Classification by Clustering (CbC): An Approach of Classifying Big Data based on Similarities

Sakib Shahriar Khan (011 143 138)
Shakim Ahamed (011 143 066)
Miftahul Jannat (011 151 136)
Irin Monwar (011 131 131
Department of Computer Science and Engineering
United International University

A thesis submitted for the degree of
BSc in Computer Science & Engineering

January 2019

Abstract

Data classification in supervised learning is the process of classifying data for data mining task that helps to analyse data for decision making. The objective of a classification model is to correctly predict the categorical class labels of known/ unknown instances. In machine learning for data mining applications, the classification models are trained based on labelled training datasets. In this paper, we have investigated if we can build a classification model based on the similarities of the instances instead of class labels of instances. Data labelling is always very costly and time consuming process, and it's become very difficult task if the data is big data. The proposed approach clusters the big data and builds the classifier based on the clusters without considering the class labels, which basically improve the performance of the classifier. However, we can relate the clusters with class labels. We have collected 10 big data from the UC Irvine machine learning repository for experimental analysis and applied three popular decision tree induction algorithms: ID3 (Iterative Dichotomiser 3), C4.5 (extension of ID3 algorithm), and CART (Classification & Regression Tree) for classifier construction.

We have dedicated this work to our respectful parents.

Published Papers

Work relating to the research presented in this thesis has been published/ submitted by the author in the following peer-reviewed journals and conferences:

1. Sakib Shahriar Khan, Shakim Ahamed, Miftahul Jannat, Swakkhar Shatabda, and Dewan Md. Farid, Classification by Clustering (CbC): An Approach of Classifying Big Data based on Similarities, 2018 International Joint Conference on Computational Intelligence (IJCCI 2018), December 14-15, 2018, Dhaka, Bangladesh, pp. 1-13.

Acknowledgements

We take this moment solely, to thank Almighty Allah who kept His constant mercy and blessings on us while taking on this thesis, without which undertaking all the extra mileage of working hours and taking up all the tricks and turns would not have been possible. Following I would like to express our earnest gratitude to our supervisor, Dr. Dewan Md. Farid, not only for his continuous support, motivation, immense knowledge, but also for the hard assessments which helped us to widen my research from various perspectives. Nonetheless, I would like to give an extra credit to our family, our parents, our siblings, who have been parallel to everyone mentioned above, without their support and guidance in our life and these past four years, conducting our thesis would have been anything but easy.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In current digital era, due to the rise of digital technology uses the volume of digital data is rapidly increasing day by day. It has opened a new research area in the field of data science and attracts computational intelligence researchers to work on big data [6]. Big data generally refer to 3Vs: volume, variety, and velocity [14]. Volume is the quantity of the data, variety is the different types of attributes in the data, and velocity is the continuous incoming data in and out. There are few other Vs can be added for big data analytics such as vision, verification, and validation [5]. Vision is the planning/ purpose of the data that would be achieved. The verification ensures that the data conforms to a set of specifications and validation checks whether the big data generation is fulfilled or not. In case of big data, there could be several different sources (e.g. customer analysis, medical research, government research, security and law enforcement, and financial trading etc.) [15]. Mining big data employing data mining algorithms is a challenging and difficult task. Knowledge extraction from big data is the process of discovering patterns or unknown structures in big data [20]. Machine learning for mining big data can be bifurcated into descriptive and predictive tasks [9, 10]. Descriptive mining is the task of analysing, summarising, and finding historical patterns in big data. On the contrary, predictive mining is the process of decision making or prediction of unseen data by building a learning model from the big data [12]. In general, predictive mining is categorised into supervised and unsupervised learning. Supervised learning focuses on classification and regression techniques. Whereas, unsupervised learning looks for structures in the data. In classification/ regression, a predictive model is created from the labelled training data having the output variables or target predictions [8, 19].

Labelling training data is a costly and time consuming process. In many real-world applications, human expert manually label the training instances like Bioinformatics, genomic data processing, intrusion classification etc. [6]. Big data contain thousands/ millions of the instances in the train set data and labelling these large numbers of training instances is really very difficult task. In this paper, we have proposed an approach of classifying unlabelled big data employing cluster analysis. The main objective of this study is the classification of big data by clustering (CbC). The idea is to build a classifier based on the similarities of the instances instead of class labels of the instances for classifying big data. The proposed method clusters the big data into several groups/ clusters and builds a classification model based on the clusters. Clustering is a process of grouping a set of instances into a set of sub-classes that called clusters, it's also known as learning by observation, automatic classification, data segmentation, unsupervised learning, etc. Similarities and dissimilarities of instances can be determined by the feature values. Clustering groups the big data into several clusters so that instances in a same cluster have high similarity to each other and instances in different clusters are high dissimilar [7]. To design and build classifiers, we have used several decision tree learning algorithms: ID3, C4.5, and CART, as decision tree induction is one of the most popular machine learning algorithms for data mining task [11]. We have collected 10 big datasets from the UC Irvine machine learning repository for experimental analysis and validated the proposed method. The experimental results show that the proposed method scales up the classification accuracy for most of the datasets.

