# Application of SOM to Pen-Based Recognition of Handwritten Digits Data

CECS 590

Computational Intelligence

Submitted to Dr. Frigui

by

Behnoush Abdollahi

February 2013

## Problem description and data description

For this project I used the “pen-based recognition of handwritten digits” data set from the UCI online repository. The training dataset has 7494 samples that are written by 30 different writers, and it consists of 16 attributes that are (x, y) coordinates of the 8 different points.

By connecting all the points, we will get the hand written style for the digits. The visualization of some records is shown in Figure 1.

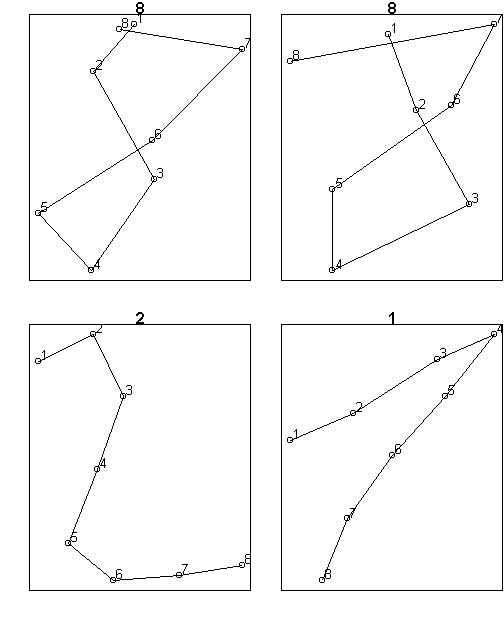


Figure 1- Visualization of sample records

## Clustering: Self Organizing Maps (SOM)

The goal of clustering is to look for patterns in the set of objects without any supervised help from an expert. Artificial Neural Networks (ANN) is one of the methods that can be used to find the association pattern between the attributes in a dataset. One of the clustering approaches of the ANN is Self Organizing Maps (SOM) algorithm.

SOM assumes a rectangular or a hexagonal topology for arranging the units in the network and every time a record is assigned to a winning node the neighboring nodes also get updated. This results in obtaining a map in which similar nodes are located close together.

SOM is a clustering algorithm that is mainly used for visualization, because it decreases the attributes to one or two dimension that can be perceived better by the human eyes. SOM is easy to implement and for this dataset using SOM helps look at the similarity of writing different digits by different people.

## Implementation

For this project I used R language and it is implemented in R studio. I used the package “kohonen” which has the self-organizing maps algorithm implemented. Following is the function I used with the input parameters:

som(data, grid, rlen, alpha, radius, init)

* **grid**: the rectangular or hexagonal grid of network units.
* **rlen**: the number of epochs.
* **alpha**: the learning rate, determining the size of the adjustments during training. The decrease is linear.
* **radius**: the initial size of the neighborhood, by default chosen in such a way that two-thirds of all distances of the map units fall inside this number. The size of the neighborhood decreases linearly during training;
* **init**: optional matrix of codebook vectors. If it is not given, randomly selected objects from the data are used.

|  |
| --- |
|  |

## Results

The hit results from the application of SOM method to the dataset with the following input variables is shown in Figure 2:

* grid: 5 by 5
* topology: rectangular
* rlen = default = 100
* alpha = from 0.05 decreasing linear to 0.01
* radius = 3
* init = default = random

