ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

Clustering with Decision Trees: Divisive and Agglomerative Approach

Lauriane Castin and Benoit Frénay

NADI Institute - PReCISE Research Center
Université de Namur - Faculty of Computer Science
Rue Grandgagnage 21 - 5000 Namur, Belgium

Abstract. Decision trees are mainly used to perform classification tasks. Samples are submitted to a test in each node of the tree and guided through the tree based on the result. Decision trees can also be used to perform clustering, with a few adjustments. On one hand, new split criteria must be discovered to construct the tree without the knowledge of samples labels. On the other hand, new algorithms must be applied to merge subclusters at leaf nodes into actual clusters. In this paper, new split criteria and agglomeration algorithms are developed for clustering, with results comparable to other existing clustering techniques.

1 Introduction

Decision trees are well-known tools to solve classification problems. They use only a small subset of features to classify the samples and are therefore interesting for their computational performances. Decision trees can be extended to clustering problems with the constraint that labels of the samples are unknown during the training process. The definition of a new split criterion that does not require the labels for the tree construction is therefore needed. In the literature, most papers refer to the CLUS algorithm that uses the variance reduction as split criterion. Years have passed and almost no other algorithm was presented.

The objective of this paper is to propose an innovative algorithm allowing to perform clustering with decision trees. It takes the unlabeled dataset and the desired number of clusters as input, and outputs a decision tree. Unseen samples can be guided through the tree to discover to what cluster they belong.

2 State of the Art

Clustering *aims at identifying groups of similar objects and, therefore helps to discover distribution of patterns and interesting correlations in large datasets* [1]. This definition means that the dataset is divided into several groups, called clusters, in such a way that clusters consist of samples that are similar to each other, and are dissimilar from samples belonging to other clusters.

2.1 k-means

k-means is the most famous clustering algorithm. In order to find clusters, a known number of centroids is randomly located in the data space. Then two steps alternate. At first, each sample is assigned to its closest centroid, then the

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

centroid position is updated as the mean of the assigned samples. The process iterates until no change of assignment occurs anymore.

2.2 Hierarchical Clustering

In hierarchical clustering, a tree of clusters, called a *dendogram*, is built. All samples belong to the root cluster, and while descending down the tree, they are divided into different partitions according to some characteristics. Samples from the leaf clusters therefore share the characteristics of all their ancestor clusters.

CURE [2] (1998) and CHAMELEON [3] (1999) are part of the agglomerative hierarchical clustering techniques. They start with single-sample clusters and merge the most appropriate ones to form new clusters until all samples belong to the same cluster. In CURE, a fixed number of well-scattered points is chosen in each cluster as representatives of the cluster. After the identification of all representatives, the distance between each pair of representatives from different clusters is measured. The clusters represented by the two closest representatives are merged. New representatives are then computed for each cluster and the operation is repeated until a predefined number of clusters is reached. In CHAMELEON, a sparse k-nearest neighbor graph is constructed and a first set of small clusters is retrieved from it. Then, two clusters are merged if the interconnectivity and the closeness of the two merged clusters is higher compared to the internal inter-connectivity and closeness of the separated clusters.

2.3 Decision Trees

A decision tree consists of a root node, branches with regular nodes and leaf nodes. Each node of the tree proposes a test on a feature, while each branch specifies the possible values taken by this feature. Starting from the root, the sample to classify is tested in each node and then guided to a certain branch following the test result, towards the leaf that will return the label of the sample.

Unlike other classification and clustering methods, decision trees only classify samples by looking on a subset of relevant features instead of the full set. This is a great computational advantage, but also induces a limited accuracy. Other advantages are that non-experts can easily understand them, they are robust to outliers and compactly stored. Unfortunately, the design of an optimal tree is difficult especially if the cluster boundaries are complicated.

For the construction of a basic decision tree with the ID3 algorithm (Iterative Dichotomiser 3), for each node, the most relevant feature and threshold are selected thanks to an impurity measure. The impurity is computed from the features only because no label is available in clustering. Most papers in the literature build on the CLUS algorithm [4], using variance reduction as a split criterion and stopping the growth of the tree when the gain in variance reduction becomes small. DIVCLUS-T [5] is another algorithm using a variance-like split criterion. For both these algorithms, even if the data is structured thanks to the decision tree, clusters remain difficult to identify because the label of the training samples are necessary to determine what leaf nodes belong to the same actual cluster. In this paper, this limit is overcome thanks to the clusters agglomeration.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

3 Proposed approach

In this section, the algorithm framework is presented. Then new split criteria and agglomeration mechanisms are introduced to perform clustering with decision trees on unlabeled datasets. They are combined in Section 4.

3.1 Framework

The proposed algorithm consists of the 4 steps shown in Fig. 1.

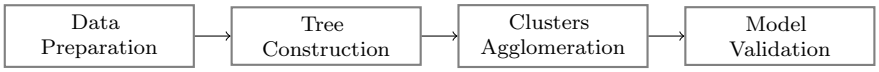


Fig. 1: Algorithm framework consists in four steps.

