International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-3, Issue-1 E-ISSN: 2347-2693

An Analysis of Classification and Clustering Techniques used in Data Mining

Vishakha D. Charhate[1], Ms Poonam A. Manjare[2]
Dept. of CSE, Sgbau University, Amravati, India
www.ijcaonline.org

Received: Dec /21/2014 Revised: Jan/4/2015 Accepted: Jan/17/2015 Published: Jan/31/2015

Abstract— We live in a time where the need for data mining is prevalent for extracting knowledge and understanding patterns, given the vast amount of data being generated. Clustering is one of the many data mining functionalities which divide data into groups containing similar data objects Classification is a technique used for discovering classes of unknown data. Before applying any mining technique, irrelevant attributes needs to be filtered. Filtering is done using different feature selection techniques like wrapper, filter and embedded technique. This paper is an introductory paper on different techniques used for classification and clustering.

Keywords- Data Mining, Clustering Techniques, Classification Techniques

I. INTRODUCTION

Over the past several years,the field of data mining has seen an explosion of interest from both academia and industry. Data mining is an interdisciplinary field and draws heavily on both statistics and machine learning. In these two areas, such problems as learning how to classify data and finding natural clusters of data have been studied extensively for decades. Furthermore, the earliest mathematical programming formulations of both data classification [1] and data clustering [2] date back almost 40 years. Building on this early work, the recent growing interest in data mining has been paralleled by a similar growth of research in optimization for data mining [3,4], and the operations research community has the potential to continue to contribute significantly to this field.

Classification: Classification Data mining algorithms can follow three different learning approaches: supervised, unsupervised, or semi-supervised. In supervised learning, the algorithm works with a set of examples whose labels are known. The labels can be nominal values in the case of the classification task, or numerical values in the case of the regression task. In unsupervised learning, in contrast, the labels of the examples in the dataset are unknown, and the algorithm typically aims at grouping examples according to the similarity of their attribute values, characterizing a clustering task. Finally, semi-supervised learning is usually used when a small subset of labelled examples is available, together with a large number of unlabeled examples. The classification task can be seen as a supervised technique where each instance belongs to a class, which is indicated by the value of a special goal attribute or simply the class attribute. The goal attribute can take on categorical values,

Each of them corresponding to a class. Each example consists of two parts, namely a set of predictor attribute values and a goal attribute value. The former are used to predict the value of the latter. The predictor attributes should be relevant for predicting the class of an instance. In the classification task the set of examples being mined is divided into two mutually exclusive and exhaustive sets, called the training set and the test set. The classification process is correspondingly divided into two phases: training, when a classification model is built from the training set, and testing, when the model is evaluated on the test set. In the training phase the algorithm has access to the values of both predictor attributes and the goal attribute for all examples of the training set, and it uses that information to build a classification model. This model represents classification knowledge – essentially, a relationship between predictor attribute values and classes – that allows the prediction of the class of an example given its predictor attribute values. For testing, the test set the class values of the examples is not shown. In the testing phase, only after a prediction is made is the algorithm allowed to see the actual class of the just-classified example. One of the major goals of a classification algorithm is to maximize the predictive accuracy obtained by the classification model when classifying examples in the test set unseen during training. The knowledge discovered by a classification algorithm can be expressed in many different ways like rules, decision trees, Bayesian network etc. Various techniques used for classification are explained in the following section.

II. CLSSIFICATION TECHANIQUE

A. *Rule Based Classifiers*

Rule based classifiers deals with the discovery of high-level, easy-to-interpret classification rules of the form if-then. The rules are composed of two parts mainly rule antecedent and rule consequent. The rule antecedent, is the if part, specifies a set of conditions referring to predictor attribute values, and the rule consequent, the then part, specifies the class predicted by the rule for any example that satisfies the conditions in the rule antecedent. These rules can be generated using different classification algorithms, the most well known being the decision tree induction algorithms and sequential covering rule induction algorithms [1].

