Human Psychophysics Project Report

Meldrick Reimmer
Universite de Bourgogne
Master of Science in Computer Vision
Email: meldrick_reimmer@etu.u-bourgogne.fr

This is a report for my course module in Human Psychophysics, where I am to construct a Kohonen Network in order to carry out the classifications of given vectors. I developed a Self Organizing Map (SOM), where it was able to distinguish between two different sets of vectors. It was originally developed by Prof. Kohonen. This is an unsupervised learning but more robust clustering algorithm. In this report, you will find in the references the source codes for the implementation and the datasets are given.

I. Introduction

THE Kohonen network is to map input vectors (patterns) of arbitrary dimension N onto a discrete map with 1 or 2 dimensions. Patterns which are close to one another in the input space should be close to one another on the map. They should be in a topological order. A Kohonen network is composed of a grid of output units and N input units. The input pattern is fed to each output unit. The input lines to each output unit are weighted. These weights are initialized to small random numbers.

The winning output unit is simply the unit with the weight vector that has the smallest Euclidean distance to the input pattern. The neighborhood of a unit is defined as all units within some distance of that unit on the map (not in weight space). In the demonstration below all the neighborhoods are square. If the size of the neighborhood is 1 then all units no more than 1 either horizontally or vertically from any unit fall within its neighborhood.

You will find in this report the process in which I undertook in implementing the network just as described above.

II. Objectives

For this project to be done I underwent the following steps:

1) Constructed a Kohonen network in carry out the classifcations of vectors given.
2) Once the training was completed, I carried out a test with another vectors given.
3) Based on the Kohonen network that I have already constructed., I have to classify between a patient and a control given a data.
4) Lastly, I am to state differences if any between bio-inspired algorithm like the Kohonen SOM and two other well-known similar clustering algorithms.

III. Performing Tasks

Task 1 & 2: Construct a Kohonen Network

I created a function in which I called “KohonenTrainNTest(TrainData,TestData)”. This function takes two inputs which are the given vectors . One being the train and the other test vector. The learning process is as roughly as follows:

- initialise the weights for each output unit
- loop until weight changes are negligible
- for each input pattern :
  - present the input pattern
  - find the winning output unit
  - find all units in the neighbourhood of the winner
  - update the weight vectors for all those units
- reduce the size of neighbourboods if required

These are basically the steps I used to construct my network. The Weight forms an essential part since it is what we will use to be able to determine which cluster the data given belongs to. We determine the rate by converging synaptic values. I initiated my learning rate to 0.8 and my iterations were to 50. This converges since my learning rate was high.

With this, my next task was to test the network by the test vectors given (0, 0, 0, 0.9), (0, 0, 0.8, 0.9), (0.7, 0, 0, 0), (0.7, 0.9, 0, 0). The vectors consisted of two classes and the expected results was to classify each vector in accordance to each class. The vectors (0, 0, 0, 0.9) and (0, 0, 0.8, 0.9) falls in ClassII and the vectors (0.7, 0, 0, 0) and (0.7, 0.9, 0, 0) falls in Class I.

Below is the results :

```
Test Data for Task 1:

TestData =

         0         0         0    0.9000
         0         0    0.8000    0.9000
    0.7000         0         0         0
    0.7000    0.9000         0         0

Results:
[0          0           0          0.9 ] This Vector Belongs to Class 2
[0          0         0.8          0.9 ] This Vector Belongs to Class 2
[0.7          0           0          0 ] This Vector Belongs to Class 1
[0.7        0.9           0          0 ] This Vector Belongs to Class 1
```

We can see from the figure above that the expected results were achieved using the network in which constructed. This is done by calculating the Euclidian Distance of each test vector with weight vectors. And the minimum distance vector wins and assigns its class.

Task 3: Training With New Data & Test

Having to be able to get accurate results from the previous task, the next task was to be able to train my network with data given which were “Patient.txt” and “Control.txt”. Each had a dimension of (10x650). Each line corresponds to the data (time series) coming from one subject. The time series is made up of the displacements of markers placed on the joints of subjects. There are ten subjects in each file.

For this part, I created a function called “kohonenProj3(Data, MyTestData, Nweights)” . This takes in three inputs being “Data” which is the output of the two train data and “MyTestData” being my personalized data with 4 subjects and “Nweights” being the weights in which I determined the cluster in which my personal data belonged to.

This process was first done by getting the weight from the two training data given after I reshaped it and assigned it to the variable “Nweights”. After I used it in getting the “Data” by using the function “KohonenProj2(RealTrainData,RealTestData)” . After this, I then used my function “kohonenProj3(Data, MyTestData, Nweights)” to determine which cluster each vector belong to.

Below is the result :

```
Test Results using MeldrickTest Data:

Test Vector 1 Belongs to Control
Test Vector 2 Belongs to Control
Test Vector 3 Belongs to Patient
Test Vector 4 Belongs to Patient
>>
```

Task 4: Differences Between Kohonen SOM, K-Means & K-Nearest Neighbour

Kohonen SOM	K-Means	K-Nearest Neighbour
More Robust Learning	Sensitive to initialization	The target function may be either discrete_valued or real valued
Classification Algorithm	Clustering Algorithm	Classification Algorithm
Unsupervised learning	Unsupervised learning	Supervised learning
Less sensitive to the noise present in the datase	More sensitive to the noise present in the dataset	Dominated by large number of irrelevant features
Clusters by geometrically	Clusters are formed through centroid and cluster size	Use of Vornoi diagrams

IV. Conclusion

Kohonen Network, in general, is very easy to understand and very simple to implement especially if they are close together and they are similar. You will be able to classify the data well which makes it good to differentiate how good one is to the other.

Some of the weakness though is the facts that getting the right data at the start is very much complexed compared to other algorithms and also they are very computationally expensive which is a major drawback since as the dimensions of the data increases, dimensions reduction visualization techniques become more important but time to compute them also increases.

References

[1] Lectures On Kohonen Networks By: Elizabeth Thomas

[2] Kohonen Networks (SOM) Algorithm : https://www.cs.bham.ac.uk/~jlw/sem2a2/Web/Kohonen.htm

[3] Teuvo Kohonen, Self-Organizing Maps, Springer Series in Information Sciences, Vol. 30, Springer, Berlin, Heidelberg, New York, 1995, 1997, 2001, 3rd edition

[4] Github Accout link (Source Code) : https://github.com/brown4eva/Classification-of-Patient-and-Control-using-Kohonen-Networks