IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Principal Component Analysis on Mixed Data For Deep Neural Network Classifier in Banking System

Chittem Leela Krishna[1*], Poli Venkata Subba Reddy[2]

[1,2]Dept. of CSE, S.V.U College of Engineering, S.V. University, Tirupati, India

Corresponding Author: chittem.leelakrishna03@gmail.com, Mobile: +91-86863-91266

DOI: https://doi.org/10.26438/ijcse/v7i5.129134 | Available online at: www.ijcseonline.org

Accepted: 12/May/2019, Published: 31/May/2019

Abstract— Data stored in repositories are rapidly growing in terms of instances represented with multiple attributes/dimensions. To represent characteristics of an instance, mixed type attributes are used. Banking System is one of the areas which store information of bank customers in multiple dimensions. Principal Component Analysis (PCA) is a Dimensionality reduction technique in Data Mining used to transform the attributes of a dataset to a lesser dimensional space. Classification is a Supervised Machine Learning technique used to distinguish the instances of a dataset into a number of classes. In this work, we have analyzed the Bank Marketing dataset containing 1000 instances of bank clients represented with 17 attributes, with a class label as the last attribute. Principal Components (PCs) are generated from the input dataset by applying PCA on mixed attributes. A Deep Neural Network classifier is built by applying Backpropagation on the PCs. Experimental results show that our proposed PCA mixed Deep Neural Network classifier outperforms existing classifiers in terms of accuracy.

Keywords— Banking System, Mixed Data, PCA, Classification, Backpropagation, Deep Neural Network

I. INTRODUCTION

Data mining is a knowledge discovery process from the data stored in huge repositories at different locations [1]. The knowledge can be obtained in the form of patterns so that they can be analyzed with different visualization tools [1]. Data warehouse is a repository that stores huge amounts of data from multiple heterogeneous sources such as Databases, Flat-files, The Web, etc. To store data belonging to different sources in a warehouse, there is a need for data preprocessing [1]. The four major preprocessing techniques used in data mining are Data Cleaning, Data Integration, Data Selection and Data Transformation [1]. In this paper, we focus on Principal Component Analysis [2], which is a data dimensionality reduction technique.

Once the data is preprocessed, mining tasks such as Association Rule Mining [2], Classification [2], Regression [2], Clustering [2], etc., are to be applied to it depending on the problem to be solved for decision making [2]. This paper focuses on Classification, which is a supervised form of Machine Learning [3]. The process of classification distinguishes records of a dataset into a predefined set of classes in a two-step fashion. The first step is to build a classifier/model by the classification algorithm [3], from the training set (a subset of the original dataset) containing attributes with their class labels. The second step makes use of the classifier/model to assign class labels to the test set (a subset of the original dataset other than training set). The first step in classification is named as the learning or training phase, whereas the second step is the testing phase. Since classification deals with class labels while building a classifier, it is said to be a supervised machine learning technique.

The remaining sections of this paper are organized as follows: Section 2 provides a brief description of Principal Component Analysis. Section 3 describes a Backpropagation based Neural Network. Section 4 provides details about a dataset belonging to the Banking System. Section 5 provides related work on PCA and Neural Network classifier. Section 6 provides a detailed explanation of Proposed work including Methodology, Data Preprocessing and the process of building a Deep Neural Network Classifier. Experimental Results and discussion are provided in section 7. Section 8 concludes the paper with future directions.

II. PRINCIPAL COMPONENT ANALYSIS FOR DATA DIMENSIONALITY REDUCTION

A. Need for Dimensionality Reduction

Mining huge volumes of data stored in warehouses can be time-consuming. So there is a need for representing the whole data set in a reduced form (or subset) such that the smaller data set reduces variance and maintains the integrity of the original dataset [2]. Mining on the reduced data set produces similar results to that of the original data set with more efficiency.

B. Principal Component Analysis

Principal Component Analysis is a Dimensionality Reduction [2], which uses data encoding techniques for attributes of the original data set to obtain a reduced representation. Dimensionality reduction can be either lossless or lossy, depending on the reconstruction of the original data set from its reduced representation. Principal Component Analysis (PCA) is a lossy dimensionality reduction technique where only an approximated original data set can be reconstructed from its reduced data set.

