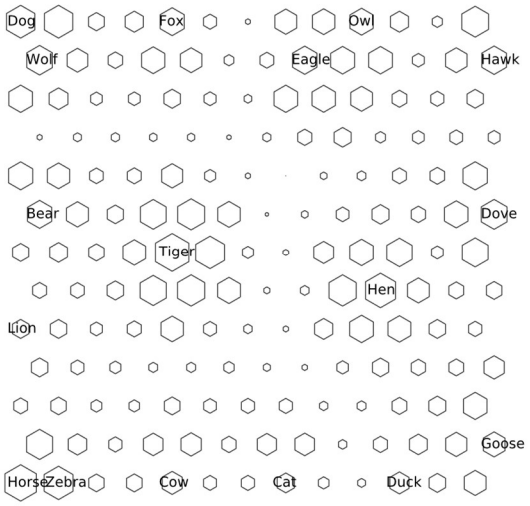

Fig. 1. Contextual map resulting from SOM toolbox

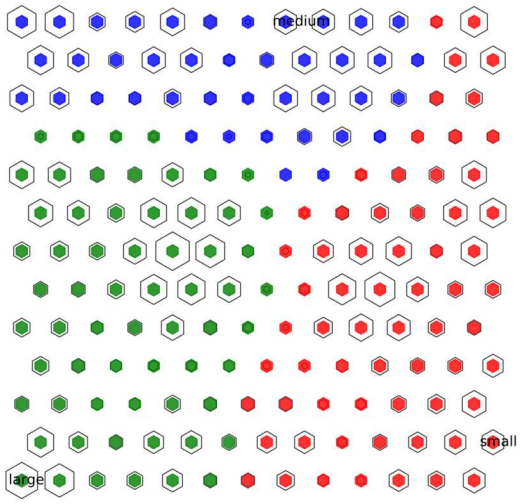

Fig. 2. Size map resulting from SOM toolbox

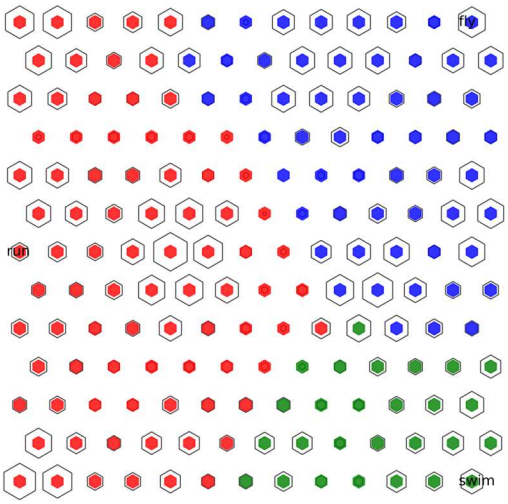

Fig. 3. Ability map resulting from SOM toolbox

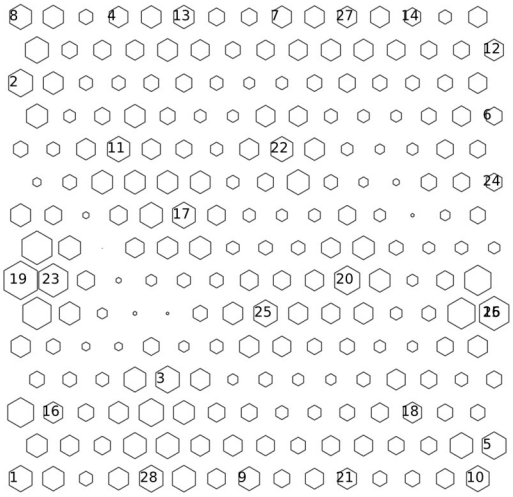

Fig. 5. Contextual map resulting from SOM toolbox

