Coursework on Distance Metrics and Neural Networks

Ashish Pandey
Dept. of Electrical and Electronics Engineering
Imperial College London
ashish.pandey17@imperial.ac.uk
CID-01383450; Login-ap8516

Ilias Chrysovergis
Dept. of Electrical and Electronics Engineering
Imperial College London
ilias.chrysovergis17@imperial.ac.uk
CID-01449042; Login-ic517

Abstract

Clustering and Classification are two important techniques used to categorize data into groups. Complexity increases when data sets are very similar, making the task of classification challenging. Distance metrics play a very important role in accomplishing this task because often choosing a suitable metric leads to translating the problem into a much simpler one. In this report we use multiple distance metrics alongside Nearest Neighbor classifier to classify given Wine dataset. Metrics are compared on the basis of resulting classification error. Further, we make use of K-means to reduce the complexity of Nearest Neighbor classifier and present a comparison in our report. Neural Networks have been used to successfully solve complex problems in multiple domains because they are very good at pattern recognition problems. Using Neural Network Toolbox in MATLAB© we train our designed network and test it with samples in wine dataset. Associated parameters are varied to obtain different results and a discussion on results is thereafter presented in our report.

1. Introduction

In our report we make use of Wine dataset provided in coursework to accomplish various objectives. Provided dataset has 178 data points or samples of wines grown in the same region in Italy. Chemical analysis of each of these samples gives us 13 dimensions or attributes. These attributes are: Alcohol, Malic acid, Ash, Alkalinity of ash, Magnesium, Total phenols, Flavonoids, Nonflavonoid phenols, Proanthocyanins, Color Intensity, Hue, OD280/OD315 of dedulted wines, and Proline. Data used is pre-classified in one of three classes 1,2 and 3.

This report has 5 sections. Section 1 introduces the dataset we use throughout this report. In section 2, we present the results of nearest neighbor classification experiment by employing different distance metrics and a comparison is presented between different metrics by using classification accuracy as parameter. Section 3 aims to reduce the complexity of experiment performed in section 2 by using K-means algorithm and in section 4 we design a neural network, train and test it using our wine dataset. Section 5 concludes our report. In each section an attempt has been made to provide insight and reason behind the results obtained on varying associated parameters. Experiments have been performed using MATLAB R2017b on Computer with i7 processor and 7.87 GB RAM.

2. Distance Metrics

Distance metrics have a very crucial role in measuring the similarity or regularity amongst multiple data-points/ samples. Before proceeding to solve a problem using techniques of Pattern Recognition, it is very important to determine the similarity or dissimilarity between different samples and understand how we can compare these data-points. In essence, main objective is to find a similarity function that helps to reduce the complexity of our original problem of pattern recognition. It is imperative to mention that it is not necessarily true that a particular metric applied to a problem will yield similar good result when used on another problem. A proper metric should satisfy certain properties [1].

In our report, to perform experiment of Pattern Recognition on Wine Data using nearest neighbor classifier [2], we calculate the classification error as a fraction of incorrectly classified test points for multiple distance metrics. The table given below presents the division of samples into different classes and between testing and training set.

Table 1: Division of Wine Data in classes & testing, training set.

	Class-1	Class-2	Class-3
Training Set	39	51	28
Testing Set	20	20	20

First, we proceed with un-normalized data and calculate the classification error by using different distance metric in nearest neighbor classifier. Figure 1 illustrates the variation in classification error with different distance metric for normalized & un-normalized data. An essential point to be noted is that technically Kullback-Leibler Divergence is not a metric because it does not satisfy the property of symmetry. Square root of distance using Jensen Shannon Divergence which build upon KL is a metric.

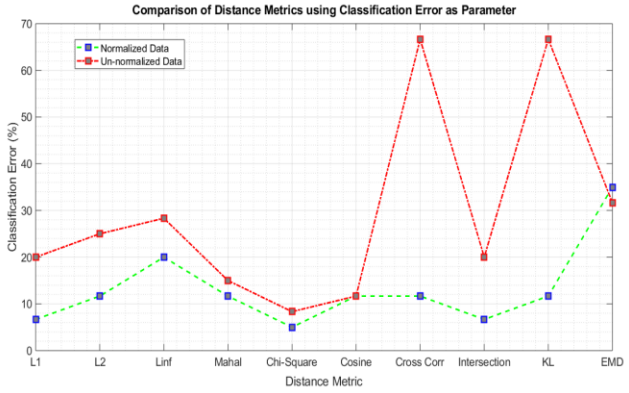

Figure 1: Classification Error for different metrics using Normalized and Un-normalized data.