1.1 Motivation

Mining big data is always a very difficult task in data science, it is very completed to extract knowledge from a huge dataset and when to comes to mining unlabelled data it becomes too difficult, costly and time consuming. Many real-world datasets are unlabelled and they are very big in size. Most often big datasets are not fit in computer memory. So we realized that if we can make a method which can handle huge amount of data at a time and works with unlabelled data or has no dependency on class labels, it would be fascinating.

1.2 Objectives of the Thesis

Main object the thesis is to build an efficient, scalable and adaptive approach for mining big data. We tried to make mine bigdata efficiently with traditional classifiers which can take new instances and update the decision tree to reflect the changes in data without constructing a new decision tree from the scratch. It can handle big data and automatically label the instances by clustering.

1.3 Organization of the Thesis

The thesis is organized as follows:

Chapter 2 Discusses about Machine Learning Algorithms.

Chapter 3 Presents the proposed method.

Chapter 4 Discusses the results and experimental analysis.

Chapter 5 Presents the conclusions, summaries the thesis contributions, and discusses the future works.

Chapter 2

Machine Learning Algorithms

In this chapter, we will discuss about some machine learning learning algorithms we have used in our experiment. All of them are very popular in the field of data mining and machine learning. Both unsupervised and supervised learning will be described in this chapter.

2.1 Decision Tree Induction

Decision tree (DT) is one of the most popular machine learning algorithms for data mining task [11]. DT can be used for solving many real-life classification and regression problems in supervised learning. It presents a set of rules; each path from root to leaf node represents a classification rule. DT is basically a top-down recursive divide and conquer approach. DT starts with a root node chosen by the highest entropy of feature and follows the tree down the branches using gain until a leaf node represents a single class. Information gain, which is also known as mutual information that is a ratio of information gain to reduce the bias towards multi-valued features by taking the number and size of branches into account when choosing a node of the tree [18]. The following equations Eq. 2.1 to Eq. 2.3 exhibit the gain calculation.

$$Entropy(X) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{2.1}$$

$$Entropy_A(X) = \sum_{j=1}^{n} \frac{|X_j|}{|X|} \times Entropy(X_j) \tag{2.2}$$

$$Gain(A) = Entropy(X) - Entropy_A(X) \tag{2.3}$$

The Gain Ratio is applied in C4.5 algorithm [17], which is an extension of the entropy based method that is the successor of ID3 (Iterative Dichotomiser 3) algorithm [18]. Gain Ratio uses "split information" a kind of normalisation of entropy estimation. The feature with maximum Gain Ratio is selected for further splitting in the tree. Eq. 2.4 and Eq. 2.5 present the Gain Ratio calculation.

$$SplitInfo_A(X) = -\sum_{j=1}^{n} \frac{|X_j|}{|X|} \times log_2\left(\frac{|X_j|}{|X|}\right) \tag{2.4}$$

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)} \tag{2.5}$$

Another DT induction algorithm is called Classification and Regression Tree (CART) [2], which uses Gini Index in Eq. 2.6 and Eq. 2.7. The attribute, A_j that maximises the reduction in impurity is selected as the splitting feature.

$$Gini(X) = 1 - \sum_{j=1}^{n} p_j^2 \tag{2.6}$$

The splitting of X into subsets X_1 and X_2 is dependent on the following equation:

$$Gini_A(X) = 1 - \frac{\mid X_1 \mid \times Gini(X_1)}{\mid X \mid} + \frac{\mid X_2 \mid \times Gini(X_2)}{\mid X \mid} \tag{2.7}$$

2.2 Clustering Analysis

Clustering is the process of grouping or partitioning unlabelled data/ objects [1]. Unknown hidden patterns/ structure can be found in the data by applying clustering analysis. Clustering algorithms try to group the data into many clusters so that instances/ data points in a same cluster have very high similarity to each other and instances in different clusters are highly dissimilar. Another principle of cluster analysis is that every cluster should have minimum one instance and one instance, should belong to only one cluster [3]. Let, X be the set of unlabelled instances.

