International Electronic Engineering Mathematical Society IEEMS
http://www.ieems.org

In coloration with
Institute for Mathematics, Bio-informatics, Information-technology and Computer-science IMBIC
International e-Journal of Numerical Analysis and Related Topics
http://www.ieems.org/iejenart.htm
Volume (5), September 2010, pp. 1-8
ISSN: 1687-787X

APPLICATION OF PROBABILISTIC NEURAL NETWORK FOR WINE CLASSIFICATION

Madhusree Kundu[1]and Naga Chaitanya Kavuri[2]
Department of Chemical Engineering,
National Institute of Technology,
Rourkela,
India
Corresponding Author: Associate Professor, E-mail: mkundu@nitrkl.ac.in
Phone: +91661-2462263,
Fax: +91661-2462999
[2]M.Tech Research Student,
E-mail: biochaitanya@gmail.com
India

ABSTRACT

Monitoring of a product quality & controlling it for ensuring certain standards using multivariate statistics has become almost a norm in food and beverage industry. A pattern recognition tool, principal component analysis (PCA) was applied to discriminate reliably among 178 samples of wine possessing 13 number of feature variables. *K-means* clustering, a supervised clustering technique was used to designate the classes available among the wine samples with the help of first two principal components. Hierarchical clustering technique was also applied to classify them with a mention of their classification level in the produced dendrograms. A classifier was developed using probabilistic neural networks (PNN) which can help in online process motoring.

INTRODUCTION

Classification is one of the most important decision making task exercised by us. The determination of quality of food stuffs, water and beverages and maintenance of their correspondence to standards needed to be attended and ensured. Statistical Quality Control (SQC) was designed to sample a large population on an infrequent basis. In recent years, the SQC techniques which worked well for final product quality control; are now being applied to materials and operating conditions of a process. The procedure is now known as SPC or Statistical Process Control. The classification aspect, an integral part of SPC decides the attribute; ‘quality’ of the product to a predefined class based on the parametric values or features that influence the product quality. Those parametric values can be determined by various classical analytical techniques; such as various chromatography and spectrometry. However, they are time-consuming, expensive, and laborious which can hardly be done on-site or on-line. For quality control, it is necessary to monitor a group of certain features that reflects the quality of the product such as ageing and spoilage for a beverage product. These components can be numerous or unknown and the problem appear to be quite difficult.

Besides, it is impractical and very hard to correlate and compare the results of instrumental analysis to biological sensing [1]. The use of sensor arrays for producing features followed by the multivariate data analysis (MVDA) and different clustering techniques to discriminate among various samples paves the way to a successful design of a classifier. The use of various decision rules qualifies the classifier to be used for authentication purpose. Discrimination and classification of the feature variables produced from multi-sensory array owe a profound debt to the multivariate statistics these days. In these procedures, an underlying probability model must be assumed in order to calculate the posterior probability upon which the classification decision is made. One major limitation of the statistical models is that they work well only when the underlying assumptions are satisfied. The effectiveness of these methods depends to a large extent on the various assumptions or conditions under which the models are developed. Users must have a good knowledge of both data properties and model capabilities before the models can be successfully applied. Discrimination is concerned with separating distinct sets of objects (or observations) on a one-time basis in order to investigate observed differences when casual relationships are not well understood. The operational objective of classification is to allocate new objects (observations) to predefined groups based on a few well defined rules evolved from discrimination analysis of such kind of allied group of observations. Neural networks, either supervised or unsupervised have emerged as an important tool for classification. The recent vast research activities in neural classification have established that neural networks are a promising alternative to various conventional classification methods. The advantage of neural networks lies in the following theoretical aspects. First, neural networks are data driven self-adaptive methods in that they can adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model. Second, they are universal functional approximators in that neural networks can approximate any function with arbitrary accuracy [2-4]. Since any classification procedure seeks a functional relationship between the group membership and the attributes of the object, accurate identification of this underlying function is doubtlessly important. Third, neural networks are nonlinear models, which makes them flexible in modeling real world complex relationships. Finally, neural networks are able to estimate the posterior probabilities, which provide the basis for establishing classification rule and performing statistical analysis [5]. The present work is based on application of principal component analysis (PCA) for discrimination and hierarchical as well a non-hierarchical clustering techniques for the clustering among the 178 numbers of available wine samples on the basis of certain selected features capable of qualifying their process of aging and spoiling etc.. A classification model for the wine quality monitoring has been developed using *Probabilistic Neural Networks* (PNN).