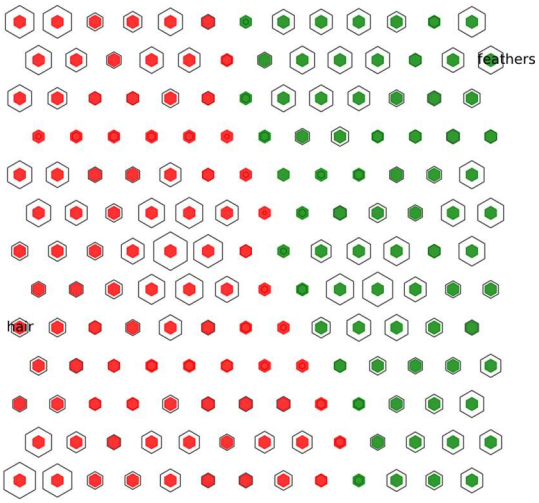

Fig. 4. Cover map resulting from SOM toolbox

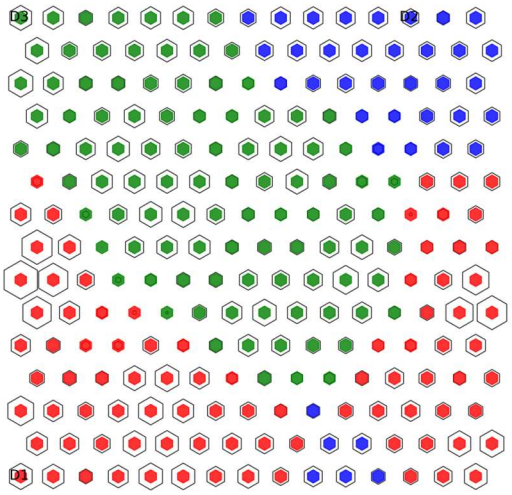

Fig. 6. Department map resulting from SOM toolbox

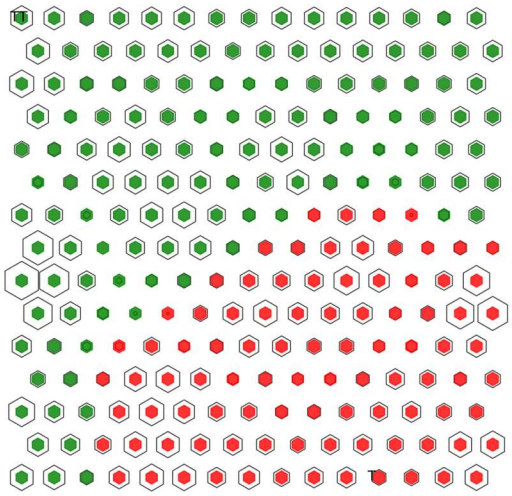

Fig. 7. Tenures vs tenure track map resulting from SOM toolbox

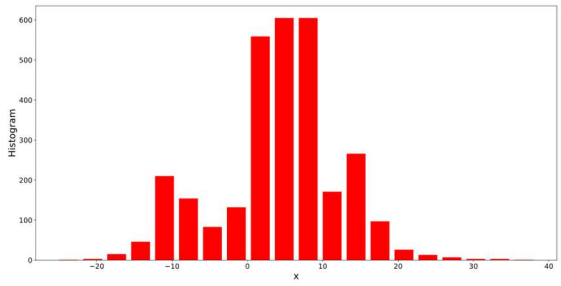

Fig. 8. Histogram with N=20