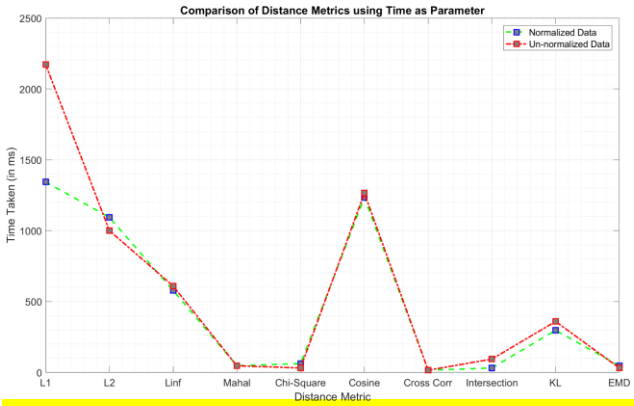

Figure 2: Time Taken for classification using different metrics for Normalized and Un-normalized data.

A comparison of classification error depicted in figure 1 shows that for each metric except EMD, we obtain a lower classification error by normalizing provided wine dataset before proceeding with our experiment. The reason for this observation is that normalization does not allow a/any particular dimension/s with comparatively bigger value to outweigh other dimensions while computing similarity/ distance. We obtain a higher classification error using EMD with normalized data because normalization leads to sets with differently scaled but otherwise identical density functions to become indistinguishable [3]. We observe that using Chi-Square distance metric we get minimum classification error of 5% and 8.33% for normalized and un-normalized data values respectively. Similarly, for normalized data value we get maximum error of 36% using EMD metric and maximum error of 68.33% for KL and cross-correlation distances with un-normalized data. Using time taken for classification of all testing samples as a parameter, we compare different metrics in figure 2. We obtain comparatively very large classification time while using L1, L2, Chessboard and Cosine metrics. Time taken for classification using Mahalanobis, Chi-Square, Cross-Correlation and EMD metrics is ~ 10 ms.

3. K-means Clustering

Since classification has been a topic of immense research in different domains, in this section we will make use of K-means Clustering technique [4] to reduce the complexity of classification performed in previous section. As samples and dimension of samples increases, both time and memory assume critical role in classification problems. K-means has been popular due to its scalability and efficiency.

Using aforementioned approach, we proceed by grouping wine data into three clusters using the K-means and the centroids of clusters hence formed is calculated. These centroids are used as the most representative features of each of the three classes. Distance is calculated by making use of metrics discussed in previous section, but now only three training data are provided, in contrast to the previous section, where 118 features were used. This leads to an increased classification error but lower CPU's utilization, since the complexity and number of calculations to be performed is reduced using K-means approach.

An important fact to be considered is the randomness of K-means algorithm. Since, the first three centroids are randomly chosen, the algorithm performs in a different way according to that choice. Therefore, an iterative algorithm was implemented to achieve the best clustering according to the provided labels. The smallest error achieved was 37 out of 118 for normalized data and 38 out of 118 for unnormalized data.

Using three centroids that provide the smallest clustering error, different distance metrics are compared using classification error and time performance as parameters of interest. Figure 3 and figure 4 illustrate this comparison. As mentioned above we have a higher classification error, because less data points are used with this approach. Although we have a higher classification error, time performance has improved considerably. Time taken for classification with K-means for metrics like L1, L2, Linf and cosine is nearly ten times smaller than time taken without using K-means algorithm.

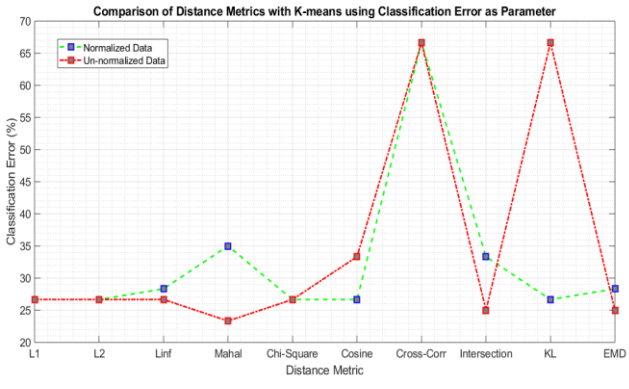

Figure 3: Classification Error for different metrics with K-means using Normalized and Un-normalized data.