$$X = \{x_1, x_2, x_3, ..., x_n\} \tag{2.8}$$

After partitioning X into k clusters:

$$C_p \neq \emptyset, p = 1, ..., k \tag{2.9}$$

$$X = C_1 \cup C_2 \cup C_3 \cup .... \cup C_k \tag{2.10}$$

$$C_p \cap C_q = \emptyset, p \neq q, p, q = 1, ..., k \tag{2.11}$$

Similarities and dissimilarities of instances can be determined by the feature values in the dataset. Clustering refers to the automatic classification, which is also known as data segmentation, unsupervised learning, learning by observation etc. Clustering methods are divided into 4 categories: (1) partitioning method, (2) hierarchical method, (3) density-based method, and (4) grid-based method [7, 12]. Partitioning based clustering method clusters N instances into K number of clusters of convex shape ($N \geq K$). It uses traditional distance measures (e.g. Euclidean distances, Manhattan distance) to cluster the data points. It uses mean or medoid to find cluster center and an iterative relocating approach to improve the cluster validation (e.g. K-means clustering) [13].

2.3 Summary

This chapter discusses about various machine learning algorithms like Decision Tree and Clustering. We have discussed about three variant of decision tree and which are ID3, C4.5 and CART. We have also discussed about different clustering approaches and how they work.

Chapter 3

Method for Classifying Big Data

In this chapter we have discussed about how we have done this work. We tried to remove the dependency from class labels as some dataset may not have class labels and finding them is too difficult. And also we have worked for scalibilty of the method, we tried to make an approach which does not require change when new instances come. The main idea of classifying big data without considering the class labels with an scalable approach is shown in this chapter.

3.1 Method Description

Classifying big data is a challenging and difficult task, as there are thousands/ millions of instances in the data and we cannot analysis and store the full data at a time. The most common problem to deal with big data is to store the total data into the computer memory as big data are so big in volume. Therefore, parallel and distributed computing becomes very useful for handling big data. In general, big data is divided into several small sub-sets that we can deal with and each of which fits into the computer memory. Most real-world datasets are unlabelled and labelling data is very costly and time consuming. Most of the time human experts label the unlabelled the data instances. In this paper, we have proposed an approach for classifying big data employing classification by clustering (CbC) method that does not consider the class labels of data instances for building classifiers. Initially, the big data is divided into several small sub-datasets. Then clustering technique is applied to each sub-dataset to cluster the data. The proposed method used similarity-based clustering algorithm, which is a robust method for

clustering instances based on the similarities. The similarity-based clustering method automatically generates the cluster numbers and form different volumes of clusters. After that, several decision trees are built from sub-datasets. Finally, all the decision trees are examined and analysed to form a single decision tree that turns out to be very close to the tree that would have been generated if the big data had fit in memory. To select an informative attribute/ feature for constructing a decision tree, we can use any attribute selection techniques, e.g. ID3, C4.5, and CART. Algorithm 1, outlines the proposed algorithm for classifying unlabelled big data using similarity-based clustering with decision tree.