The data preparation step covers for example the possible normalizations applied to the dataset. The tree construction step builds the decision tree from the training set based on a specific split criteria. At this stage, each leaf node of the tree represents a sub-cluster and there are certainly more leaves than the actual number of clusters in the dataset. Sub-clusters referring to the same actual cluster must be merged together, this is done in the specifically implemented clusters agglomeration step. Finally, in the validation step, metrics are computed to assess the performance of the tree to classify the test set. The tree construction and the clusters agglomeration are the main focus of this research.

3.2 Split criteria

A modified ID3 algorithm constructs the tree thanks to a chosen split criterion. Several thresholds on this value are compared to create two partitions. As clustering is part of the unsupervised techniques, this computation only takes the features of the samples as input. Two innovative split criteria were designed.

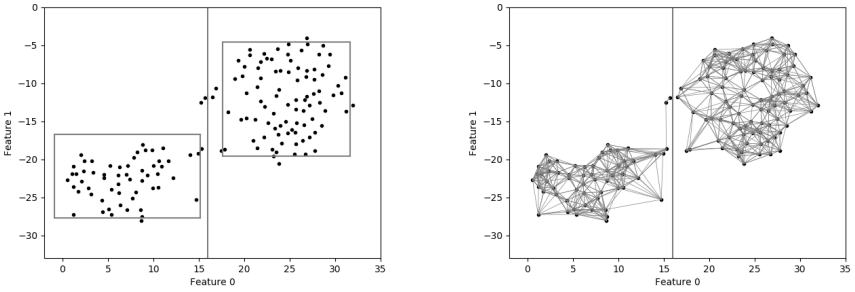
3.2.1 Box Volume

The objective of the box volume split criterion is to localize the clusters in each partition by contouring them with a box. To achieve this, each feature is regarded separately. At first, a lower bound and a upper bound are placed around the mean. Then, the number of samples with feature value in between the two bounds are counted. Step by step, the bounds are moved apart and the samples counted again. When 95% of the samples are captured, the width of the box is defined for that dimension, the box volume is the product of all widths. A partitioning with two compact clusters is desirable, in this case the sum of both boxes volumes is smaller. Figure 2a shows examples of boxes shapes.

3.2.2 Graph Closeness

For this split criterion, a k-nearest neighbors graph is constructed from all the samples of the node to split. The weight of each edge is the euclidean distance

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.



(a) Box Volume: for each partition, the impurity is the area of the box contouring 95% of samples along each feature axis. Notice that outliers are automatically excluded from boxes.

(b) Graph Closeness: from an original k-nearest neighbors graph, edges crossin the partition border are removed and the impurity for each partition is the inverse of the sum of remaining edges weights.

Fig. 2: Two split criteria for two-dimensional dataset.

between the linked points. When defining the two partitions, edges going from one partition to the other are removed from the original graph, as shown on Fig. 2b. For each side, the weights of remaining edges are summed and the impurity is computed as the inverse of this sum. This algorithm encourages cuts discriminating well-separated clusters because only few edges are deleted. This method is slightly inspired by CHAMELEON [3] where the graph is used to merge the clusters instead of splitting the data space.

3.3 Clusters Agglomeration

After the construction of the tree, each leaf node represents a region in data space where a possible sub-cluster is located. Because no stop criterion was defined, many actual clusters are divided in smaller sub-clusters. This section will describe how sub-clusters are merged until the number of actual clusters is reached. At that time, the same label is assigned to all the merged sub-clusters. Thanks to the following technique, nodes that do not share the same ancestors might still be assigned the same label if they are close in the data space.

3.3.1 Prototypes

For each sub-cluster, one or several prototypes, or representatives, can be defined. The mean can act as a single prototype, or similarly to the CURE algorithm [2], a set of samples can be randomly picked and moved towards the mean. Both approaches were implemented here. No matter the number of prototypes per sub-cluster, the next steps are identical: the pair of closest prototypes is identified and the related sub-clusters are merged. Then the prototypes positions are re-computed from the new groups. These two steps are repeated until the number of actual clusters in the dataset is reached.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

3.3.2 Box Distance

Similarly to the Box Volume split criterion, a box contours each sub-cluster. For each pair of boxes, the distance between boxes edges is computed along each dimension and those distances are summed. The two sub-clusters having the lowest sum are merged and a new box is created for the new sub-cluster.

3.3.3 Graph Connectivity

This agglomeration mechanism is based on the intuition that actual clusters are well-connected. A k-nearest neighbors graph is constructed. For each pair of sub-clusters, the number of edges going from one to the other is computed. The two sub-clusters connected with the higher number of edges are merged.