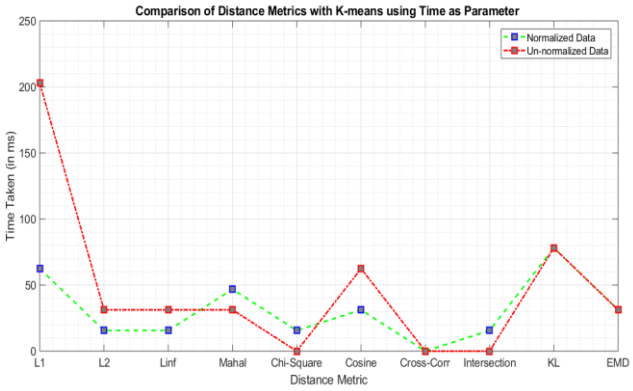

Figure 4: Time Taken for classification using different metrics with K-means for Normalized and Un-normalized data.

K-means has a trade-off between classification error and time performance. Classification error can be reduced by increasing the number of clusters.

4. Neural Network

Neural Networks have been very popular in solving problems of pattern recognition. We design a neural network using MATLAB©'s Neural Network Toolbox and train it using wine dataset and then test it to observe effect of varying different parameters associated with this toolbox. The toolbox by default normalizes input data using *mapminmax* before proceeding to perform training or testing. Figure 5 depicts the variation in classification accuracy on varying number of layers (1 to 4; step-size of 1) and simultaneously the number of neurons per layer (5 to 25; step-size of 5). We also see the effect of using a different algorithm in our designed neural nets. Our toolbox implements Levenberg-Marquardt optimization as *trainlm* algorithm and it is the fastest backpropagation algorithm in the toolbox [5]. Similar plots for other backpropagation algorithms can be found in Appendix-I of this report. Classification accuracy obtained for best set of values is included in the plots.

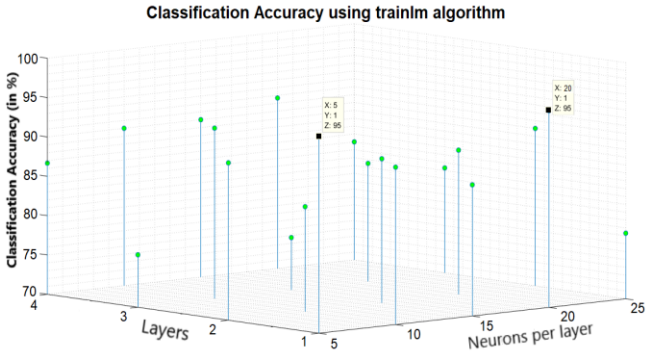

Figure 5: Classification Accuracy with variation in number of layer and variation in number of neurons per layer using trainlm algorithm for Neural Network.

There has been ongoing research to optimize hyperparameters associated with a neural network to obtain best result for a given dataset. From a beginner's perspective brute force technique is useful in optimizing few important hyperparameters.

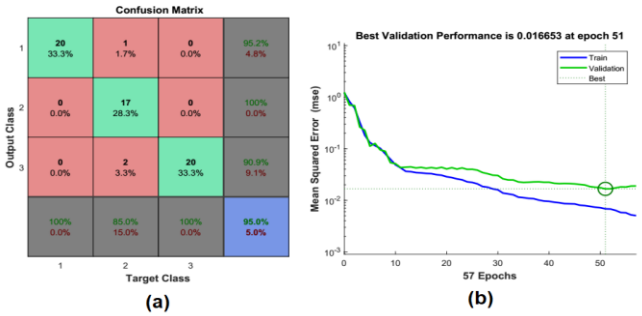

Figure 6: (a) Confusion Matrix — 1 Layer with 20 neurons using trainlm algorithm (b) Performance of Neural Network— 1 Layer with 10 neurons using trainscg algorithm.

Three important hyperparameters are Initial Learning Rate, Decay Rate and Regularization Parameter. The effect of varying these on classification accuracy can be observed in figure 7 below. As evident in figure 7(b), taking a large value for regularization parameter results in overfitting which reduces our classification accuracy for values > 0.5.

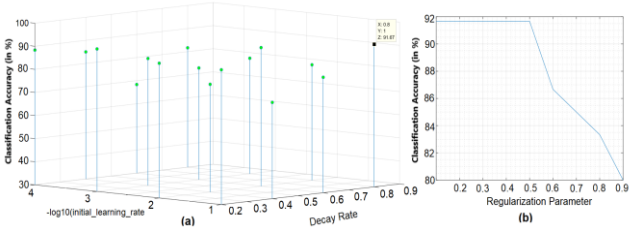

Figure 7: (a) Classification accuracy with variation in initial learning rate (0.1, 0.01, 0.001, 0.0001) and simultaneous variation in decay rate (0.1 to 0.9; step size of 0.1)— 1 Layer with 10 neurons using trainlm algorithm (b) Classification accuracy with variation of regularization parameter (0.1 to 0.9; step size of 0.1)— 1 Layer with 10 neurons using BFGS algorithm.

