Augmenting automatic clustering with expert knowledge and explanations

Szymon Bobek[1,2][0000−0002−6350−8405] and
Grzegorz J. Nalepa[1,2][0000−0002−8182−4225]

[1] Jagiellonian Human-Centered Artificial Intelligence Laboratory (JAHCAI) and Institute of Applied Computer Science, Jagiellonian University, 31-007 Kraków, Poland
[2] AGH University of Science and Technology
{szymon.bobek,grzegorz.j.nalepa}@uj.edu.pl

Abstract. Cluster discovery from highly-dimensional data is a challenging task, that has been studied for years in the fields of data mining and machine learning. Most of them focus on automation of the process, resulting in the clusters that once discovered have to be carefully analyzed to assign semantics for numerical labels. However, it is often the case that such an explicit, symbolic knowledge about possible clusters is available prior to clustering and can be used to enhance the learning process. More importantly, we demonstrate how a machine learning model can be used to refine the expert knowledge and extend it with an aid of explainable AI algorithms. We present our framework on an artificial, reproducible dataset.

Keywords: data mining · explainable AI · clustering

1 Introduction

An effective analysis of data can often pose a major challenge in cases where data is highly-dimensional, produced in fast rate and large volumes (i.e. big data). This is where methods of Artificial Intelligence prove to be useful. Moreover, besides the use of data mining techniques in order to build machine learning models, the human expert knowledge regarding the specificity of domain of interest should be used.

Such a case most often arises in Industry 4.0 that aims at using number of information and communications technology solutions for the monitoring and optimization of industrial processes. The installations in modern factories are equipped with many of sensors gathering data about the operation of the machines involved in these processes.

In this work we focus on automated discovery of device states from machinery sensor log to enrich the expert knowledge about machinery operational states. Our main goal was to develop a workflow, which would provide a mechanism for detecting device states that can be applied to different types of industrial machinery. We confronted it with states that were discovered with knowledge-based approach to prove its validity and expand the knowledge-base itself with an usage of eXplainable Artificial Intelligence (XAI) algorithms. For this purpose we propose two algorithms. The first one is for *splitting* clusters that possibly scramble within two or more concepts. The second one is for *merging* expert clusters possibly represent the same concept, and therefore

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_48

can be considered redundant. The recommendations are justified by rules, explaining why such suggestions were made. The final decision on whether to trust recommendation or discard them is left to an expert. The whole process is iterative and can be repeated until the convergence is achieved. In this paper we limit the discussion only to artificially generated samples to provide full reproducibility of the experiments.

This work is carried out in the CHIST-ERA Pacmel project. [3] The project is oriented at the development of novel methods of knowledge modeling and intelligent data analysis in Industry 4.0.

The rest of the paper is organized as follows: In Sect. 2 we discuss selected challenges regarding clustering. Then in Sect. 3 we introduce our approach regarding knowledge augmented clustering. We summarize the paper in Sect.4.

2 Clustering highly-dimensional data

Clustering aims at unfolding hidden patterns in data to discover similar instances and group them under common cluster labels. This task is often performed to either discover unknown groups, to automate the process of discovering possibly known groups or for segmentation of data points into arbitrary number of segments. Either of the above can be done in unsupervised, semi-supervised or supervised manner.

The problem of effective analysis of clustering results, and bringing semantics into the clustering results has also been investigated. In [1] authors focus on solutions which assist users to understand long time-series data by observing its changes over time, finding repeated patterns, detecting outliers, and effectively labeling data instances. It is performed mostly via visualization layer over data that dimensionality was reduced with UMAP [6] allowing 2D/3D plotting. However, no explicit knowledge is used in this method to enhance the process of clusters analysis.

In [4] the Grouper framework was presented which is an interactive approval, refinement or decline toolkit for analysis of results of clustering. It combines the strength of algorithmic clustering with the usability of visual clustering paradigm. In [11] similar approach was presented, however it assumes more interactions with visualized clusters that alters the cluster layout. Yet, neither of these use any kind of formalized knowledge neither for clustering nor after it for refinement. Therefore, the knowledge input by en expert in a form of interactions in the system is lost for further re-use.

The human-in-the-loop paradigm was also investigated in the clustering algorithms. In [12] a similar approach was used as in our solution, where the contextual information and user feedback is used to merge clusters of photographs into larger groups. However, no prior knowledge is used in clustering, nor extended at the end.