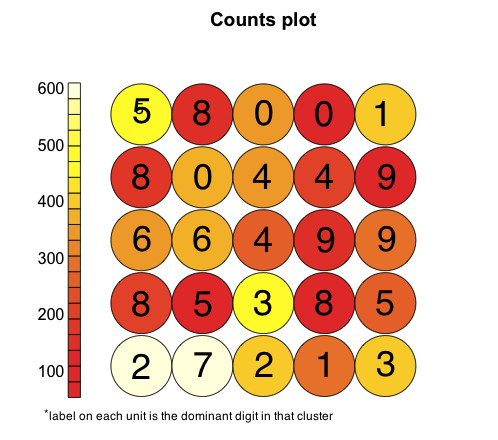


Figure 2- Cluster hits

The counts plot shows how many records are mapped to each cluster. The label on each unit shows the dominant digit in that unit. From the way that units are arranged, it can be seen that same digits, and digits that are similar are located close to each other. For example, from this graph it can be obtained that there are 4 different style of writing 4 and they are located close to each other on the graph which due to SOM that updates neighbor units in addition to the winning unit. Also, we can see that how some digits are written like each other. For example, the units containing 2 and 7 are close to each other, which mean they have similar shape and similar input features. Some digits have happened to be scattered like digit 5. However, all 5 and 8’s are located close to each other. This is because of the initialization of the grid, which has affects on the final organization of the units, and on the organization of the units.

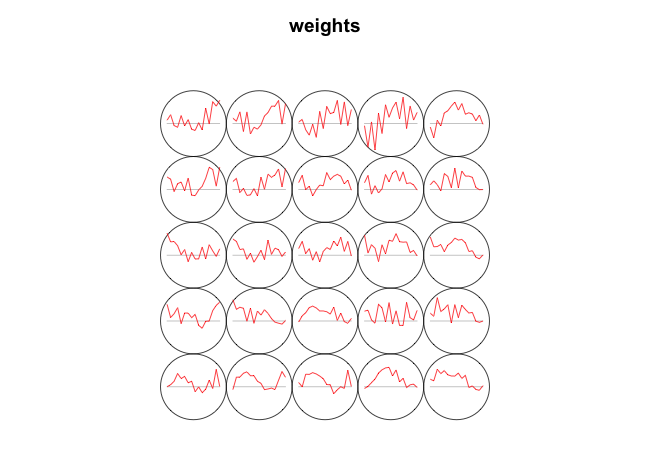


Figure 3- Cluster weights

The graph of final weights of the clusters, in Figure 3, which is also called code vector shows that clusters close to each other have similar patterns, which is due to the topological structure of the SOM.

The color of each unit shows how many records are in that cluster. Most units have records smaller than 400 records. However, a few units have around 600 records. To find out how digits are mapped to each cluster it is a good idea to look at the distribution of the digits in each cluster.

Figure 4 shows the distribution of the clusters in one plot. Overall, it is obtained that in most clusters mainly contain the records of one digit. However, for example, cluster 9 contains 2 main digits.

Quality of the clusters, which is shown in the following figure, is measured as the mean distance of objects mapped to a unit to the codebook vector of that unit. The smaller the distances, the better codebook vectors represent the objects.

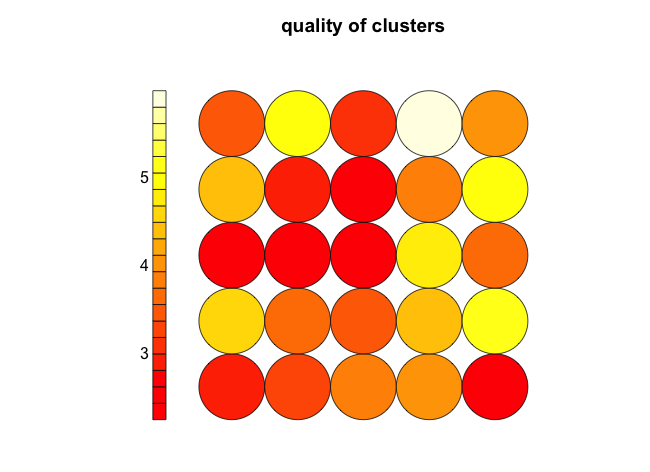


Figure 4- Mean distance of objects mapped to a cluster

The sum of the distances to all immediate neighbors for each unit can be viewed in Figure 5. Units that are very similar have smaller average distances to their neighbors.