PCA & K-MEANS CLUSTERING

A wine dataset of 178 numbers of samples containing 13 numbers of features, like proline, magnesium content. Ash_ alkalinity, pH etc was considered. The data matrix was auto-scaled before processing. PCA is a multivariate statistical technique that can extract the essential features from a data set by reducing its dimensionality without losing any valuable information of it. Principal components (PCs) are a new set of coordinates orthogonal to each other. The first PC is the direction of largest variation in the data set. The projection of original data on the PCs produces the score data or transformed data as a linear combination of those fewer mutually orthogonal dimensions. PCA technique was applied on the auto-scaled data matrix to determine the principal eigenvectors, associated Eigen values and score or the transformed data set. Clustering technique is more primitive in that no a-priori assumptions are made regarding the group structures. Grouping of the data can be made on the basis of similarities or distances (dissimilarities). Hierarchical clustering techniques proceed either by a series of successive mergers or a series of successive divisions. Agglomerative hierarchical methods start with individual objects ensuring as much number of clusters as objects initially. Besides Hierarchical clustering, non- hierarchical method, *K-means* clustering was also applied in this work. The number of clusters *K* can be pre-specified or can be determined iteratively as a part of the clustering procedure. The K-means clustering proceeds in three steps, which are as follows,

1. Partition of the items in to *K* initial clusters.
2. Assigning an item to the cluster whose centroid is nearest (distance is usually a Euclidian). Recalculation of the centroid for the cluster receiving the new item and for the cluster losing that item.
3. Repeating of the step-2 until no more reassignment takes place or stable cluster tags are available for all the items.

The *K-means* clustering has a specific advantage of not requiring the distance matrix as required in hierarchical clustering, hence ensures a faster computation than the latter. Once the clustering had been done, the data (13attributes of each sample and their corresponding cluster description) was divided in to training sets and testing sets for the neural networks.

PROBABILISTIC NEURAL NETWORKS (PNN)

The *probabilistic neural network* (PNN) was introduced by Donald Specht (1990) [6] and it can be used for classification problems. The multilayer feed forward network can be used to approximate non-linear functions where the network structure is sufficiently large. Any continuous function can be approximated by carefully choosing the parameters in the network. For the determination of those highly non-linear parameters learning should be based on non-traditional optimization techniques. A viable alternative is the radial basis function neural network. A RBF network uses radial basis functions as activation functions. Radial basis function (RBF) networks typically have three layers: an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer as shown in Fig.1. In the basic form all inputs are connected to each hidden neuron. The inputs vector $X = [x_1, x_2, x_3, ...x_n]$ is applied to the neurons in the hidden layer. Each hidden layer neurons computes the following exponential functions, called Gaussian response function as given by:

$h_i = \exp[-\frac{D_i^2}{2\sigma^2}]$, where X=an input vector (1)

$D_i^2 = (x - u_i)^T (x - u_i)$= squared distance between the input vector and training vector. (2)

u_i=weight vector of hidden layer neuron i. The weights of each hidden layer neuron are assigned the values of an input training vector. The output neuron produces the linear weighted summation of these as given by:

$y = \sum h_i w_i$, where w_i= a weight in the output layer. (3)

Sometimes the outputs are optionally normalized according to the following formula that divides the output of each neuron in the output layer by the sum of all hidden layer outputs. i.e., $out_i = \sum_i h_i w_i / \sum_i h_i$

Thus the output has a significant response to the input only over a range of values of spread parameter, called the receptive field of the neuron, the size of which is determined by the value of σ as shown in figure (1).