The biggest disadvantage of all of the above methods is that they do not explicitly use nor update domain knowledge. Therefore new knowledge, even if discovered by cooperation of AI and expert, is hidden into complex models and not reusable for future.

Using background knowledge in DM has been proposed in the area of semantic data mining, where the formalized expert knowledge is used for domain-specific configuration to improve the overall results of DM/ML algorithms [3].Initial approaches reusing

[3] See the project webpage at http://PACMEL.geist.re.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_48

knowledge and experiences in DM for configuring DM tasks have been discussed in [2]. However, in those approaches, domain knowledge is only used in a very specific setting, and there is lack of feedback loop that allows for knowledge flow in opposite direction.

Therefore, in our approach we use explainable AI algorithms, that aim to inverse the process of encoding knowledge into black-box models and allow for more insight into decisions made by machine learning or data mining algorithms. This closes the feedback loop between domain experts and machine-learning algorithms by allowing knowledge exchange between an expert and ML algorithm. Although there exist a variety of XAI methods that allow explaining ML models decision [9, 5, 10, 8, 7], in our work we will focus on Anchor [10] which produces rule-based, explanations that can easily be integrated with domain knowledge encoded with the same formalism.

In the following sections we will describe our framework in more details.

3 Knowledge augmented clustering

Although obtaining high quality automated clustering of data is an important initial step for utilizing our framework, we skip this step in the discussion. The main goal of the work presented in this paper is to refine the initial clustering with XAI methods and expert knowledge via splits and merges of existing clusters.

In both of the cases for cluster splits and merges we assume that there exists a set of clusters obtained with an utilization of expert knowledge, denoted as:

$$E = \{E_1, E_2, \ldots, E_n\}$$

This set of clusters needs to be refined with complementary clustering performed with automated clustering algorithms. This clustering forms separate set of clusters, possibly of different size then E and is denoted as:

$$C = \{C_1, C_2, \ldots, C_m\}$$

For both sets we calculate confusion matrix M of size (n, m) where number at the intersection of i-th row and k-th column holds number of data points assigned both by cluster labeling to cluster E_j and automated clustering to cluster C_k.

Based on confusion matrix M we calculate two helper matrices for splitting and merging strategies defined respectively by Equations (1) and (2).

$$H^{split}_{i,j} = \frac{M_{i,j}}{H(M_i) \sum\limits_{j \in 1 \ldots m} M_{i,j}} \tag{1}$$

Where $H(M_j)$ is entropy calculated for j-th column (i.e. C_j cluster). The measure defines consistency of automated clustering with expert clustering, normalized by number of points. The H^{merge} measure is $l2$ normalized matrix M along row axis.

$$H^{merge}_{i,j} = \frac{M_{i,j}}{||M_i||_2} \tag{2}$$

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_48

These two matrices are later used for the purpose of generation of split and merge recommendations. Fig. 1 depicts the two simplified datasets with two possible scenarios covered by our method. These datasets will be used to better explain mechanisms for splitting and merging recommendations discussed in following sections.

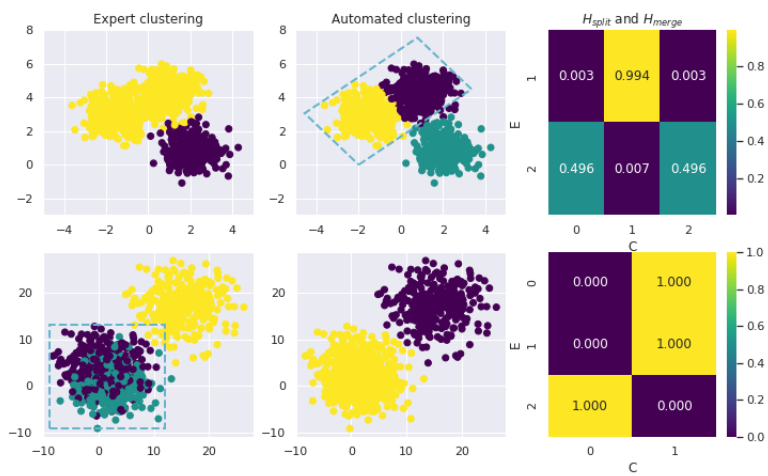


Fig. 1: Synthetic datasets with clusters to split (top row) and clusters to merge (bottom row). Columns in the figure represent clustering preformed with expert knowledge, automated clustering, and H_{split} matrix (upper) and H_{merge} matrix (lower). Dotted lines define bounding boxes for the decision stump explanation mechanism.