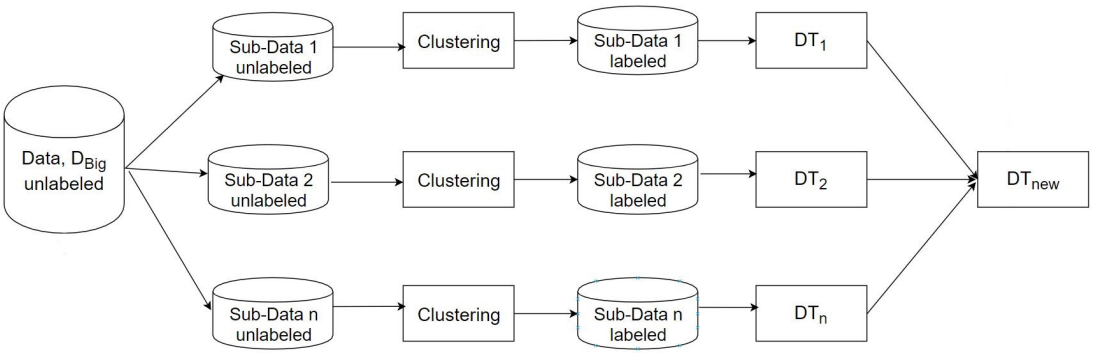

Figure 3.1: Block diagram of proposed idea

The proposed algorithm is an adaptive and scalable approach. It can take new instances and update the decision tree to reflect the changes in data without constructing a new decision tree from the scratch. The main idea of this paper is to build a scalable and adaptive classification model for mining big data based on the similarities of the instances instead of class labels of instances. We have divided the big data, D_{Big} into several small sub-datasets, $\{D_1, D_2, \cdots, D_n\}$ and applied clustering technique to cluster each of sub-data, D_i based on similarity of instances. The similarities are determined based on the features of the data, so that similar instances are grouped together. Then we have built decision tree from each sub-dataset using cluster analysis without considering class labels of the data. The *DTBuild()* is a recursive function, which builds the decision tree from a given training data. It can be used, information gain, gain ratio, or gini index for selecting the best/ root attribute/ feature. Resulting we have n number

of decision trees, DT_s. To form a single classification model we merge all the decision trees and build a new decision tree, DT_{New}. In this study, we have considered decision tree learning algorithms for classifier construction, because decision tree algorithms are very well known and commonly using in many real-world data mining applications. It's easy to build and easy to understand. Our proposed algorithm is given in Algorithm 1

3.2 Summary

In this chapter, we have shown the approach of classifying data by replacing class labels with the cluster id's of the instances, we also divided the dataset into small chunk of data for removing the memory allocation problem. We also made this approach scalable, new instances will automatically adjusted with the decision tree without building it again.

Algorithm 1 Classifying Big Data by CbC

Input: Big Data, D_{Big};

Output: A classification model, M;

Method:

divide D_{Big} into several sub-datasets $\{D_1, D_2, \cdots, D_n\}$;
for each $D_i = \{x_1, x_2, \cdots, x_N\}$ **do**
  // find clusters, $C = \{C_1, C_2, \cdots, C_k\}$;
  $C = \emptyset$, $k = 1$, & $C_k = \{x_1\}$;
  $C = C \cup C_k$;
  for $i = 2$ to N **do**
    for $l = 1$ to k **do**
      find maximum similarity, $sim(x_i, x_l)$ with lth cluster center $x_l \in C_l$;
    end for
    if $sim(x_i, x_l) \geq threshold$ **then**
      $C_l = C_l \cup x_i$;
    else
      $k = k + 1$, & $C_k = \{x_i\}$;
      $C = C \cup C_k$;
    end if
  end for
  // create a decision tree: DTBuild(D_i);
  DT_i = find the root node with best splitting attribute, $A_j \in D_i$, & add arc to root node for each split predicate and label with cluster, C_k;
  for each arc **do**
    create D_j by applying splitting predicate of D_i;
    if stopping point reached for this path **then**
      DT' = create leaf node by label C_k;
    else
      DT' = DTBuild(D_j);
    end if
    DT_i = add DT' to arc;
  end for
end for
construct a new tree, DT_{New} using decision trees, $DT_s = \{DT_1, DT_2, \cdots, DT_n\}$;
$M = DT_{New}$;

Chapter 4

Experimental Analysis

This section presents the datasets, experimental setup and results.

4.1 Dataset Description

We have selected the following 10 real benchmark datasets from the UCI machine learning repository [4]. Table 4.1 presents the details of the datasets.

1. Covertype Data Set (Covertype)
2. APS Failure at Scania Trucks Data Set (APS)
3. Gas Sensor Array Drift Dataset Data Set (Gas Sensor)
4. Character Font Images Data Set (Character Font)
5. Semeion Handwritten Digit Data Set (Semeion Digit)
6. Adult Data Set (Adult)
7. KDD Cup 1999 Data Data Set (KDD Cup)
8. ISOLET Data Set (ISOLET)
9. UJIIndoorLoc Data Set (UJIIndoorLoc)
10. Dota2 Games Results Data Set (Dota2 Games)

Table 4.1: Datasets description.