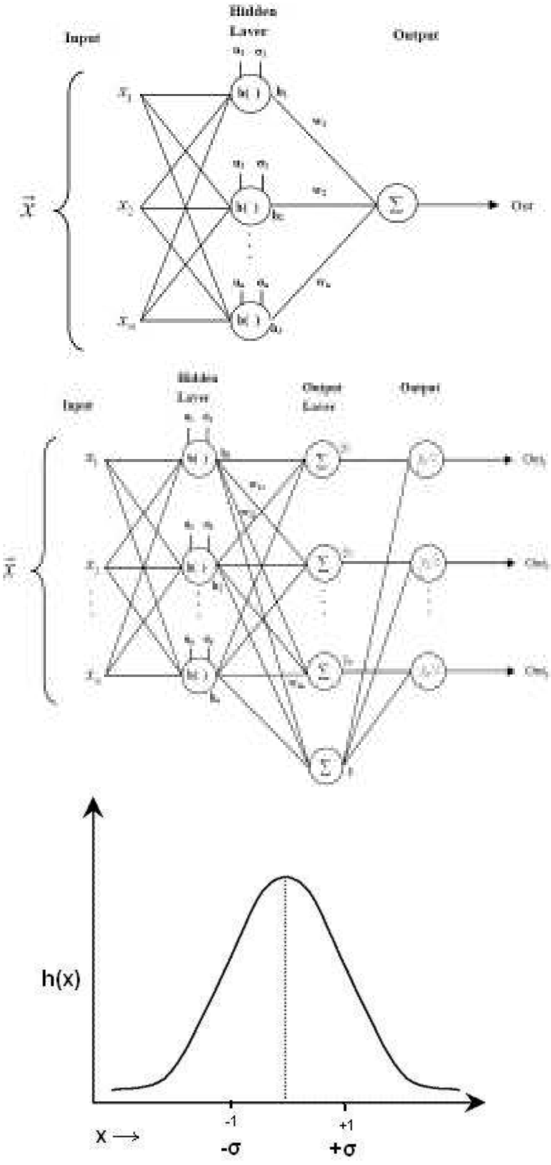

Figure 1: Radial Basis network architecture

PNN is having its theoretical foundation on Bayesian classifier theory. PNN being basically a Radial basis neural network (RBNN) containing an extra competitive layer in addition to the radial basis layer. When an input is presented, the first layer computes distances from the input vector to the training input vectors and produces a vector whose elements indicate how close the input is to a training input. The second layer sums these contributions for each class of inputs to produce as its net output a vector of probabilities. Finally, a competitive transfer function on the output of the second layer picks the maximum of these probabilities, and produces a ‘1’ for that class and a ‘0’ for the other classes. The basic architecture of PNN is presented in figure (2).