For a data set containing *m* tuples and *n* attributes (or dimensions), PCA searches for obtaining *p*-dimensional orthogonal vectors from the *n*-dimensional original data set for data representation ($p \leq n$). PCA projects the original data set into a smaller set through dimensionality reduction, even by replacing the attributes and the data with their smaller representations. PCA makes use of various statistical measures such as Covariance matrix, Eigen vectors, Eigen values. The following steps explain the process of PCA:

1. Normalize D so that each dimension will be
within the same range (0-1)
2. Generate Co-Variance matrix among the attributes of the dataset
3. Calculate Eigen values and Eigen vectors from the matrix
4. Evaluate Principal components (PCs) of D as D^1 with high variance in descending order and remove the components with low variance from D^1
5. D^1 is the reduced form of D with the PCs.

III. BACKPROPAGATION NEURAL NETWORK

A Neural Network is a set of interconnected processing nodes/neurons in the form of layers, where each connection is associated with a weight. Every neural network has one input layer, one or more hidden layers, and one output layer. To perform classification by neural networks, the classification algorithm used to build a classifier is Backpropagation [4]. During the learning phase, the neural network tries to predict class labels of the training data by updating weights between the layers. Once the weights are updated, the testing phase predicts the class labels of the unseen test data.

In Backpropagation, the attributes of training data are propagated in the forward direction from input to output layers via hidden layer(s) through the weights and the output is calculated at the output layer. Output values are calculated at both hidden layers and output layer using activation function [4]. The calculated output is compared with the expected output(class label) in the training dataset, namely the error function. Error is propagated in the backward direction from output to input layers via hidden layer(s) and the weights are updated simultaneously.

The process of Backpropagation is explained in below steps:

```
1. Initialize the Network with Random Weights
2. For each tuple in training dataset
3. Propagate input values forward by
    3.1 Calculating output values of hidden layers
    3.2 Calculating output values of output layer
4. Calculate Error at output layer i.e.,
    Error = Expected output-Actual output
5. BackPropagate Error from output to input layers
6. Update all the weights between the layers
7. Repeat from step 2 for n iterations
```

IV. BANKING SYSTEM

Experiments are conducted on a publicly available bank marketing data collected from [14]. The data is used to conduct campaigns for direct marketing with the customers by banking institutions, where customers are contacted by the banking staff through phone calls to subscribe for term deposits. The main goal of the classifier is to predict whether a customer subscribes for a term deposit.

The whole dataset comprises of 45211 instances and 17 attributes, out of which 1000 instances are chosen for our experiment. The dataset is chosen in such a way that there exists a mixed type of attributes i.e., numeric, binary and categorical types. Table 1 presents a list of all the attributes and their types in the bank marketing dataset. Splitting of training and testing data from the whole dataset is done in an 80-20 fashion. i.e., 800 instances are considered for building the classifier and the remaining 200 instances for testing the performance of the classifier.

V. RELATED WORK

A novel approach to predict the closing stock price is proposed by Gao et al. in [5], which uses two-dimensional PCA method applied on Deep Belief Networks, one of the Deep Learning models to achieve high precision in prediction.

Forecasting of stock price by Elman Neural network based on PCA with improved accuracy at faster training speeds is proposed by Shi et al. in [6] and the neural net is compared with Backpropagation network.

Table 1 Attribute Description of Bank Marketing Dataset

S.No.	Attribute Name	Attribute Type
1	Age	Numeric
2	Job	Categorical
3	Marital	Categorical
4	Education	Categorical

5	Default	Binary
6	Balance	Numeric
7	Housing	Binary
8	Loan	Binary
9	Contact	Categorical
10	Day	Numeric
11	Month	Categorical
12	Duration	Numeric
13	Campaign	Numeric
14	Pdays	Numeric
15	Previous	Numeric
16	Poutcome	Categorical
17	Deposit (Predicted)	Binary

Early stage diagnosis of lung diseases using PCA and deep learning classifiers are discussed in [7], where the input data containing 4096 features are reduced to 79 on applying PCA yet producing same accuracy by the classifier.

Feng et al. in [8] presented an SVM based classifier based on PCA to assess credit risk in the banking system and compared its accuracy with SVM and Backpropagation neural net classifiers.

