**Lab 6 – Unsupervised Learning: SOLN**

1. What is the purpose of the neighborhood function in the SOM? How does it change the learning?

**Ans:**

Neighborhood function is used to maintain topography. The training is changed to include a learning for the neighborhood neurons using a neighborhood function with different learning rate. Neighborhood function is adjusted at every epoch.

1. A simplistic intruder detection system for a computer network consists of an attempt to categorize users according to (i) the time of day they log in, (ii) the length of time they log in for, (iii) the types of programs they run while logged in, (iv) the number of programs they run while logged in. Suggest how you would train a SOM classifier to perform the categorization. What preprocessing of the data would you do, how much data would you need, and how large would you make the SOM? Do you think that such a system would work for intruder detection?

**Ans.:**

Generally speaking, all the 4 input variables i.e.(i) the time of day they log in, (ii) the length of time they log in for, (iii) the types of programs they run while logged in, (iv) the number of programs they run while logged in, are co-related as intruders actions / behavior are related when they logged on. So, ideally we can consider 2 clusters – one for all 4 inputs (features) in a normal circumstance and the other when an intruder is logged on. The data can be combined and SOM can learn the topology for both types of data. Due to this capability, SOM can learn outliers reasonably well.

[Note – this is Outlier / Anomaly detection. These can be done using various methods (including MLP or Logistic Regression). But SOM can provide better results due to topological inputs etc. See more below]

**[NOTES:**

1. Using supervised learning to learn just fraud (anomaly) will be difficult as amount of regular data is very large and intrusion data is very small. Supervised learning cannot learn outliers well. However, Supervised learning can also be used with some extra care – see the notes below in Read for more details.
2. Hence, unsupervised learning is a better approach in general. Challenge is to detect the outliers as it is mainly a outlier / anomaly detection. From this aspect, SOM can do a better job as the weights can learn the normal pattern (after good normalization) as well as outliers. However, some times abnormal data might be within noise boundary. Hence, sometimes, unsupervised SOM is combined with some supervised learning – intrusion data can be used to further refine the clusters by using some labeling.
3. For better accuracy, multiple clusters can be used for good data and intrusion data.

]

**Preprocessing** is needed at least to normalize the input vector. Other preprocessing are also needed – e.g. the sources that we should monitor may change frequently from interval to interval. At any given interval, the observed source activity could rise far above or sink far below the activity level of N sources used by the neural network in the previous interval. To accommodate such changes, we need to cluster the sources based **on time-windowed behavior** and present the cluster totals to the neural net.

**Data Size:** A good start would be to use 10 times # weights.This can be calculated by doing some initial processing on the data and coming up with a reasonable # of clusters (or Separate SOM can be used with reasonable data / neurons in each type to determine a reasonable # of clusters). Once we have a reasonable # of clusters, the size of the neural net and hence the # of weights can be calculated. In general, this is an iterative process.

NOTE: In case supervised learning is used, we would need to use Recall and Precision

instead of Accuracy as accuracy will not be the correct parameter and its value would

be wrong. It just reflects the distribution of data in the class e.g. a 90% positive and

10% negative examples will usually yield 90% accuracy etc. Alternatively, supervised

learning can be used by balancing the data using over sampling of less data class and

under sampling of the more data more data class.

**See Credit Card Fraud Detection Note for Project and labs.**

1. A bank wants to detect fraudulent credit card transactions. They have data for lots and lots of transactions (each transaction is an amount of money, a shop, and the time and date) and some information about when credit cards were stolen, and the transactions that were performed on the stolen card. Describe how you could use a competitive learning method to cluster people's transactions together to identify patterns, so that stolen cards can be detected as changes in pattern. How well do you think this would work? There is much more data of transactions when cards are not stolen, compared to stolen transactions. How does this affect the learning, and what can you do about it?

**Ans.:**

Since there are too much data when cards are not stolen compared to when cards are stolen, we somehow process the data so that final inputs to NN are balanced with regular (non-fraud) and fraud transactions. Otherwise, the learning / generalization will not be good and there would be overfitting.

One way to achieve this would be identifying the last 2 features of the feature vector and checking whether those are empty (or zero):

Input Data **vector, v** = {amount of money, shop, time, date, **when stolen**, ***the transactions performed on the stolen card***}

With such modified data input we can use any competitive learning NN (like K-Means NN or SOM) to learn and cluster the data into Normal and Fraudulent classes. Of course, we would need to normalize the data to match the data in the weight space.

**More Details:**

* 1. Supervised learning is difficult and usually inaccurate as labeling of data is not usually correct. When someone steals the credit card data and uses it, the owner does not detect it immediately unless the person checks bank statement everyday. Also, it takes time before the merchant reports it to the bank. So, data labeling for fraud usually have a good offset of time. This makes error in learning the fraudulent transactions. Besides, there are lot more regular data than abnormal (fraud) data. So, a Supervised learning will mainly learn the good data and not the abnormalities (similar to Intrusion detection problem mentioned above).
  2. Hence, unsupervised learning is a better approach in general. Challenge is to indirectly detect the outliers as it is mainly a outlier detection. From this aspect, competitive learning (with k-means NN or SOM) can do a better job as the weights can learn the normal pattern (after good normalization) well. So, any abnormal pattern with small change (of course, beyond noise) could be detected (like the intrusion case discussed above).
  3. Challenge with Unsupervised Competitive Learning – depending on the outliers, it would be hard to detect noise versus real outliers. One solution is to improve the clusters by some supervised learning when we have good labeled data.
  4. Supervised learning may still be used as mentioned above for Q #2 - **See Credit Card Fraud Detection Note for Project and labs.**

1. Consider the following data set consisting of the scores of two variables on each of seven individuals (subjects). This data set is to be grouped into two clusters using k-Means clustering algorithm.



Initialization - .  As a first step in finding a sensible initial partition, let the A & B values of the two individuals furthest apart (using the Euclidean distance measure), define the initial cluster means, giving:



Complete the iterations needed to finalize your answers (maximum 5 iterations will suffice).

Ans:

The Euclidean distance of person 2 from Group 1 is sqrt((1.5-1)^2 + (2-1)^2) = 1.11

The Euclidean distance of person 2 from Group 2 is sqrt((1.5-5)^2 + (2-7)^2) = 6.1

Hence, person 2 is assigned to group 1.

Doing similar computations for other person, we get the following after first iteration:



The centroid is computed by taking the average of x and y i.e. (1 + 1.5 + 3) / 3 = 1.8 for the first group and (5+3.5 + 4.5 + 3.5)/4 = 4.1 (Note person 3 can be assigned to either group).

**Repeat the Process:**

Now repeat the process, considering the new population and new centroid. The next iteration will yield the following result:



In fact, the result will not change after more iterations. So, we can consider this as the final solution.