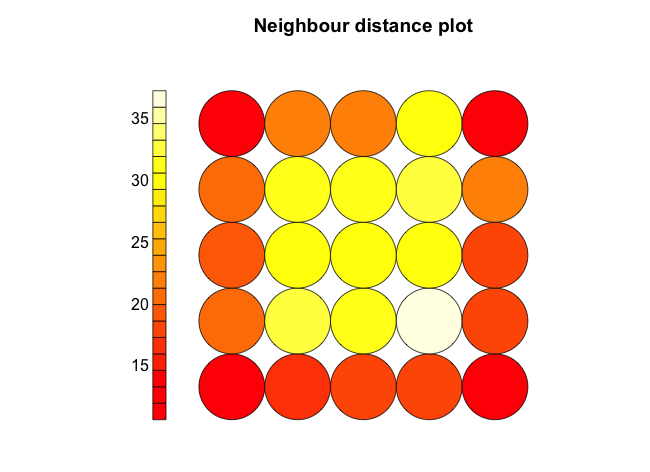


Figure 5- The sum of the distances to all immediate neighbors

Comparing the count and the neighbor distance plot, we can see that the bottom left clusters that have smaller distance, also have higher counts of their mapped digit and the digits (7 and 2) are having similar written style.

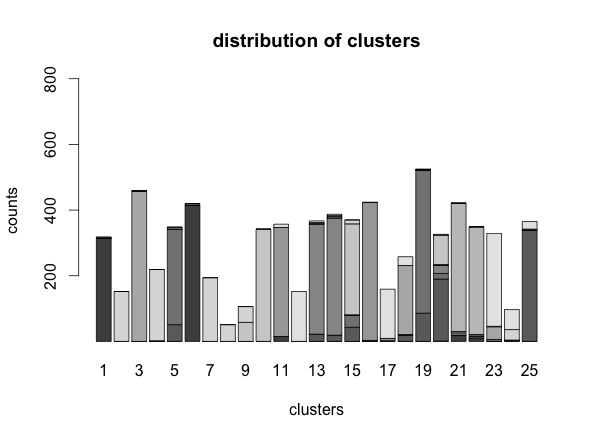


Figure 6- Distribution of clusters

From Figure 6 it is not clear what digits and how many are grouped in each cluster. Using the distribution of the digits in each cluster, it can be obtained that how many of which digits are mapped to a cluster (Figure 7).

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Figure 7- distribution of digits in each cluster

**Grid**:

One of the important parameters to initialize is the number of clusters. In this data set we are looking at different ways that people write the ten digits, so at least 10 clusters are needed. I started with a 4 by 4 grid, and 5 by 5 grids.

The plots in Figure 8 show the distribution of the digits and clusters.

|  |  |  |
| --- | --- | --- |
| 4 by 4 grid |  |  |
| 5 by 5 grid |  |  |

Figure 8- Distribution of clusters and digits

By looking at the first column it can be seen that by increasing the number of clusters, there are still some clusters that have obtained more than one digits in them like cluster number 15 and 20. Second column shows the distribution of digits. For some digits, like 3, we can see that by increasing the number of clusters, more major groups have appeared and it shows that most likely there are two major types of writing digit 3. Another main significance can be seen for digit 7. In 4 by 4 grid there is one main cluster for 7, but in 5 by 5 grid another major group has appeared. The distributions of other digits don’t seem to have changed significantly.

Looking at the data it appears that there are at most 3 types of main style for writing each digit.

For this matter I also tried a non-symmetric grid of size 4 by 6 and compared it with 5 by 5 grid (Figure 9).

|  |  |  |
| --- | --- | --- |
| 5 by 5 grid |  |  |
| 4 by 6 grid |  |  |

Figure 9- Distribution of clusters and digits

Comparing the distributions of digits in both cases, it can be seen that the only major difference is for digit 5 when changing the dimensions from 5 by 5 grid to the 4 by 6 grid. In 4 by 6 grid three major groups for digit 5 is obtained. The other digits seem to have a similar distribution over the clusters regarding the major number of ways of writing a digit.

Another grid arrangement would be a one-dimension grid. The results for 15 by 1 grid comparing to 4 by 4 grid is shown in Figure 10:

|  |  |  |
| --- | --- | --- |
| 4 by 4 grid |  |  |
| 15 by 1 grid |  |  |

Figure 10- Distribution of clusters and digits

Comparing the two-dimension grid with the one-dimension, from the first column, the distributions of the clusters seem to be similar. From the second column, distributions of the digits are almost the same too, except for digits 6, and 9.