Min in [9] proposed a Fuzzy SVM classifier based on PCA in assessing credit risk and compared the accuracy of the classifier with SVM and Backpropagation neural net classifiers.

Application of PCA in loan granting to the customers in the banking system is discussed in [10], where the components contributing to analyze the credit score are extracted from the features.

VI. PROPOSED WORK

A. Methodology

The process of building a Deep Neural Network classifier by applying PCA on mixed attributes of a bank marketing data starts with the input dataset gathering from a repository and performing basic data preprocessing techniques [1, 2]. Attributes other than numeric type are encoded using a Label Encoder filter and the whole input dataset is to be split into training and test sets. PCA is applied to the training data to generate *n* Principal Components (PCs) (*n*<16 since the input dataset has 16 attributes and one class label attribute). PCA is applied now on the test set to generate the same number of *n* PCs. Training data of *n*-dimensions is applied to the classification algorithm to generate classifier. The Test set is now applied on the classifier to evaluate its performance. Figure 1 presents the complete process of workflow.

B. Input Preprocessing

The bank marketing dataset is collected from the UCI repository [14]. The total number of instances is 45211 with 17 attributes (16 are normal and one is a class attribute), out of which 1000 instances are chosen for our experimentation purpose. Preprocessing of the input dataset includes detecting missing values and noise in the input dataset and replacing them with the mean value of the attribute.

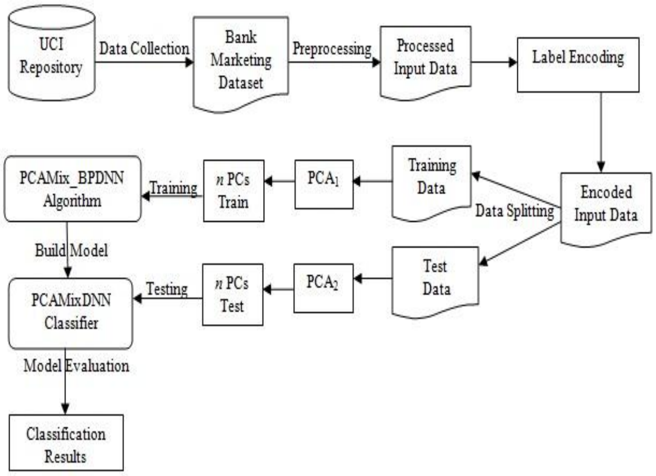

Figure 1 Proposed Workflow Process

C. Label Encoding Mechanism

Label encoder [11] is one of the machine learning encoding techniques used to convert categorical and text data of attributes into numbers. Applying Label encoding on non-numeric attributes help in improving the accuracy of classifiers. Label encoder encodes all the values of each non-numeric attribute by assigning a value between 0 to n-1, where n is the number of total classes of the attribute.

For example, consider a dataset with five instances containing 2 categorical attributes (Country, Gender) and 1 binary attribute (Gender) as in table 2:

Table 2 Sample Dataset for Label Encoding

Age	**Country**	**Gender**	**Job**
45	France	Male	Management
48	Germany	Male	Services
36	Spain	Female	Self-employed
40	Germany	Male	Management
52	Spain	Male	Management
38	Spain	Female	Self-employed

To perform classification on the above data, text values of the three attributes are to be transformed into numbers. Column **Country** has three classes i.e., France, Germany, and Spain. So, Label Encoding on the attribute results in replacing country names with numbers 0,1 and 2 respectively. **Gender** has two classes i.e., Male and Female, and they are replaced with 0 and 1. **Job** has again three classes i.e., Management, Services and Self-employed. Label Encoding on the attribute replaces the text values into numbers 0,1 and 2 as in table 3.

Table 3 Label Encoding of Sample Dataset

Age	**Country**	**Gender**	**Job**
45	0	0	0

48	1	0	1
36	2	1	2
40	1	0	1
52	2	0	1
38	2	1	2

D. PCA on Mixed Attributes

The input dataset showed in table 1 has mixed types of attributes (Categorical, Numeric and Binary). No changes are made to the numeric type attributes. For categorical and text-based binary attributes, Label encoding is performed on the values of the attributes and the text-based values are transformed into numeric values based on the number of classes represented by each attribute. After performing Label Encoding, the encoded input dataset is split into training data and test data based on 80-20 fashion. i.e., 800 instances are treated as training data and 200 instances as test data.