No.	Datasets	No. of Features	Types of Features	Instances	Classes
1	Covertype	54	Categorical, Integer	581012	7
2	APS Failure	171	Integer, Real	60000	2
3	Gas Sensor	128	Real	13910	6
4	Character Font	411	Integer, Real	745000	153
5	Semeion Digit	256	Integer	1593	10
6	Adult	41	Categorical, Integer	48842	2
7	KDD Cup	42	Categorical, Integer	4000000	23
8	ISOLET	617	Real	7737	26
9	UJIIndoorLoc	529	Integer, Real	21048	5
10	Dota2 Games	117	Integer, Real	102944	2

4.2 Experimental Setup

We have used scikit-learn machine learning library in python for experimental setup and design [16]. We have used Spyder 3.2.6 for python coding. Spyder is a part of Anaconda distribution, which is cross-platform and includes most of the python packages. Scikit-learn library includes a range of machine learning algorithms that contains pre-processing, feature selection, dimensionality reduction, and model selection tools. In our experiments, ID3 and CART algorithms are directly used from scikit-learn library. We have implemented C4.5 in python. We have used the classification accuracy, precision, recall, F-score, and 10-fold CV (cross validation) to evaluate our proposed method that are shown in Eqs. 4.1 to 4.4 where TP, TN, FP and FN are true positive, true negative, false positive, and false negative respectively.

$$accuracy = \frac{\sum_{i=1}^{|X|} assess(x_i)}{|X|}, x_i \in X \tag{4.1}$$

$$precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$recall = \frac{TP}{TP + FN} \tag{4.3}$$

$$F - score = \frac{2 \times precision \times recall}{precision + recall} \tag{4.4}$$

4.3 Result

At first, we have evaluated the performance of existing decision tree algorithms such as ID3, C4.5, and CART on benchmark datasets from Table 4.1. We have measured the accuracy, precision, recall, F-score with 10-fold cross validation to test the performance of these classifiers. We have considered a weighted average of precision, recall, and F-score. The results are summarised in Table 4.2, Table 4.3, and Table 4.4 for ID3, C4.5, and CART respectively. The ID3 and C4.5 algorithms work well with more than 98% accuracy on APS Failure, Gas Sensor, Character Font, KDD Cup, and UJIIndoorLoc datasets. Contrarily, CART performs better on APS Failure, Gas Sensor, Character Font, and KDD Cup datasets. As the decision tree is a strong classifier the overall performance of ID3, C4.5, and CART algorithms are good on benchmark datasets except for the Dota2 Games dataset. Then, we have tested the performance of the proposed classification by clustering (CbC) approach with ID3, C4.5, and CART on benchmark datasets from the Table 4.1. The results of CbC are tabulated in Table 4.5, Table 4.6 and Table 4.7 respectively. The proposed approach improves the classification results of Dota2 Games dataset from 52% to greater than 98%, Adult dataset from 81% to 99%, and Semeion Digit dataset from 74% to 78%. It classifies the Covertype, KDD Cup, and UJIIndoorLoc datasets with 99% accuracy and APS Failure dataset with 98% accuracy. Figs. 4.1 to 4.3 show the comparison of classification accuracies of decision tree algorithms (ID3, C4.5, and CART) with proposed CbC approach. Fig. 4.4 shows the comparison of accuracies of classifiers.

Table 4.2: Classification results of ID3 algorithm using 10-fold CV on benchmark datasets.

No.	Datasets	Accuracy (%)	Precision (weighted average)	Recall (weighted average)	F-Score (weighted average)
1	Covertype	93.13	0.927	0.927	0.928
2	APS Failure	98.66	0.986	0.986	0.986
3	Gas Sensor	98.58	0.984	0.984	0.984
4	Character Font	99.99	0.999	0.999	0.999
5	Semeion Digit	74.81	0.765	0.748	0.748
6	Adult	81.40	0.815	0.814	0.814
7	KDD Cup	99.90	0.998	0.998	0.998
8	ISOLET	83.35	0.839	0.834	0.834
9	UJIIndoorLoc	98.18	0.982	0.982	0.982
10	Dota2 Games	52.40	0.524	0.524	0.524

Table 4.3: Classification results of C4.5 algorithm using 10-fold CV on benchmark datasets.

No.	Datasets	Accuracy (%)	Precision (weighted average)	Recall (weighted average)	F-Score (weighted average)
1	Covertype	91.70	0.915	0.915	0.915
2	APS Failure	98.62	0.986	0.985	0.986
3	Gas Sensor	98.63	0.986	0.986	0.986
4	Character Font	99.70	0.997	0.997	0.997
5	Semeion Digit	74.25	0.753	0.747	0.748
6	Adult	81.53	0.816	0.812	0.813
7	KDD Cup	99.95	0.999	0.999	0.999
8	ISOLET	83.13	0.838	0.831	0.832
9	UJIIndoorLoc	98.33	0.983	0.983	0.983
10	Dota2 Games	52.34	0.523	0.523	0.523

Table 4.4: Classification results of CART algorithm using 10-fold CV on benchmark datasets.