3.1 Recommendation generation

Having created the H^{split} and H^{merge} matrices, we generate two types recommendations out of it: *splitting* and *merging*.

Splitting. This recommendation aim at discover clusters that were incorrectly assigned by expert knowledge. Such a case was depicted in Fig. 1 in the upper left plot. This operation can be performed using H^{split} matrix in a straightforward way. The cluster that is recommended for splitting is chosen by investigating values corresponding to it in H^{split} matrix. Values that lie on the intersection of investigated expert cluster and automated cluster, and that are greater than defined threshold ϵ_s are marked as candidates for splitting:

$$Candidates_i = \left\{ C_j : H^{split}_{i,j} > \epsilon_s \right\}$$

For the example in Fig. 1 the recommendation will look as follows:

```
SPLIT  EXPERT CLUSTER  E_2
INTO CLUSTERS  [(C_0, C_2)]  (Confidence 0.98)
```


DOI: 10.1007/978-3-030-77970-2_48

The confidence of split is an average of $H_{i,j}^{split}$ values associated with candidates for splitting normalized by the maximum entropy. The maximum entropy depends on the number of expert clusters to merge, and equals 0.5 in this recommendation case.

Merging. This recommendation goal is to detect concepts that were incorrectly labelled by an expert knowledge as two clusters. Such a case is depicted in Fig. 1 in lower left plot. Candidates for merging are chosen using H^{merge} matrix. Because the matrix is $l2$ normalized along rows, calculating dot product of selected rows produces cosine similarity between them. This cosine similarity reflect the similarity in the distribution of data points spread over the automated discovered clusters. If two expert clusters have similar distribution of points over automatically discovered clusters this *might* be a premise that they share the same concept and should be merged. Such a case was depicted in Fig. 1 in lower right plot. Similarly as in the case of splitting a threshold ϵ_m is defined arbitrarily that denotes the lower bound on the cosine similarity between clusters to be considered as merge candidates.

$$Candidates_m = \left\{E_j, E_k : sim(H_j^{merge}, H_j^{merge}) \geq \epsilon_m\right\}$$

The merge recommendation is generated as follows. The confidence value is calculated as a cosine similarity between rows associated to candidates E_0, E_1 in H^{merge} matrix.

```
MERGE  EXPERT CLUSTER E_0  WITH  EXPERT CLUSTER E_1
INTO  CLUSTER C_1 # (Confidence 1.0)
```

In the next section, the justification of the split and merge recommendation are discussed.

3.2 Recommendation explanation

Once the recommendation is generated it is augmented with an explanation. Depending on the recommendation type, the explanation is created differently.

Splitting recommendation. In case of this type of recommendation we transform the original task from clustering to classification, taking automatically discovered cluster labels as target values for the classifier.

Then, we explain the decision of a classifier to present an expert *why* and *how* two (or more) clusters $C_i, C_j, \ldots, C_n$ that were formed by splitting original one are different from each other. An explanation is formulated in a form of a rule that uses original features as conditional attributes, to help expert better understand the difference between splitting candidates. We use the Anchor algorithm for that [10].

The explanation for the splitting of cluster E_2 presented in Fig. 1 looks as follows:

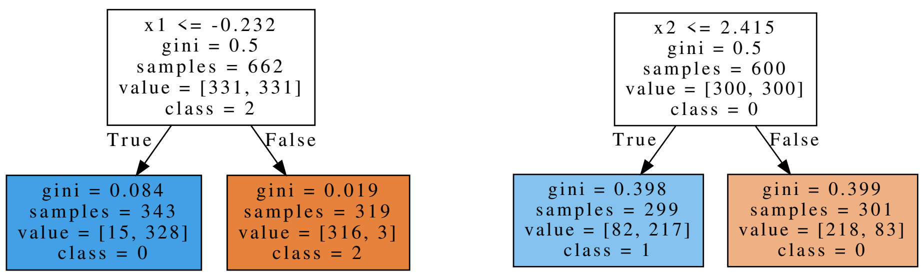
```
C_0: x1 > 0.88 AND x2 > 3.61 (Precision: 1.00, Coverage: 0.21)
C_2: x1 <= -1.51 (Precision: 1.00, Coverage: 0.25)
```

If the difference is important, the clusters can be split and the rules generated above can be added to knowledge base. The final decision on weather splitting E_2 into C_0 and C_2 is needed, is left to the expert.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_48

Alternatively a decision stump can be build on the dataset narrowed to points that form clusters candidates C_0 and C_1 and its visual form can also be presented to an expert. It is worth noting that the decision stump can be different than Anchor rule, as the former is build on just a fraction of data, while Anchor takes into consideration whole dataset.