B. Bayesian Networks

A Bayesian network (BN) consists of a directed, acyclic graph and a probability distribution for each node in that graph given its immediate predecessors [2]. A Bayed Network Classifier is based on a Bayesian network which represents a joint probability distribution over a set of categorical attributes. It consists of two parts, the directed acyclic graph G consisting of nodes and arcs and the conditional probability tables. The nodes represent attributes whereas the arcs indicate direct dependencies. The density of the arcs in a BN is one measure of its complexity. Sparse BNs can represent simple probabilistic models (e.g., naive Bayed models and hidden Markov models), whereas dense BNs can capture highly complex models. Thus, BNs provide a flexible method for probabilistic modelling [3].

C. Decision Tree

A Decision Tree Classifier consists of a decision tree generated on the basis of instances. The decision tree has two types of nodes: a) the root and the internal nodes, b) the leaf nodes. The root and the internal nodes are associated with attributes, leaf nodes are associated with classes. Basically, each non-leaf node has an outgoing branch for each possible value of the attribute associated with the node. To determine the class for a new instance using a decision tree, beginning with the root, successive internal nodes are visited until a leaf node is reached. At the root node and at each internal node, a test is applied. The outcome of the test determines the branch traversed, and the next node visited. The class for the instance is the class of the final leaf node [4].

D. Nearest Neighbour

A Nearest Neighbour Classifier assumes all instances correspond to points in the n-dimensional space. During learning, all instances are remembered. When a new point is classified, the k- nearest points to the new point are found and are used with a weight for determining the class value of the new point. For the sake of increasing accuracy, greater weights are given to closer points [5].

E. Artificial Neural Network

An artificial neural network, often just called a neural network is a mathematical model or computational model based on biological neural networks, in other words, is an emulation of biological neural system. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase [6]. A Neural Network Classifier is based on neural networks consisting of interconnected neurons. From a simplified perspective, a neuron takes positive and negative stimuli (numerical values) from other neurons and when the weighted sum of the stimuli is greater than a given threshold value, it activates itself. The output value of the neuron is usually a non-linear transformation of the sum of stimuli. In more advanced models, the non-linear transformation is adapted by some continuous functions.

Rough Sets any set of all indiscernible (similar) objects is called an elementary set. Any union of some elementary sets is referred to as a crisp or precise set - otherwise the set is rough (imprecise, vague). Each rough set has boundary-line cases, i.e., objects which cannot be with certainty classified, by employing the available knowledge, as members of the set or its complement. Obviously rough sets, in contrast to precise sets, cannot be characterized in terms of information about their elements. With any rough set a pair of precise sets - called the lower and the upper approximation of the rough set is associated. The lower approximation consists of all objects which surely belong to the set and the upper approximation contains all objects which possible belong to the set. The difference between the upper and the lower approximation constitutes the boundary region of the rough set. Rough set approach to data analysis has many important advantages like provides efficient algorithms for finding hidden patterns in data, identifies relationships that would not be found using statistical methods, allows both qualitative and quantitative data, finds minimal sets of data (data reduction), evaluates significance of data, easy to understand [7].

F. Fuzzy Logic

Fuzzy logic is a multivalve logic different from "crisp logic", where binary sets have two valued logic. Fuzzy logic variables have truth value in the range between 0 and 1. Fuzzy logic is a superset of conventional Boolean logic that has been extended to handle the concept of partial truth. A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. Fuzzy Logic consists of Type 1 and Type 2 fuzzy logic. Type 1 fuzzy contains the constant values. A Type-2 Fuzzy Logic is an extension of Type 1 Fuzzy Logic in which the fuzzy sets comes from Existing Type 1 Fuzzy. A type-2 fuzzy set contains the grades of membership that are themselves fuzzy. A Type-2 membership grade can be any subset in the primary membership. For each primary membership there exists a secondary membership that defines the possibilities for the primary membership. Type-1 Fuzzy Logic is unable to handle rule uncertainties. Type-2 Fuzzy Logic can handle rule uncertainties effectively and efficiently [8]. Type 2 Fuzzy sets are again characterized by IF–THEN rules [9]. Type-2 Fuzzy is computationally intensive because type reduction is very intensive. Type-2 fuzzy is used for modeling uncertainty and imprecision in a better way. The type-2 fuzzy sets are called as “fuzzy fuzzy” sets where the fuzzy degree of membership is fuzzy itself that results from Type 1 Fuzzy [10].

Clustering: A "clustering" is essentially a set of such clusters, usually containing all objects in the data set.