PCA is first applied by fitting the training data to find correlations, covariance among the attributes so as to represent them with a lower number of attributes, namely Principal Components (PCs). A Covariance matrix among the 16 attributes of the training dataset is evaluated. Cumulative variance [12] is calculated from the variance ratio among the components so that the entire dataset can be reduced to the number of components that yield maximum cumulative variance. The process of applying PCA on mixed type attributes is given below:

```
begin PCAMixed ($T_1$:Training data, $T_2$:Test data)
For each non-numeric attribute
        perform Label Encoding
Fit $T_1$ for transformation
For each attribute in $T_1$
        obtain co-variance matrix
        evaluate variance ratio among components
Evaluate Variance Ratio and choose $n$
Obtain $n$ PCs of $T_1$ as $T_1^1$
Fit $T_2$ for transformation
Obtain $n$ PCs of $T_2$ as $T_2^1$
Return $n$, $T_1^1$ and $T_2^1$
end PCAMixed
```

E. Deep Neural Network Classifier

The principal components (*n*) generated by applying PCA on the training data serves as the input layer of the neural network. A multilayer Perceptron is built by using Backpropagation algorithm [13] for classification. The number of hidden layers chosen in the Neural network is 20 with 25 nodes in each layer interconnected with input and output layers. The size of the output layer is two, each representing the two possible values of **Deposit** attribute. i.e., the total neural network size is *n* * (25*20) * 2. Structure of the proposed Deep Neural Network is shown in figure 2.

Due to the depth of hidden layers and considering all the attributes/features for PCA to generate Principal Components with high variance, the neural network is named as a Deep Neural Network. The process of building our Deep Neural network model using Backpropagation is given below:

```
begin BackpropDNN(Data ,n ,out)
Data, n = PCAMixed(Data)
Build a neural network of size n*(25*20)*out
Initialize the network with random node and bias weights,
Initialize momentum as 0.9 and learning rate as 0.001
For each iteration
    For each tuple in Data
        Calculate Input values of hidden layers
        $I_h = \sum I_i W_i$
        Calculate output values of hidden layers
        $O_h = \mathrm{ReLU}(I_h)$
        Calculate Input values of output layer
        $I_o = \sum I_h W_h$
        Calculate output values of output layer
        $O_o = \mathrm{ReLU}(I_o)$
        Calculate Error at the output layer
        Error = Data(class_val)- $O_o$
        Calculate Error at hidden layers
        Update node and bias weights
Repeat for 1000 iterations
end BackpropDNN
```

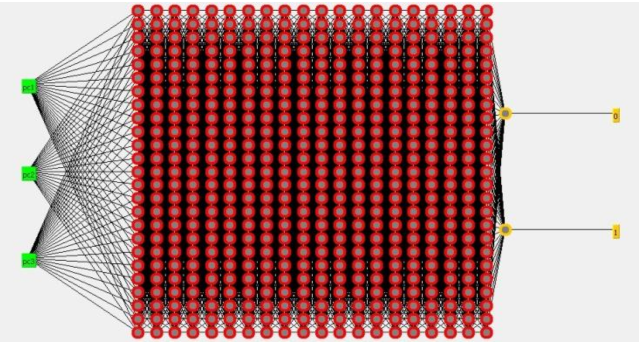

Figure 2 Structure of Deep Neural Network

The proposed Deep Neural network makes use of ReLU [13] (Rectified Linear Unit) activation function, which is widely used in Deep Learning applications. ReLU is a half rectified activation function, meaning all the negative values inputs are changed to zero as given below:

$$R(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

VII. EXPERIMENTAL RESULTS

Experiments are conducted on Bank Marketing dataset collected from [14]. As mentioned, the input dataset contains 45211 instances with 17 attributes (16 are non-class attributes/features and one is a class attribute). Among 45211 instances, we have chosen 1000 instances for our experiment. The algorithms *PCAMix* and *BackpropDNN* explained above are implemented using Python Programming language[15]. All the variances among the 16 attributes generated by

PCAMix are arranged in descending order which shows that the first component has a variance of 98.85% among others, the second and third components have a variance of 1.027% and 0.119% respectively. The cumulative variance among the first three components is calculated i.e., 99.99740%, which suggests that only the top three principal components can represent the whole data with 16 attributes. Hence the value of *n* is chosen as 3 for building the Deep Neural Network classifier.