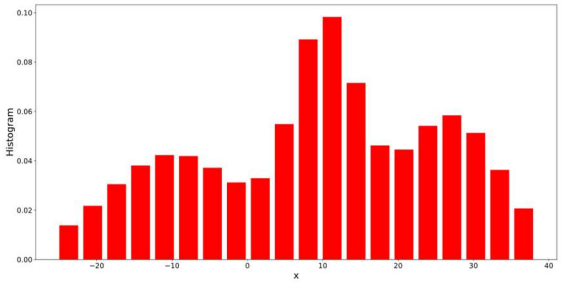

Fig. 9. EM distribution with number of components K=3 and N=20

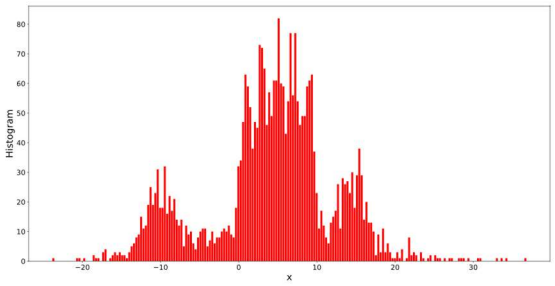

Fig. 10. Histogram with N=200

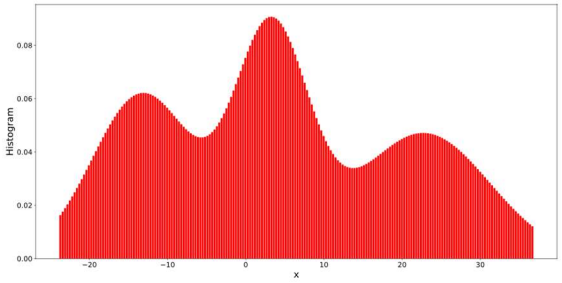

Fig. 11. EM distribution with number of components K=3 and N=200

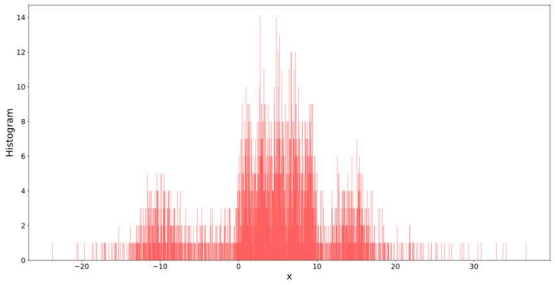

Fig. 12. Histogram with N=2000

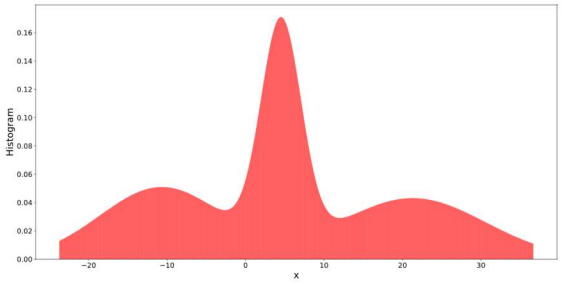