Additionally, it may specify the relationship of the clusters to each other, for example a hierarchy of clusters embedded in each other. Data Clustering is one of the challenging mining techniques exploited in the knowledge discovery process. Clustering huge amounts of data is a difficult task since the goal is to find a suitable partition in a unsupervised way (i.e. without any prior knowledge) trying to maximize the similarity of objects belonging to the same cluster and minimizing the similarity among objects in different clusters. Many different clustering techniques have been defined in order to solve the problem from different perspective, i.e. partition based clustering, density based clustering, hierarchical methods and grid-based methods etc. In this paper we represent a survey of recent clustering approaches for data mining research.

III. Clustering Technique

A. *Connectivity based clustering (hierarchical clustering)*

Connectivity based clustering, also known as hierarchical clustering, is based on the core idea of objects being more related to nearby objects than to objects farther away. As such, these algorithms connect "objects" to form "clusters" based on their distance. A cluster can be described largely by the maximum distance needed to connect parts of the cluster. At different distances, different clusters will form, which can be represented using a dendrogram, which explains where the common name "hierarchical clustering" comes from: these algorithms do not provide a single partitioning of the data set, but instead provide an extensive hierarchy of clusters that merge with each other at certain distances. In a dendrogram, the y-axis marks the distance at which the clusters merge, while the objects are placed along the x-axis such that the clusters don't mix.

While these methods are fairly easy to understand, the results are not always easy to use, as they will not produce a unique partitioning of the data set, but a hierarchy the user still needs to choose appropriate clusters from. The methods are not very robust towards outliers, which will either show up as additional clusters or even cause other clusters to merge (known as "chaining phenomenon", in particular with single-linkage clustering). In the data mining community these methods are recognized as a theoretical foundation of cluster analysis, but often considered obsolete. They did however provide inspiration for many later methods such as density based clustering [11] and [14].

B. *Centroid-based clustering*

In centroid-based clustering, clusters are represented by a central vector, which may not necessarily be a member of the data set. When the number of clusters is fixed to k, k-means clustering gives a formal definition as an optimization problem: find the k cluster centers and assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized.

The optimization problem itself is known to be NP- hard, and thus the common approach is to search only for approximate solutions. A particularly well known approximate method is Lloyd's algorithm, often actually referred to as "k-means algorithm". It does however only find a local optimum, and is commonly run multiple times with different random initializations. Variations of k-means often include such optimizations as choosing the best of multiple runs, but also restricting the centroids to members of the data set (k-medoids), choosing medians (k- medians clustering), choosing the initial centres less randomly (K-means++) or allowing a fuzzy cluster assignment.

Most k-means-type algorithms require the number of clusters k to be specified in advance, which is considered to be one of the biggest drawbacks of these algorithms. Furthermore, the algorithms prefer clusters of approximately similar size, as they will always assign an object to the nearest centroid. This often leads to incorrectly cut borders in between of clusters (which is not surprising, as the algorithm optimized cluster centres, not cluster borders).

K-means has a number of interesting theoretical properties. On one hand, it partitions the data space into a structure known as Voronoi diagram. On the other hand, it is conceptually close to nearest neighbour classification and as such popular in machine learning [12] and [15].

C. *Distribution-based clustering*

The clustering model most closely related to statistics is based on distribution models. Clusters can then easily be defined as objects belonging most likely to the same distribution. A nice property of this approach is that this closely resembles the way artificial data sets are generated: by sampling random objects from a distribution. Distribution-based clustering is a semantically strong method, as it not only provides you with clusters, but also produces complex models for the clusters that can also capture correlation and dependence of attributes. However, using these algorithms puts an extra burden on the user: to choose appropriate data models to optimize, and for many real data sets, there may be no mathematical model available the algorithm is able to optimize (e.g. assuming Gaussian distributions is a rather strong assumption on the data) [13] and [16].