No.	Datasets	Accuracy (%)	Precision (weighted average)	Recall (weighted average)	F-Score (weighted average)
1	Covertype	92.30	0.925	0.922	0.925
2	APS Failure	98.41	0.983	0.984	0.983
3	Gas Sensor	98.54	0.985	0.985	0.925
4	Character Font	99.90	0.998	0.998	0.998
5	Semeion Digit	74.82	0.759	0.748	0.747
6	Adult	74.82	0.812	0.811	0.812
7	KDD Cup	99.93	0.999	0.999	0.999
8	ISOLET	82.19	0.828	0.822	0.822
9	UJIIndoorLoc	97.84	0.979	0.978	0.978
10	Dota2 Games	52.10	0.521	0.521	0.521

4.4 Summary

In this chapter we have described the properties of the datasets used for the experiment, how we did the experiment and what tools are used. The results are elaborately shown through sufficient number of tables and figures.

Table 4.5: Classification results of CbC with ID3 algorithm using 10-fold CV on benchmark datasets.

No.	Datasets	Accuracy (%)	Precision (weighted average)	Recall (weighted average)	F-Score (weighted average)
1	Covertype	99.53	0.994	0.993	0.994
2	APS Failure	98.65	0.986	0.986	0.986
3	Gas Sensor	97.90	0.978	0.978	0.978
4	Character Font	81.79	0.818	0.887	0.887
5	Semeion Digit	77.58	0.785	0.775	0.775
6	Adult	99.99	0.999	0.998	0.998
7	KDD Cup	99.92	0.999	0.999	0.998
8	ISOLET	84.65	0.852	0.847	0.846
9	UJIIndoorLoc	99.99	0.999	0.999	0.999
10	Dota2 Games	98.55	0.984	0.985	0.985

Table 4.6: Classification results of CbC with C4.5 algorithm using 10-fold CV on benchmark datasets.

No.	Datasets	Accuracy (%)	Precision (weighted average)	Recall (weighted average)	F-Score (weighted average)
1	Covertype	99.48	0.994	0.994	0.994
2	APS Failure	98.65	0.986	0.986	0.986
3	Gas Sensor	97.70	0.976	0.976	0.976
4	Character Font	80.75	0.807	0.807	0.807
5	Semeion Digit	78.13	0.799	0.788	0.789
6	Adult	99.40	0.994	0.998	0.998
7	KDD Cup	99.99	0.999	0.999	0.999
8	ISOLET	84.75	0.853	0.857	0.857
9	UJIIndoorLoc	99.99	0.999	0.999	0.999
10	Dota2 Games	98.86	0.988	0.988	0.988

Table 4.7: Classification results of CbC with CART algorithm using 10-fold CV on benchmark datasets.

No.	Datasets	Accuracy (%)	Precision (weighted average)	Recall (weighted average)	F-Score (weighted average)
1	Covertype	99.50	0.995	0.994	0.994
2	APS Failure	99.90	0.999	0.999	0.999
3	Gas Sensor	98.62	0.986	0.986	0.986
4	Character Font	80.25	0.803	0.802	0.802
5	Semeion Digit	78.65	0.798	0.786	0.786
6	Adult	99.99	0.999	0.999	0.999
7	KDD Cup	99.99	0.999	0.999	0.999
8	ISOLET	83.04	0.835	0.830	0.832
9	UJIIndoorLoc	99.99	0.999	0.999	0.999
10	Dota2 Games	99.52	0.996	0.995	0.995

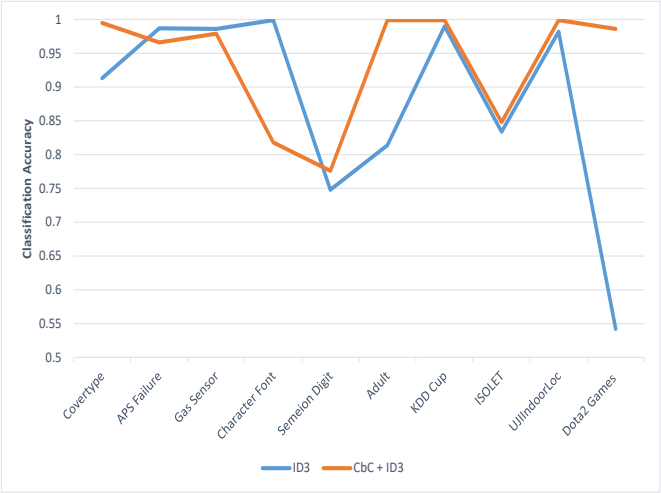

Figure 4.1: Accuracy of ID3 and CbC with ID3 algorithms on benchmark datasets.