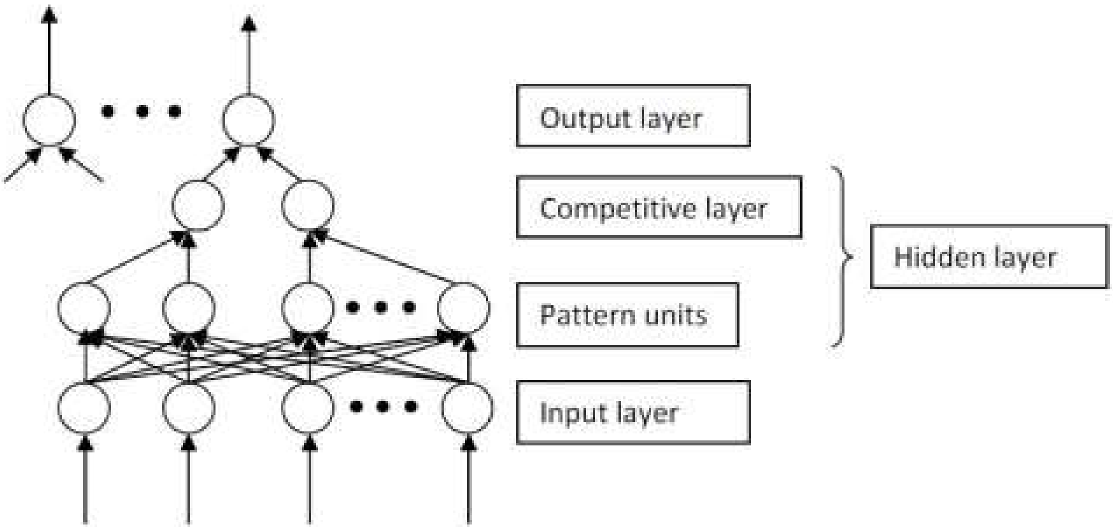

Figure 2: Basic architecture of Probabilistic neural networks (PNN)

The basis of PNN is Bayes theorem, which states:

$$P(y^i|x) = P(x|y^i)P(y^i)/P(x) \tag{4}$$

Let y denote the membership variable that takes a value of y_i if the object belongs to group i. $P(y^i)$ is the prior probability of class i and $P(x|y^i)$=Probability density function. $P(y^i|x)$ is the posterior probability of group i . $P(x)$ is the probability density function:

$$P(x) = \sum_{j=1}^{N} P(x|y^i)P(y^i) \tag{5}$$

In PNN, the probability distribution function is approximated by Parzon windows, typically using the exponential function.

By the previous definitions,

$$P(x|y^i) = (\frac{K}{N^i})\sum_{j=1}^{N^i} \exp(-\frac{D_j^{i2}}{2\sigma^2}) = (\frac{K}{N^i})\sum_{j=1}^{N^i} h_j^i \tag{6}$$

where $K = \frac{1}{2\pi^{d/2}\sigma^d}$ = the scaling factor to produce a multidimensional unit Gaussian, $D_j^{i2} = (x-u_i)^T(x-u_i)$ = the squared Euclidian distance between the current input x and training vector j in class i. Applying Bayes theorem to calculate the conditional probability of x for each class, then summing these over all classes yields,

$$P(x) = \sum_{i=1}^{N_i} P(x|y^i)P(y^i) =$$

$$\sum_{i=1}^{N_i} (\frac{K}{N^i})\sum_{j=1}^{N_i} \exp[(-\frac{D_i^2}{2\sigma^2})(\frac{N^i}{N})] = \frac{K}{N}\sum_{i=1}^{N_i}\sum_{j=1}^{N_i} h_j^i \tag{7}$$

where N_i is the number of classes. Thus the inner summation adds all hidden layer neuron outputs associated with class i; the outer summation counts these over all classes. The double summation may be eliminated by simple counting all hidden neuron outputs. If we have an object with a particular feature vector x and a decision is to be made about its group membership, the probability of classification error is

$$P(error/x) = \sum_{i \neq j} \left(P(y^i|x) \right) = 1 - P(y^i|x),\ if\ \ y^i\ \ is\ already\ decided \tag{8}$$

Hence, if the purpose is to minimize the probability of total classification error, then the following is the widely used Bayesian classification rule,

Decide y^i for x if:

$$P(y^i|x) = \max_{i=1...N} P(y^i|x) \tag{9}$$

For developing a PNN based classifier, the whole dataset containing 178 samples of 13 attributes along with their cluster numbers has been redistributed into six randomly selected datasets containing 20%, 30%, 40%, 50%, 60% and 70% of data. The random selection of data was done using the method described by Box and Muller (1958) and Devroye (1986) [7, 8]. The application of *K-means* clustering & design of PNN classifier were done using MATLAB 7.6 platform.