D. *Density-based clustering*

In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set. Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points. The most popular density based clustering method is DBSCAN. In contrast to many newer methods, it features a well-defined cluster model called "density reach ability". Similar to linkage based clustering; it is based on connecting points within certain distance thresholds. However, it only connects points that satisfy a density criterion, in the original variant defined as a minimum number of other objects within this radius. A cluster consists of all density-connected objects (which can form a cluster of an arbitrary shape, in contrast to many other methods) plus all objects that are within these objects range. Another interesting property of DBSCAN is that its complexity is fairly low - it requires a linear number of range queries on the database - and that it will discover essentially the same results (it is deterministic for core and noise points, but not for border points) in each run, therefore there is no need to run it multiple times. OPTICS is a generalization of DBSCAN that removes the need to choose an appropriate value for the range parameter , and produces a hierarchical result related to that of linkage clustering. Density-Link-Clustering combines ideas from single-linkage clustering and OPTICS, eliminating the parameter entirely and offering performance improvements over OPTICS by using a tree index

IV. CONCLUSION

Clustering is one of the most essential steps in data mining. It is the process of grouping data items based on similarity between elements in a cluster and dissimilarities between clusters. In this paper we have provided an overview of the broad classification of clustering algorithms such as partitioning, hierarchical, density based and grid based methods. Under partitioning methods, we have discussed k-means, and its variant k-medoids. Under hierarchical, we have discussed the two approaches which are the top-down approach and the bottom-up approach. We have also discussed the DBSCAN and OPTICS algorithms under the density based methods. Finally we have discussed the STING and CLIQUE algorithms under the grid based methods. It is to be noted that rapid changes in computer science will necessitate changes in clustering algorithms built upon the existing methods such as the ones discussed here.

REFERENCE

[1] G.L. Pappa and A.A. Freitas, Automating the Design of Data Mining Algorithms. An Evolutionary Computation Approach, Natural Computing Series, Springer, 2010

[2] A. Darwiche, Modeling and Reasoning with Bayesian Networks, Cambridge University Press, 2009

[3] G.F. Cooper, P. Hennings-Yeomans, S. Visweswaran and M. Barmada, “An Efficient Bayesian Method for Predicting Clinical Outcomes from Genome-Wide Data”, AMIA 2010 Symposium Proceedings, 2010, pp. 127-131

[4] M. Garofalakis, D. Hyun, R. Rastogi and K. Shim, “Building Decision Trees with Constraints”, Data Mining and Knowledge Discovery, vol. 7, no. 2, 2003, pp. 187 – 214

[5] T.M. Mitchell, Machine Learning, McGraw-Hill Companies, USA, 1997

[6] Y. Singh Y, A.S. Chauhan, “Neural Networks in Data Mining”, Journal of Theoretical and Applied Information Technology, 2005, pp. 37-42

[7] Z. Pawlak, “Rough sets”, International Journal of Computer and Information Sciences, 1982, pp. 341-356

[8] L. Tari, C. Baral and S. Kim, “Fuzzy c-means clustering with prior biological knowledge”, Journal of Biomedical Informatics, 42(1), 2009, pp. 74-81

[9] N.N. Karnik, J.M. Mendel and Q. Liang, “Type-2 Fuzzy Logic Systems”, IEEE Transactions on Fuzzy Systems, Vol. 7, No. 6, 1999, 643-658

[10] J.R. Castro, O. Castillo and L.G. Martínez, “Interval Type-2 Fuzzy Logic Toolbox”, Engineering Letters 15(1), 2007, pp. 89-98

[11] Lan Yu, “Applying Clustering to Data Analysis of Physical Healthy Standard”, 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010), pp. 2766-2768.

[12] Yun Ling and Hangzhou, “Fast Co-clustering Using Matrix Decomposition”, IEEE 2009 Asia- Pacific Conference on Information Processing, pp. 201-204.

[13] Vignesh T. Ravi and Gagan Agrawal, “Performance Issues in Parallelizing Data-Intensive Applications on a Multi-core Cluster”, 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 308-315.

[14] David Pettinger and Giuseppe Di Fatta, “Scalability of Efficient Parallel K-Means”, IEEE e- Science 2009 Workshops, pp. 96-101.

[15] Madjid Khalilian, Farsad Zamani Boroujeni, Norwati Mustapha, Md. Nasir Sulaiman, “K-Means Divide and Conquer Clustering”, IEEE 2009, International Conference on Computer and Automation Engineering, pp. 306-309.

[16] F. Yang, T. Sun, C. Zhang, An efficient hybrid data clustering method based on K-harmonic means, and Particle Swarm Optimization, Expert Systems with Applications 2009, pp. 9847–9852.