Fig. 13. EM distribution with number of components K=3 and N=2000

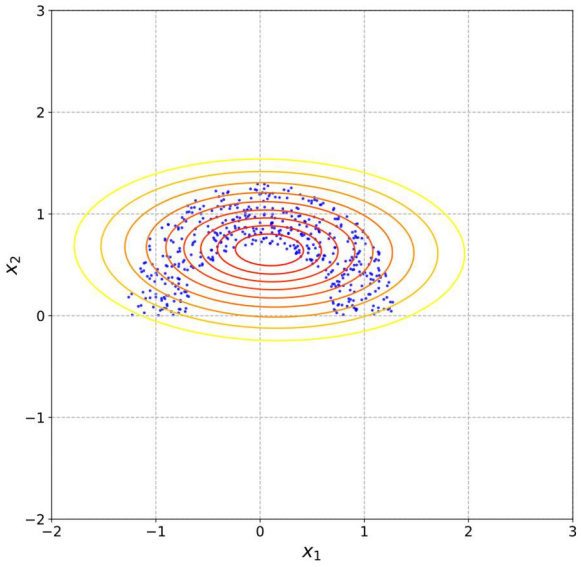

Fig. 14. GMM with K=1 fit to upper half-moon

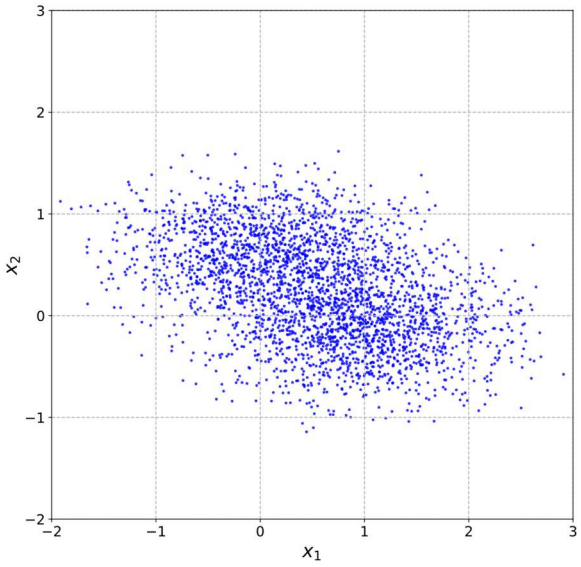

Fig. 16. Dataset drawn from GMM with K=1

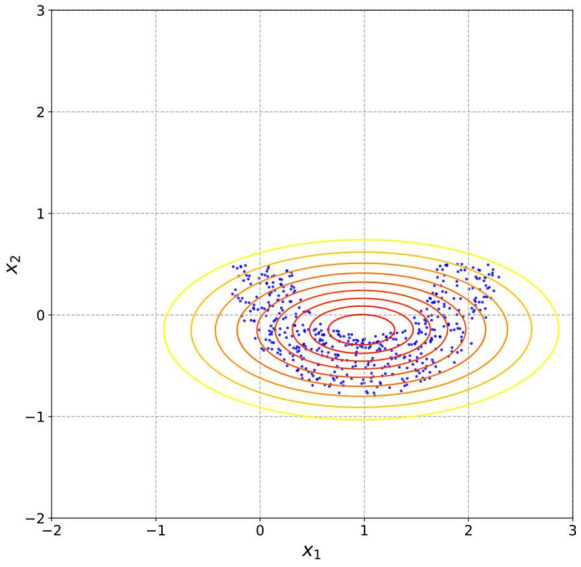

Fig. 15. GMM with K=1 fit to lower half-moon

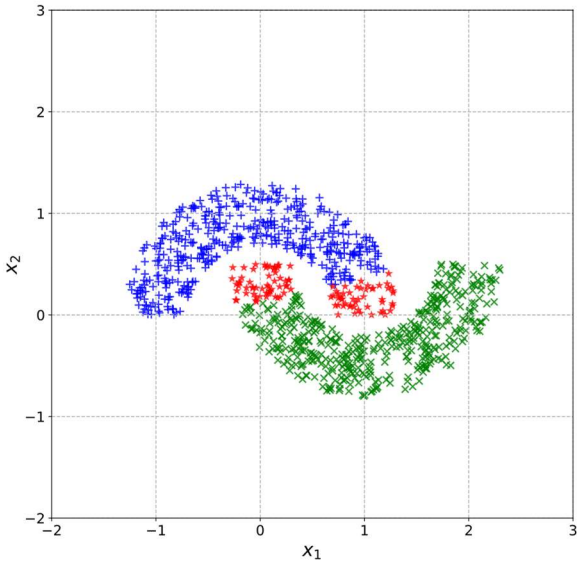