We do not want to choose a very high initial learning rate because this prevents our network from converging and learning to spot patterns and trends in given data. We specify decay rate to decrease the learning rate with increasing epochs i.e. as our network begins to learn.

5. Conclusion

In our report several distance metrics have been compared using classification error and time as parameters. K-means algorithm has also been implemented using multiple distance metrics. We observe that proceeding with normalized data gives better results. Chi-Square metric yields lowest classification error amongst all metrics used.

We also present a brief study on neural networks in our report.

References

[1] Krystian Mikolajczyk, Pattern Recognition Lecture Notes, 2017, Imperial College London.

[2] Wikipedia, Nearest Neighbour Algorithm, Date Accessed: 19/12/2017,https://en.wikipedia.org/wiki/Nearest_neighbor_algorithm.

[3] A. Gardner, On the Definiteness of Earth Mover's Distance and Its Relation to Set Intersection, in IEEE Transactions on Cybernetics, vol. PP, no. 99, pp. 1-13.

[4] J. B. MacQueen, Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297.

[5] Mathworks, trainlm, Date Accessed: 21/12/2017, https://uk.mathworks.com/help/nnet/ref/trainlm.html.

APPENDIX-I

This appendix illustrates the plots of classification accuracy for our neural network.

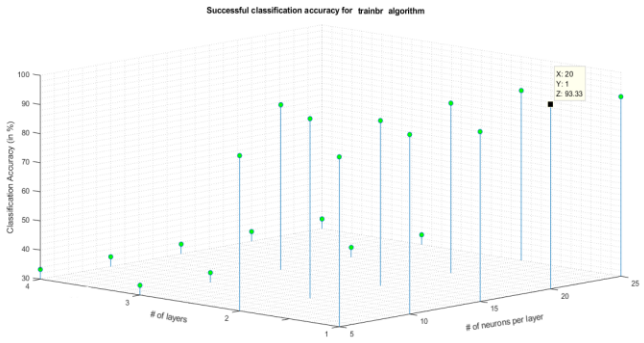

Figure A-1: Classification Accuracy with variation in number of layer and variation in number of neurons per layer using trainbr algorithm for Neural Network.

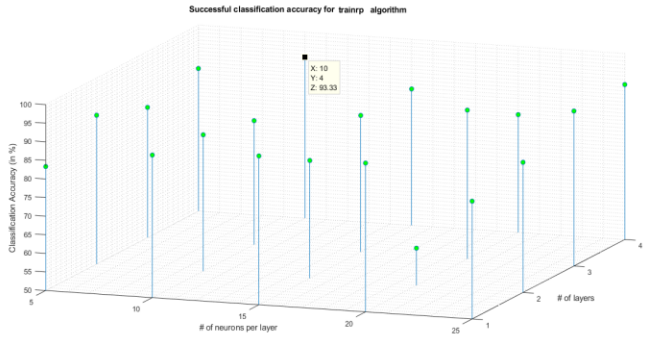

Figure A-2: Classification Accuracy with variation in number of layer and variation in number of neurons per layer using trainrp algorithm for Neural Network.

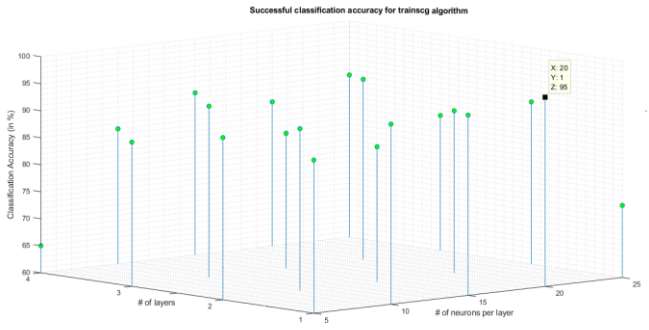

Figure A-3: Classification Accuracy with variation in number of layer and variation in number of neurons per layer using trainscg algorithm for Neural Network.

In each plot show in this appendix we show the effect of varying number of layers (1 to 4; step-size of 1) and number of neurons per layer (5 to 25; step-size of 5) on classification accuracy for our designed neural network. Every plot uses a different backpropagation algorithm.