BackpropDNN is implemented with an input layer representing the Principal components, twenty hidden layers with 25 nodes in each layer and two nodes in the output layer. The total number of iterations for the algorithm to run and backpropagate error, update weights is set as 1000 and Rectified Linear Unit activation function is chosen to eliminate negative values. Once the PCAMixDNN classifier is built, the test dataset is applied on the classifier to evaluate its performance by predicting whether a customer subscribes for a term deposit. The result of evaluation is represented by a two-class confusion matrix [16], from which accuracy [16] is calculated. The structure of a two-class confusion matrix is shown in table 4.

Table 4 Two-class Confusion Matrix

		Predicted class value	
		Negative	Positive
Actual class value	Negative	A	B
	Positive	C	D

Accuracy is defined as the ratio of instances correctly predicted with their actual class to the total number of possible values i.e., instances belonging to a negative class are predicted as negative and positive class instances are predicted as positive by the classifier.

$$Accuracy = \frac{A + D}{A + B + C + D}$$

Table 5 shows the confusion matrix generated by PCAMixDNN classifier. Accuracy of the classifier evaluated from the confusion matrix is 91.00%.

Table 5 PCAMixDNN Classifier Confusion Matrix

Confusion Matrix		Class
178	4	no
14	4	yes

A comparison is made on the performance of PCAMixDNN classifier with existing classifiers namely Decision Tree [19], Naïve Bayes, Support Vector Machines, Logistic Regression [20] classifiers in terms of accuracy and the results are listed in table 6 and a bar graph is plotted in figure 3. The results indicate that our proposed classifier PCAMixDNN outperforms the existing classifiers in terms of accuracy.

Table 6 Accuracy of Classifiers

Classifier	Accuracy
Decision Tree	89.00%
Naïve Bayes	88.50%
Support Vector Machines	89.50%
Logistic Regression	88.50%
PCAMixDNN	**91.00%**

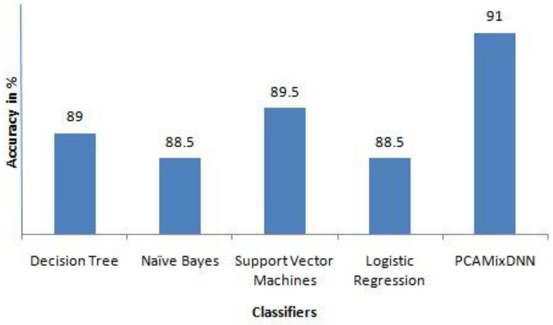

Figure 3 Accuracy Comparison Plot of Classifiers

VIII. CONCLUSIONS AND FUTURE WORK

PCA is a dimensionality reduction technique used for representing the whole input dataset in a reduced form without loss of information. In this work, we have applied PCA on bank marketing data containing mixed type attributes. Among 17 attributes of the input dataset, our PCAMix resulted in producing three Principal Components with a cumulative variance of 99.9974%. Backpropagation based Deep Neural Network algorithm is implemented on the three principal components to build a PCAMixDNN classifier, which resulted in an accuracy of 91%. Finally, a comparison made on the performance of PCAMixDNN classifier with three existing classifiers, namely Decision Tree, Naïve Bayes, and Multilayer Perceptrons and it is found that our PCAMixDNN classifier has produced better accuracy in classifying new test data.

Future work will focus on improving the accuracy of PCAMixDNN classifier by adopting different activation functions at different hidden layers, making use of natural evolution techniques such as Ant Colony Optimization [17], Particle Swarm Optimization [18] for DNN parameter optimization. Also, we analyze the performance of our PCAMixDNN on other Data Mining applications.

REFERENCES

[1] Dunhamm M. H., “Data Mining: Introductory and Advanced Topics”, Pearson Education, India, 2006.