Fig. 17. Test set accuracy for GMM with K=1

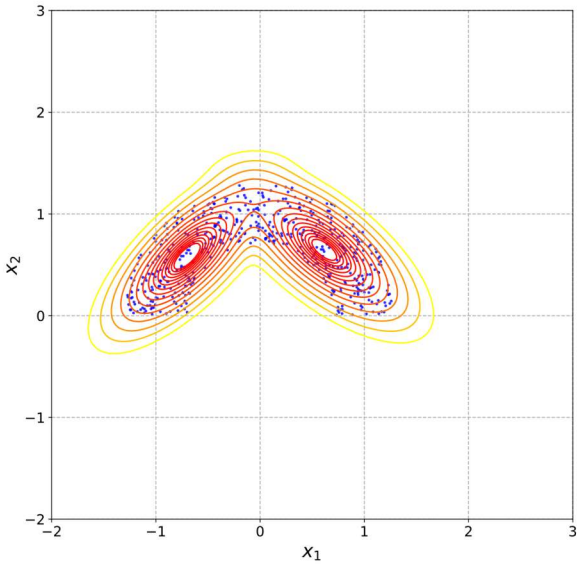

Fig. 18. GMM with K=2 fit to upper half-moon

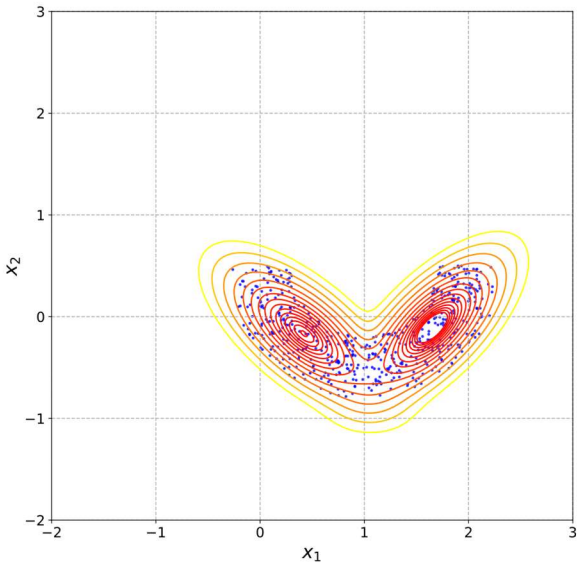

Fig. 19. GMM with K=2 fit to lower half-moon

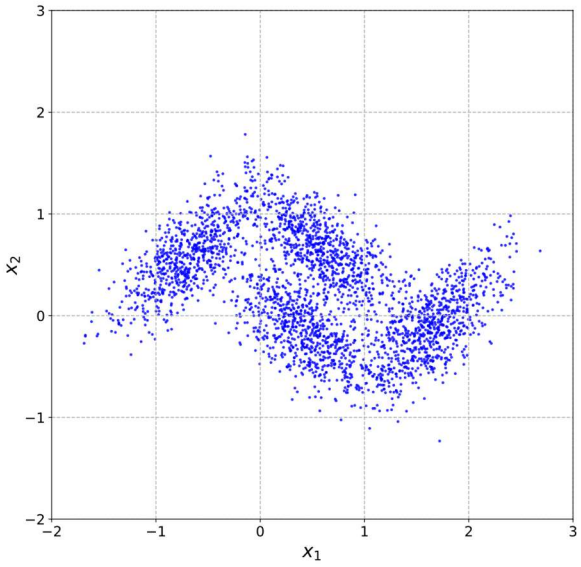

Fig. 20. Dataset drawn from GMM with K=2

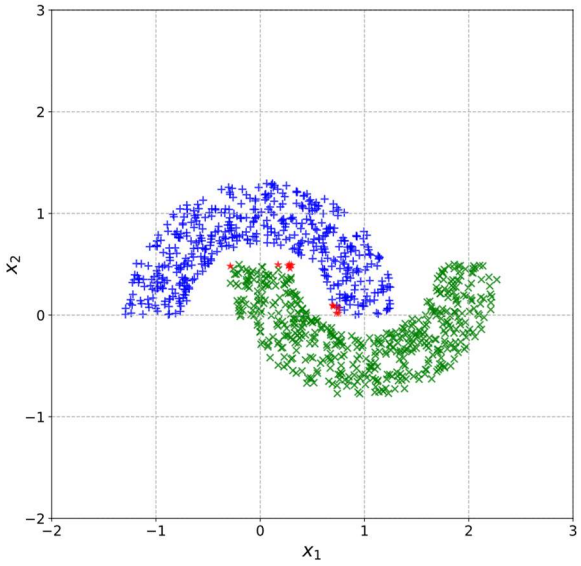