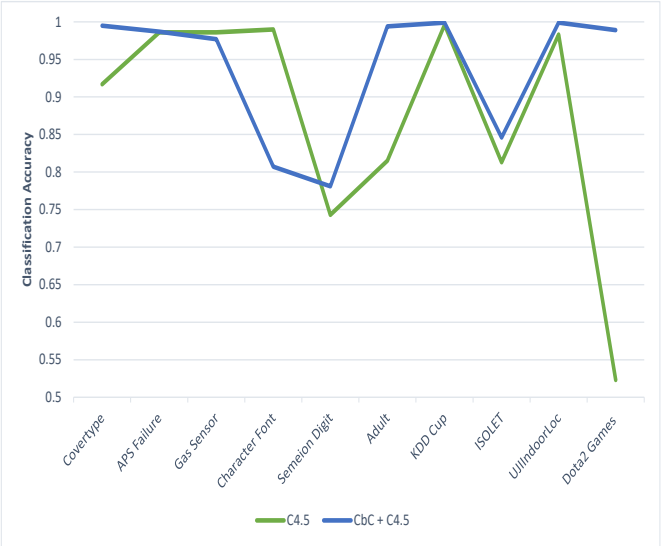

Figure 4.2: Accuracy of C4.5 and CbC with C4.5 algorithms on benchmark datasets.

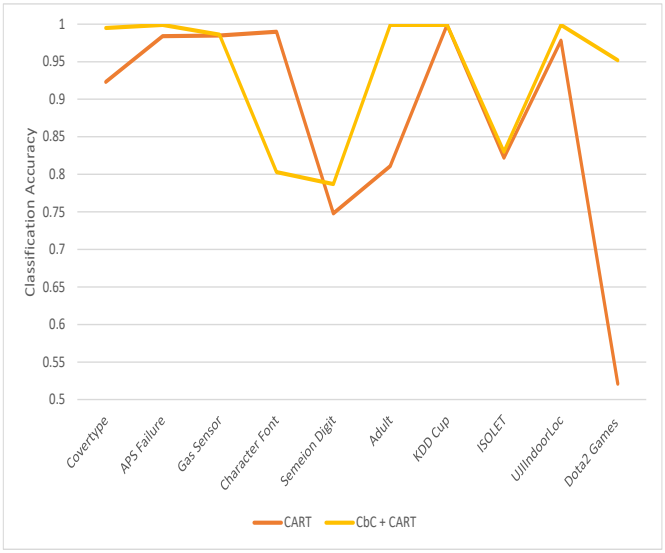

Figure 4.3: Accuracy of CART and CbC with CART algorithms on benchmark datasets.

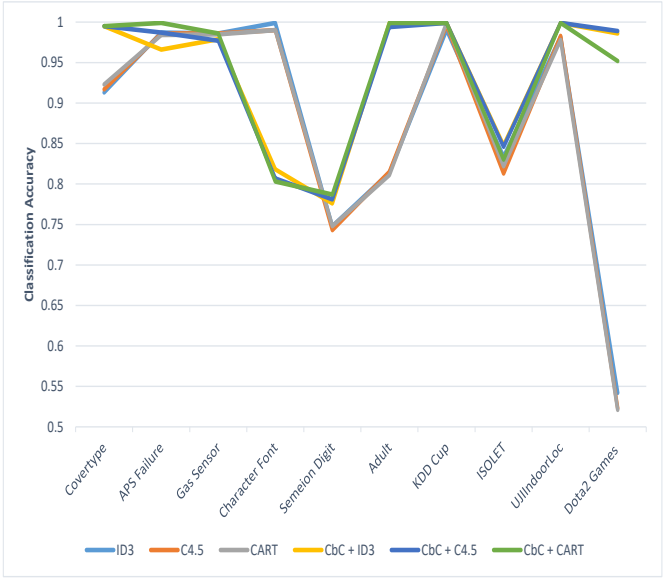

Figure 4.4: Classification accuracy of the different classifiers.