**Topology:**

Another topology for the SOM is hexagonal option. The results of the clustering in both cases for the 5 by 5 grid is in Figure 11:

|  |  |
| --- | --- |
|  |  |

Figure 11- “Rectangular” and “hexagonal” topology

By changing the topology, a different arrangement for the digits is obtained. However, there are many similarities in both cases: 2 and 7 are close on the map. 9 and 4 and 5 are also close. Also, another difference is for the number of clusters obtained for each digit. For example, 3 is grouped in to 2 clusters in the rectangular topology, however using hexagonal it can be perceived that most people write 3 the same way. A major difference between these two topologies is the structure of neighbor nodes that are taken into consideration when neighbor nodes get updated. Figure 12 shows the difference between the “rectangular” and “hexagonal” topologies.

|  |  |
| --- | --- |
|  |  |

Figure 12- Neighborhood

In the “rectangular” topology, for neighbors in distance equal one 8 nodes are considered as the neighbors, however in the “hexagonal” topology, 6 nodes are in that neighborhood.

**Rlen:**

Rlen is the number of epochs. The default value for this parameter is 100 in the package used. For the 5 by 5 grid the mean average of distance to the closest unit in each iteration is calculated (Figure 13).

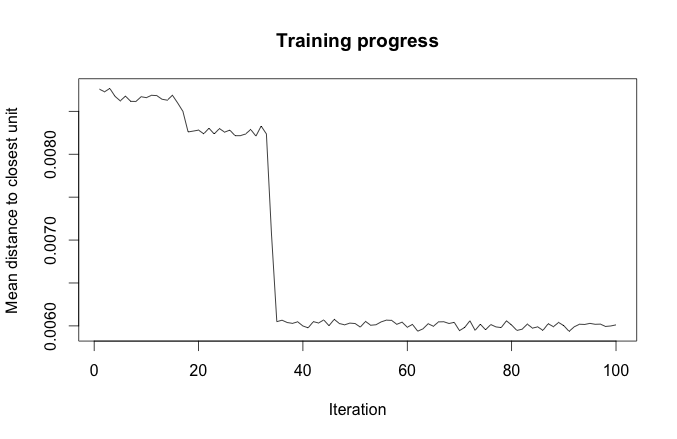


Figure 13- Training progress

from the plot it is obvious that from some point around 35 iterations there is no significant change in the record distances to the wining units and Rlen can be set to any number larger than 35 to have the clusters converge.

**Radius:**

Radius variable indicates the size of the neighborhood to be considered. The size of the neighborhood decreases linearly during the operation; after one-third of the iterations only the winning unit is being adapted and the algorithm corresponds to k-means. To see the impact of the radius on the results, the distance maps for radius = 1 and radius = 3 for the 5 by 5 grid is obtained.

|  |  |
| --- | --- |
|  |  |

Figure 14- left: radius = 1, right: radius = 3

From the Figure 14, it can be obtained that the sum of the distances to all immediate neighbors. For radius=1 the sum of the distances are larger than radius =3. This is due to the fact that smaller neighborhood is considered when radius is small and the SOM acts more like K-means, in which the clusters close to each other aren’t necessary similar.

## Conclusion

In this project the application of SOM to the pen-based handwritten dataset is assessed. Changes in different parameters are looked at in detail. The number and the arrangement of the clusters and neighbors have impact on the final map result. Some digits that are located close to each other seems to have similar features. However, by increasing the number of clusters there are still some misclassified records because they are written very close to the digit they are placed in. SOM gave a good output results to look at this similarities and the number of clusters that can be assumed for each digit based on the dataset.

## References

1. <http://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Self-Organizing_Maps_(SOM>)
2. http://archive.ics.uci.edu/ml/datasets
3. Self- and Super-organizing Maps in R: The kohonen Package by: Ron Wehrens and Lutgarde M. C. Buydens
4. Package “Kohonen” document