Fig. 21. Test set accuracy for GMM with K=2

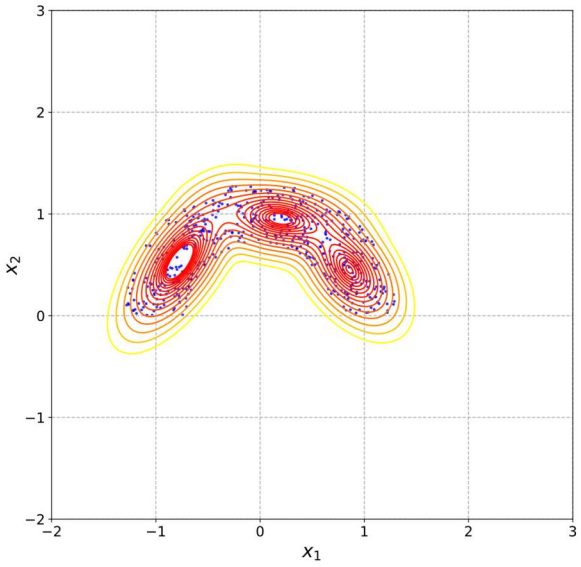

Fig. 22. GMM with K=3 fit to upper half-moon

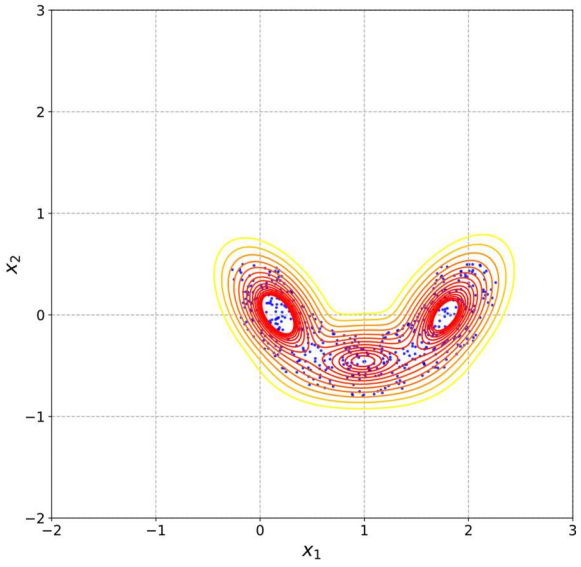

Fig. 23. GMM with K=3 fit to lower half-moon

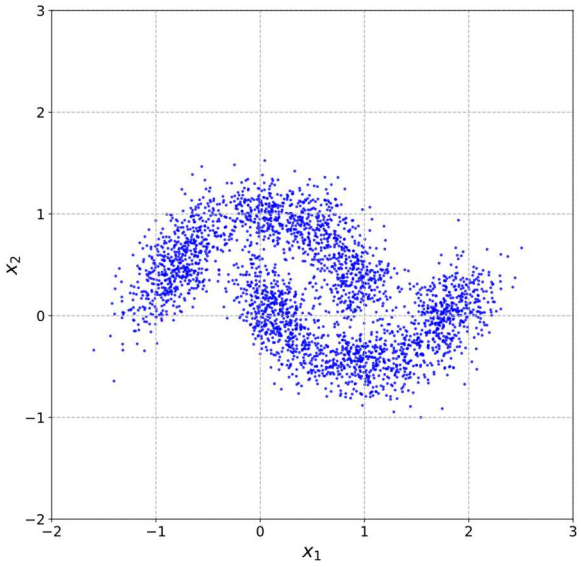

Fig. 24. Dataset drawn from GMM with K=3

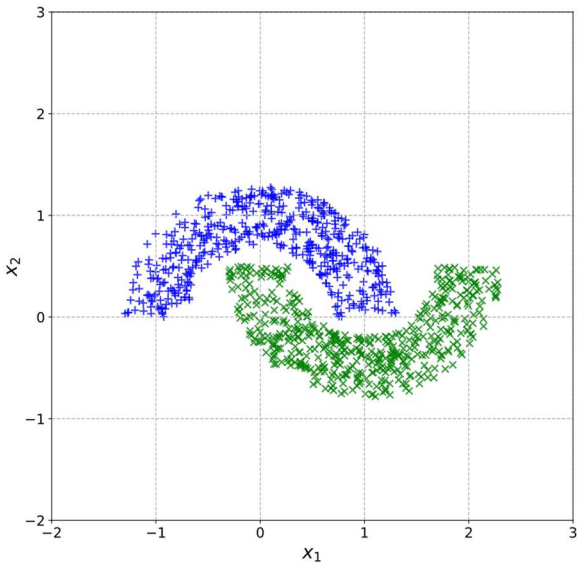