4 Experiments

4.1 Algorithm Steps

Data Preparation For the experiments, the datasets of handwritten figures digits [6] and MNIST [7] were used with 5 or 10 classes, and so the same number of clusters were to discriminate. As decision trees are not efficient on raw pixels and to make sure that clusters can be identified, a dimensionality reduction with t-SNE [8] was applied to work in 2 or 3 dimensional data space. Two thirds of the data is dedicated to the training set, while the last third forms the test set.

Tree Construction The adapted ID3 algorithm was naively implemented for the tree construction: it has no pruning nor early stopping criterion. However, the maximum tree depth can be chosen to stop the growth and avoid overfitting.

Model Validation Digits and MNIST datasets actually have labels indicating what figure is on the image. Those labels were never read during the tree construction or the cluster agglomeration, but they can act as ground truth to evaluate the accuracy of the algorithm. Table 1 presents the mean and standard deviation of the misclassification rate over 30 repetitions. The trees with maximal depth 5 implementing the new split criteria and the new agglomeration mechanisms are compared to other clustering techniques: a supervised decision tree using the original labels, a k-means directly applied on the data without any tree involved and a supervised decision tree using the labels given by k-means.

Discussion According to Table 1, the supervised decision tree presents the lowest misclassification rates thanks to the extra knowledge of the labels. Direct k-means and k-means followed by a decision tree show slightly better results than the new algorithm. However, the new algorithm results seem promising and different tracks are open to improve it, as discussed in the conclusion.

4.2 Results

During the research, all possible combinations of the two split criteria and the three clusters agglomeration mechanisms were compared. The three best are presented in Table 1. Graph Closeness performs better than Box Volume because the latter tends to crop clusters edges. Prototype-based agglomerations show

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

			A	B	C	D	E	F
DIGITS	2D	5C	0.7 ± 0.2	6.2 ± 1.5	6.3 ± 1.5	8.5 ± 2.2	8.2 ± 1.9	14.1 ± 3.3
		10C	5.0 ± 1.3	12.8 ± 1.5	13.5 ± 1.5	17.8 ± 1.5	20.0 ± 1.9	22.8 ± 2.7
	3D	5C	1.6 ± 0.4	15.5 ± 3.5	16.0 ± 4.0	13.9 ± 3.0	11.8 ± 2.6	14.0 ± 3.0
		10C	7.6 ± 1.1	13.0 ± 1.6	14.7 ± 1.6	20.9 ± 1.7	26.8 ± 3.5	28.5 ± 3.6
MNIST	2D	5C	5.5 ± 0.5	17.5 ± 2.5	18.0 ± 2.4	21.0 ± 2.5	22.9 ± 2.7	33.4 ± 4.4
		10C	8.1 ± 4.4	11.3 ± 5.9	12.0 ± 6.2	13.4 ± 6.8	13.6 ± 7.0	18.1 ± 9.1
	3D	5C	6.2 ± 0.6	17.2 ± 3.0	18.4 ± 3.2	24.1 ± 3.7	27.4 ± 3.5	37.3 ± 5.9
		10C	25.9 ± 1.3	33.5 ± 1.5	37.3 ± 1.8	44.1 ± 1.8	45.7 ± 1.9	57.6 ± 2.4

Table 1: Misclassification rate for (A) Supervised decision tree, (B) Direct application of k-means, (C) k-means followed by supervised decision tree, (D) Box Volume combined with Single Prototype, (E) Graph Closeness combined with Three Prototypes , (F) Graph Closeness combined with Graph Connectivity.

better misclassification rates, followed by Graph Connectivity. Box Distance was excluded because boxes can be close along one dimension but far apart in another, leading to weak performances.

5 Conclusion

In this paper, two split criteria and three clusters agglomeration mechanisms were designed to perform clustering with decision trees. So far in the literature, mostly the CLUS algorithm was used, but it needs the training samples labels to recognize the clusters. Our algorithm is able to identify clusters inside an unlabeled dataset. In order to improve the results, either pruning or a stopping criterion could be introduced in the tree construction. Interesting research would also assess the performance of the algorithm on more than three-dimensional datasets and add cross-validation on maximal depth hyper-parameter. The performance of such algorithm could also be reviewed on irregular shaped clusters.

References

[1] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of intelligent information systems*, 17(2):107–145, 2001.

[2] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: an efficient clustering algorithm for large databases. In *ACM Sigmod Record*, volume 27, pages 73–84, 1998.

[3] George Karypis, Eui-Hong Han, and Vipin Kumar. CHAMELEON: Hierarchical clustering using dynamic modeling. *IEEE Computer*, 32(8):68–75, 1999.

[4] Hendrik Blockeel, Luc De Raedt, and Jan Ramon. Top-down induction of clustering trees. In *Proc. of the 15th International Conference on Machine Learning*, pages 58–63, 1998.

[5] Marie Chavent, Yves Lechevallier, and Olivier Briant. DIVCLUS-T: A monothetic divisive hierarchical clustering method. *Computational Statistics & Data Analysis*, 52(2):687–701, 2007.

[6] Moshe Lichman. UCI machine learning repository, 2013.

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[8] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.