RESULTS AND DISCUSSIONS

The application of PCA on the auto-scaled matrix containing 178 samples of wine resulted in 5 numbers of principal components (PCS). Table 1, which is a list of Eigen values corresponding to the column vectors of the loading matrix, reveals that the features (variables) like proline, magnesium, ash-alkalinity, color intensity, and maleic acid has captured almost 80 % of the variance of the data thus become the dominant PCS.

Table (1): Principal Eigen values and percentage variance captured by principal components

Component	**PC1**	**PC2**	**PC3**	**PC4**	**PC5**
Eigen values	4.706	2.497	1.446	0.919	0.853
% Variance	36.19%	19.20%	11.12%	7.06%	6.50%
Cumulative % Variance	36.19%	55.40%	66.53%	73.59%	80.16%

To represent the score data the bi-plot (Fig.3) is chosen, which clearly shows the presence of three clusters in the data. The stable *K-means* statistics of the score along pc1-pc2 are presented in Table 2, which presented all the 3 cluster centroids, the number of data points pertaining to each cluster with a mention to the individual sample numbers belonging to that cluster.

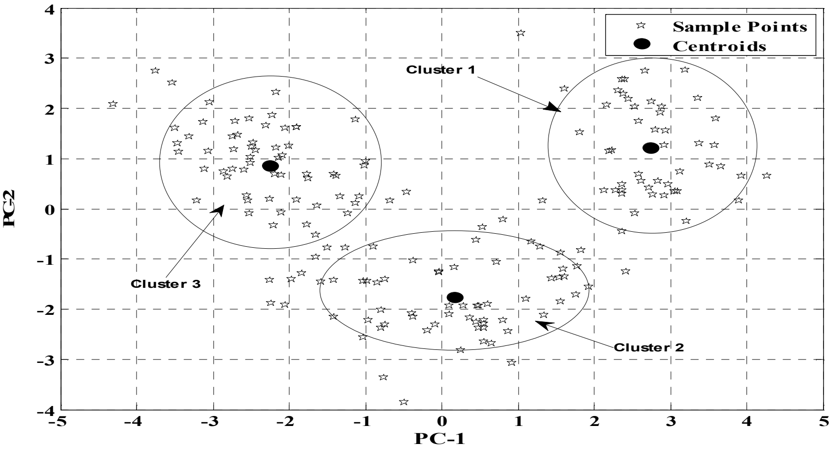

Figure 3: Discrimination and Clustering of scores along PC1-PC2

Table (2): Statistics of *K-means* Clustering (Score along pc1-pc2)

Cluster Identity	No. of Samples	Cluster Centroid
1	49 (84,131-178)	(2.7362, 1.2108)
2	65 (60-71, 73, 75-83,85-95,97-98,100-121,123-130)	(0.1623, -1.7626)
3	64 (1-59, 72, 74,96,99,122)	(-2.2598, 0.8632)

Hierarchical clustering technique was also applied to classify them with a mention of their classification level in the produced dendograms. As a part of hierarchical clustering, the distance matrix or a dissimilarity matrix was determined, which was symmetric along the diagonal (all the diagonal elements are zero). A hierarchical cluster tree was then created with that distance matrix to form the dendrogram figure (4) originated from the score vectors along pc1- pc2.