Fig. 25. Test set accuracy for GMM with K=3

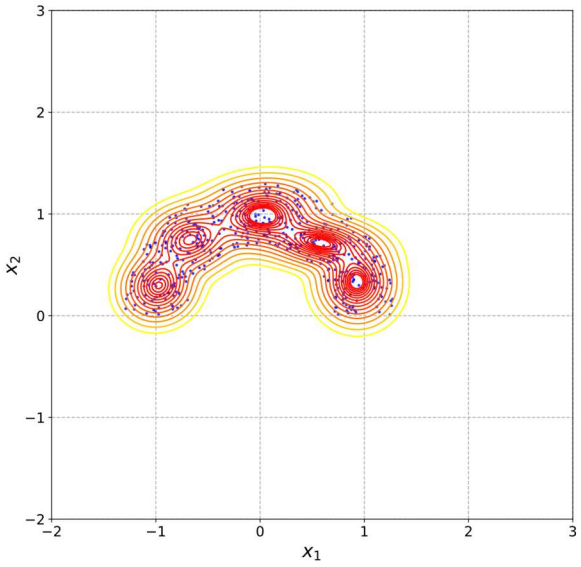

Fig. 26. GMM with K=5 fit to upper half-moon

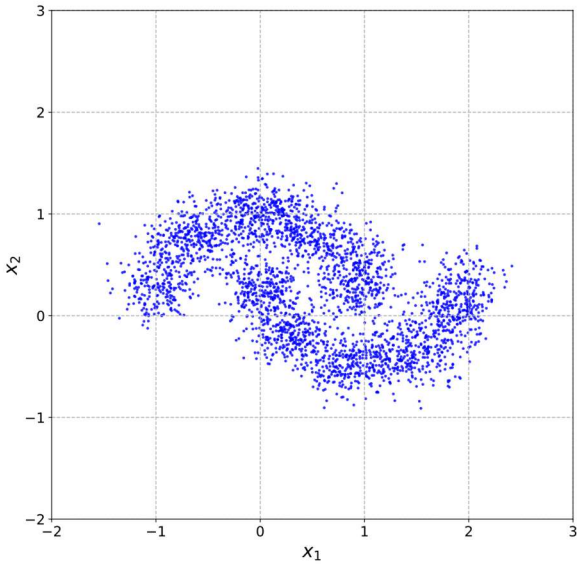

Fig. 28. Dataset drawn from GMM with K=5

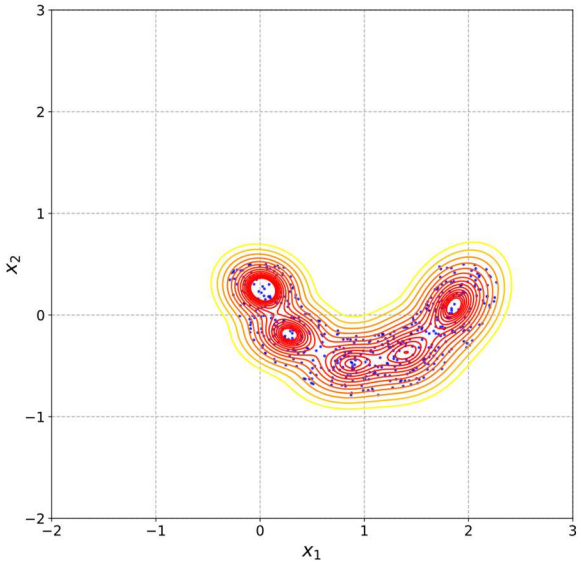

Fig. 27. GMM with K=5 fit to lower half-moon

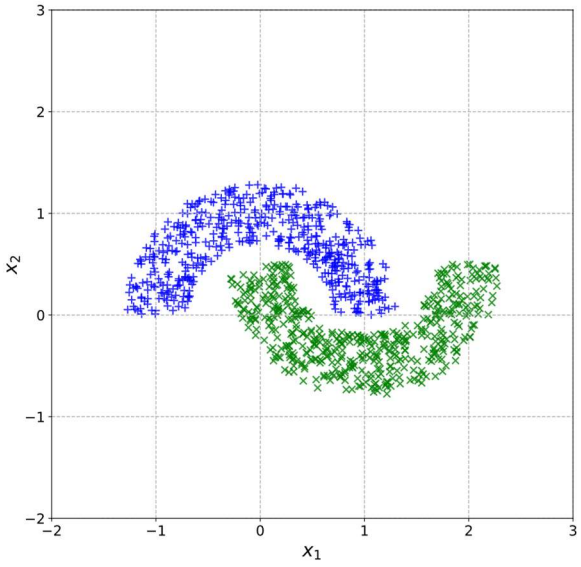

Fig. 29. Test set accuracy for GMM with K=5