Chapter 5

Conclusions, and Future Work

5.1 Conclusions

In this paper, we have introduced a method for classifying unlabelled big data based on classification by cluster (CbC) technique. We have used similarity-based clustering and decision tree algorithms in the proposed method. Both decision tree and similarity-based clustering methods are very popular data mining techniques that are used in many real-world data mining applications. The objective of this paper is to build a classification model based on the similarity of instances instead of the class labels of the instances. The idea is to avoid contradictory and redundant class labels in the training data while building a classifier. As noisy class labels are very common in real-life datasets. The proposed method builds a classifier based on the data, not on the class labels. The experimental results proved that the proposed method of similarities of instances for classification improves the classification results. We have taken 10 benchmark datasets from the UCI machine learning repository with large number of features and instances for experimental analysis that we can consider the datasets as big data. We have used scikit-learn machine learning library in python for experimental setup, which is a simple and efficient tool for data mining and data analysis. The proposed algorithm overall improves the classification results in comparing with traditional decision tree algorithms (ID3, C4.5, and CART) for all datasets except Character Font dataset. Specially, it improves the classification accuracy of the Dota2 Games dataset from 52% to 98% and Adult dataset from 81% to 99%.

5.2 Future Work

In the future, we will apply adaptive ensemble classifiers with the proposed method for classifying big data under dynamic feature set and also for novel class classification in concept drifting.

Bibliography

[1] Aggarwal, C.C., Reddy, C.K. (eds.): Data Clustering: Algorithms and Applications. Chapman and Hall/CRC Data Mining and Knowledge Discovery Series. Chapman and Hall/CRC (2013) 5

[2] Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Chapman and Hall/CRC (1984) 5

[3] Chen, X., and Xiaofei Xu, Y.Y., Huang, J.Z.: A feature group weighting method for subspace clustering of high-dimensional data. Pattern Recognition **45**(1), 434–446 (2012) 5

[4] Dheeru, D., Taniskidou, E.K.: UCI machine learning repository (2017). URL `http://archive.ics.uci.edu/ml` 11

[5] Fan, W., Bifet, A.: Mining big data: Current status, and forecast to the future. ACM SIGKDD Explorations Newsletter **14**(2), 1–5 (2013) 1

[6] Farid, D.M., Al-Mamun, M.A., Manderick, B., Nowe, A.: An adaptive rule-based classifier for mining big biological data. Expert Systems with Applications **64**, 305–316 (2016) 1, 2

[7] Farid, D.M., Nowé, A., Manderick, B.: A feature grouping method for ensemble clustering of high-dimensional genomic big data. In: Future Technologies Conference, pp. 260–268. San Francisco, United States (2016) 2, 6

[8] Farid, D.M., Rahman, C.M.: Assigning weights to training instances increases classification accuracy. International Journal of Data Mining & Knowledge Management Process **3**(1), 129–135 (2013) 1

[9] Farid, D.M., Rahman, C.M.: Mining complex data streams: discretization, attribute selection and classification. Journal of Advances in Information Technology **4**(3), 129–135 (2013) 1

[10] Farid, D.M., Zhang, L., Hossain, A., Rahman, C.M., Strachan, R., Sexton, G., Dahal, K.: An adaptive ensemble classifier for mining concept drifting data streams. Expert Systems with Applications **40**(15), 5895–5906 (2013) 1

[11] Farid, D.M., Zhang, L., Rahman, C.M., Hossain, M., Strachan, R.: Hybrid decision tree and naïve bayes classifiers for multi-class classification tasks. Expert Systems with Applications **41**(4), 1937–1946 (2014) 2, 4

[12] Han, J., Kamber, M., Pei, J.: Data Mining Concepts and Techniques, third edn. Morgan Kaufmann (2011) 1, 6

[13] Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recognition Letters **31**(8), 651–666 (2010) 6

[14] L'heureux, A., Grolinger, K., Elyamany, H.F., Capretz, M.A.M.: Machine learning with big data: Challenges and approaches. IEEE Access **5**, 7776–7797 (2017) 1

[15] Özköse, H., Arı, E.S., Gencer, C.: Yesterday, today and tomorrow of big data. Procedia-Social and Behavioral Sciences **195**, 1042–1050 (2015) 1

[16] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Édouard Duchesnay: Scikit-learn: Machine learning in python. The Journal of Machine Learning Research **12**, 2825–2830 (2011). URL `http://dl.acm.org/citation.cfm?id=1953048.2078195` 12

[17] Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993) 5

[18] Quinlan, J.R.: Induction of decision tree. Machine Learning **1**(1), 81–106 (1986) 4, 5

[19] Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, third edn. Morgan Kaufmann (2011) 1

[20] Wu, X., Zhu, X., Wu, G.Q., Ding, W.: Data mining with big data. IEEE Transactions on Knowledge & Data Engineering **26**(1), 97–107 (2014) 1