[2] Han J. and Kamber M., “Data Mining Concepts and Techniques”, Morgan Kauffmann Publishers, India, 2006.

[3] Lin C. and Yan F., “The Study on Classification and Prediction for Data Mining”, Seventh Int’l Conf. on Measuring Technology and Mechatronics Automation, 2015.

[4] Zhang P.G., “Neural networks for classification: a survey”, IEEE Trans. Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 30, Issue 4, Nov 2000.

[5] Gao T., Li X., Chai Y. and Tang Y., "Deep Learning with Stock Indicators and two-dimensional Principal Component Analysis for closing price prediction system", Seventh IEEE Int'l Conf. Software Engineering and Service Science, IEEE, Aug 2016.

[6] Shi H. and Liu X., "Application on Stock Price Prediction of Elman neural networks based on Principal Component Analysis Method", 11th Int'l Conf. Wavelet Actiev Media Technology and Information Processing, IEEE, Dec 2014.

[7] Ming C.T.J., Noor M.N, Rijal M.O., Kassim M.R and Yunus A., "Lung Disease Classification Using Different Deep Learning Architectures and Principal Component Analysis", 2nd Int'l Conf. BioSignal Analysis, Processing and Systems, IEEE, July 2018.

[8] Feng W., Zhao Y. and Deng J., "Application of SVM Based on Principal Component Analysis to Credit Risk Assessment in Commercial Banks", WRI Global Congress on Intelligent Systems, IEEE, May 2009.

[9] Min Z., "Credit Risk Assessment Based on Fuzzy SVM and Principal Component Analysis", Int'l Conf. Web Information Systems and Mining, IEEE, Nov 2009.

[10] Ioniţă I. and Şchiopu D., "Using Principal Component in Loan Granting", Seria Matematică – Informatică – Fizică, Vol. LXII, pp. 88-96, 2010.

[11] https://pbpython.com/categorical-encoding.html

[12] https://en.wikipedia.org/wiki/Principal_component_analysis

[13] https://en.wikipedia.org/wiki/Backpropagation

[14] https://archive.ics.uci.edu/ml/datasets/bank+marketing

[15] https://www.jetbrains.com/pycharm/

[16] Naraei P., Abhari A. and Sadeghian A., "Application of Multilayer Perceptron Neural Networks and Support Vector Machines in Classification of Healthcare Data", Future Technologies Conference, IEEE, Dec. 2016.

[17] Abd-Alsabour N., "A Review on Evolutionary Feature Selection", European Modelling Symposium, IEEE, Oct 2014.

[18] Holden N. and Freitas A.A., "A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data", Proc. Swarm Intelligence Symposium, IEEE, June 2005.

[19] Fernandes M., "Data Mining: A Comparative Study of its Various Techniques and its Process", International Journal of Scientific Research in Computer Science and Engineering, Vol. 5, Issue 1, Feb. 2017, pp.19-23.

[20] Ghuse N., Pawar P. and Potgantwar A., "An Improved Approach For Fraud Detection In Health Insurance Using Data Mining Techniques", IJSRNSC, Vol. 5, Issue 5, June 2017.

Authors Profile

Mr. Chittem Leela Krishna received his B.Tech degree from SVU Tirupati in the department of Computer Science and Engineering in 2012 and M.Tech degree from JNTU Anantapur in the department of Computer Science and Engineering in 2014. He is currently pursuing his Ph.D as a Full-time Research Scholar in the department of Computer Science and Engineering at S.V. University, Tirupati. His areas of Research include Data Mining, Web Intelligence and Machine Learning.

Prof. Poli Venkata Subba Reddy is currently working as Professor and Chairman, Computer Science and Engineering, College of Engineering, Sri Venkateswara University, Tirupati, India. He did B.S, SVU, 1984., M.S., SVU, 1986), M.Phill (DBMS) SVU 1988., Ph.D(AI) SVU 1992) in S V University, PDF (fuzzy algorithms) IISc/JNCSR, 1995 and PGDCM&P, CSI 1985. His research interests are Artificial Intelligence, fuzzy Systems and Database systems. He is member in IEEE. He published more than 50 publications in reputed International Journals and Conferences. He visited Taiwan. Hong Kong, Thailand, Malaysia, South Korea and Dubai to present the research papers.