The decision stump for the case discussed in this section is given in Fig. 2a. The cyan bounding-box in Fig. 1 in the upper middle plot roughly defines the dataset used for building the decision stump.



(a) Decision stump for the split. (b) Decision stump for the merge.

Fig. 2: Decision stumps for explanations of recommendations.

Merging recommendation. In explanation for merging recommendation we use the same approach as previously described. The difference is that the classification models are now trained with expert labels as target.

After that the explanation that answers the questions how two expert clusters E_i and E_j are different from each other while looking at them not through the definition in knowledge base, but in data.

The answer to this question is given in a form of Anchor rules, and the final decision left to an expert. The explanation for the case presented in Fig. 1 is given below:

```
E_0: x2 <= 5.16 AND x1 > 0.23 (Precision: 0.74, Coverage: 0.32)
E_1: x1 <= 3.73 (Precision: 0.58, Coverage: 0.50)
```

Similarly to splitting explanation a decision stump can be created as presented in Fig. 2b. The results are again slightly different than in case of the Anchor explanation, as the decision stump is build only on a fraction of the data and can omit other variable dependencies. The cyan bounding-box in Fig. 1 in the lower left plot roughly defines the dataset used for building decision stump.

4 Summary

In this paper we presented a framework for expert knowledge extension with a usage of clustering algorithms for multidimensional time series. We described how automated mechanism for labeling device operational states can be used to refine expert-based labeling and demonstrated its functionality on a synthetic, reproducible scenario.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_48

These refinements was defined by us as *splits* and *merges* of expert labeling and were augmented with detailed explanations. The explanations were formulated as rules and therefore can be easily interpreted incorporated with expert knowledge.

For the future works, we plan to extend the framework with additional methods supporting splits and merges. In particular we would like to exploit different linkage methods known form hierarchical clustering for merges and clustering metrics, such as silhouette score for splits.

Acknowledgements

The paper is funded from the PACMEL project funded by the National Science Centre, Poland under CHIST-ERA programme (NCN 2018/27/Z/ST6/03392). The authors are grateful to ACK Cyfronet, Krakow for granting access to the computing infrastructure built in the projects No. POIG.02.03.00-00-028/08 ”PLATON - Science Services Platform” and No. POIG.02.03.00-00-110/13 ”Deploying high-availability, critical services in Metropolitan Area Networks (MAN-HA)”

References

1. M. Ali, M. W. Jones, X. Xie, and M. Williams. TimeCluster: Dimension reduction applied to temporal data for visual analytics. *Vis. Comput.*, 35(6–8):1013–1026, June 2019.
2. M. Atzmueller. Experience Management with Task-Configurations and Task-Patterns for Descriptive Data Mining. In *Proc. of KESE 2007, 30th German Conference on Artificial Intelligence (KI-2007)*, 2007.
3. M. Atzmueller and D. Seipel. Declarative Specification of Ontological Domain Knowledge for Descriptive Data Mining (Extended version). In *Proc. 18th International Conference on Applications of Declarative Programming and Knowledge Management*, 2008.
4. A. Coden, M. Danilevsky, D. Gruhl, L. Kato, and M. Nagarajan. *A Method to Accelerate Human in the Loop Clustering*, pages 237–245.
5. S. M. Lundberg, G. G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S. Lee. Explainable AI for trees: From local explanations to global understanding. *CoRR*, abs/1905.04610, 2019.
6. L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.
7. F. Mujkanovic, V. Doskoč, M. Schirneck, P. Schäfer, and T. Friedrich. timexplain – a framework for explaining the predictions of time series classifiers, 2020.
8. P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann. Explainability methods for graph convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10764–10773, 2019.
9. M. T. Ribeiro, S. Singh, and C. Guestrin. “Why should i trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.
10. M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI*, 2018.
11. J. Wenskovitch and C. North. Observation-level interaction with clustering and dimension reduction algorithms. HILDA’17, New York, NY, USA, 2017. Association for Computing Machinery.
12. L. Zhang, D. V. Kalashnikov, and S. Mehrotra. Context-assisted face clustering framework with human-in-the-loop. *Int. J. Multim. Inf. Retr.*, 3(2):69–88, 2014.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_48