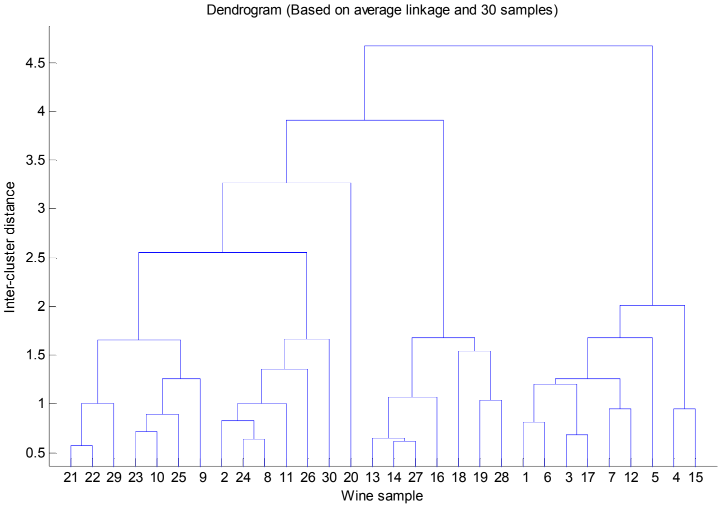

Figure 4: Dendrogram on score along PC1-PC2

Dendrogram consists of many U-shaped lines connecting objects in a hierarchical tree. The height of each U represents the distance between the two objects being connected. If there were 30 or fewer data points in the original dataset, each leaf in the dendrogram corresponds to one data point. If there were more than 30 data points, the complete tree can look crowded, and dendrogram collapses lower branches as necessary, so that some leaves in the plot correspond to more than one data point. The dendrograms were created using 1-30 numbers of samples for its clarity in representation. Fig. 4 demonstrated that over an inter-cluster distance of 4.5 there will be the merger of all the data points (samples) in one group. At a inter-cluster distance of 3.5, there are 3 classes present in the transformed data along pc1- pc2 coordinates. The performance of PNN network developed is presented in Table 3. Among the six randomly selected data sets, the network was trained by using each dataset and tested against the other remaining datasets. The results clearly indicate that sufficient training is required for a good accuracy in prediction. Accuracy was defined as classification or misclassification percentage.

Table (3): Performance of the PNN network

		Training Set					
		20%	**30%**	**40%**	**50%**	**60%**	**70%**
% Accuracy for test sets	**20%**	--	55.88235	69.69697	66.66667	78.78788	90.90909
	30%	43.75	--	64.58333	70.83333	70.83333	83.33333
	40%	48.78049	56.09756	--	71.95122	79.26829	84.14634
	50%	47.31183	57.6087	70.65217	--	72.82609	83.69565
	60%	46.46465	49.49495	71.71717	69.69697	--	85.85859
	70%	47.72727	54.54545	68.18182	71.21212	74.24242	--

CONCLUSIONS

The wine data base containing 178 no of samples was reduced to 5 principal dimensions according to PCA analysis. The scores of pc1-pc2 were used for K-means clustering of the present data base, and existence of three clusters was concluded. Once the clustering had been done, the data (13attributes of each sample and their corresponding cluster description) was divided in to appropriate training and testing sets for the PNN. Table 3 shows that with the increase in the ratio of number of training data to number of testing data, the efficacy of classification becomes better. This particular exercise demonstrated the importance and need of sufficient number of training data for the efficient functioning of PNN based classifier.

REFERENCES

(1) Legin, A., Rudnitskaya, A., Vlasov, Y., Natale, C.D., Davide, F. and D’Amico, A. “Tasting of beverages using an electronic tongue”, *Sensors and Actuators,* vol. *B* 44, pp 291-296, 1997.

(2) Cybenko, G. “Approximation by superpositions of a sigmoidal function”, *Math. Contr. Signals Syst.*, vol. 2, pp. 303–314, 1989.

(3) Hornik, K. “Approximation capabilities of multilayer feed forward networks”, *Neural Networks*, vol. 4, pp. 251–257, 1991

(4) Hornik, K., Stinchcombe, M. and White, H. “Multilayer feed forward networks are universal approximators”, *Neural Networks*, vol. 2, pp. 359–366, 1989.

(5) Richard, M.D. and Lippmann, R. “Neural network classifiers estimate Bayesian *a posteriori* probabilities”, *Neural Comput.*, vol. 3, pp. 461–483, 1991.

(6) Specht, D. F. “Probailistic Neural Networks”, *Neural Networks*, vol. 3, pp 109-118, 1990.

(7) Box, G. E. P. and Muller, M. E. “A note on the generation of random normal deviates”, *Annals of Mathematical Statistics*, vol. 29, pp 610-611, 1958.

(8) Devroye, L. “Non-uniform random variate generation”, Springer, New York, 1986.