# Comprehensive Differentiation of Partitional Clusterings

Lars Schütz[1,2,a], Korinna Bade[1,b] and Andreas Nürnberger[2,c]

[1]Department of Computer Science and Languages, Anhalt University of Applied Sciences, Köthen (Anhalt), Germany

[2]Faculty of Computer Science, Otto von Guericke University Magdeburg, Magdeburg, Germany

Keywords: Clustering Difference Model, Cluster Comparison, Applications in Planning and Decision Processes.

Abstract: Clustering data is a major task in machine learning. From a user's perspective, one particular challenge in this area is the differentiation of at least two clusterings. This is especially true when users have to compare clusterings down to the smallest detail. In this paper, we focus on the identification of such clustering differences. We propose a novel clustering difference model for partitional clusterings. It allows the computational detection of differences between partitional clusterings by keeping a full description of changes in input, output, and model parameters. For this purpose, we also introduce a complete and flexible partitional clustering representation. Both the partitional clustering representation and the partitional clustering difference model can be applied to unsupervised and semi-supervised learning scenarios. Finally, we demonstrate the usefulness of the proposed partitional clustering difference model through its application to real-world use cases in planning and decision processes of the e-participation domain.

## 1 INTRODUCTION

Machine learning algorithms and models have been successfully applied to numerous domains of our lives for many years. However, one major challenge is that machine learning algorithms and models are not always easy to interpret. In this context, interpretability describes the extent to which a person can understand the cause of a decision (Miller, 2019) or how much a person can reliably predict the result of a machine learning model (Kim et al., 2016). We frequently have to explain the decisions or recommendations that have been generated by machine learning algorithms and models. We should at least be curious about this, especially when these algorithms and models have a significant impact on our environment and personal lives. It is immediately noticeable that humans play a central role when it comes to utilizing machine learning algorithms and models.

Clustering data sets is a very common task in machine learning which we will focus on in this paper. The general clustering objective is to partition a set of data instances into a (pre-defined) number of clusters or partitions. Data instances can either belong to only one cluster (hard clustering) or to multiple clusters at the same time (soft clustering). In this work, we specifically focus on partitional clustering in unsupervised and semi-supervised learning scenarios. In unsupervised clustering, there is no further information about the relationship between the data instances, and there is no prior assignment to a cluster for any of the data instances at hand. Well known clustering algorithms are $k$-means (Lloyd, 1982; MacQueen, 1967), $k$-medoids (Kaufman and Rousseeuw, 1987; Schubert and Rousseeuw, 2019), and fuzzy $c$-means (Bezdek, 1981). In contrast, there is some supervision available in semi-supervised clustering. For instance, this supervision can describe the relationship between parts of the data instances. In this regard, pairwise constraints are commonly used to model whether two data instances must belong to the same cluster (must-link) or to two distinct clusters instead (cannot-link) (Wagstaff et al., 2001). Exemplary semi-supervised clustering algorithms are instance-based pairwise-constrained $k$-means (Basu et al., 2004) and metric-based pairwise-constrained $k$-means (Bilenko et al., 2004).

Different clustering algorithms and algorithm parameter settings lead to different clusterings of the same data set. How can we distinguish between these clusterings? The comprehensive differentiation of clusterings is challenging. We generally have difficulties to track changes or differences (Simons and

[a] https://orcid.org/0000-0003-3303-6619
[b] https://orcid.org/0000-0001-9139-8947
[c] https://orcid.org/0000-0003-4311-0624



Schütz, L., Bade, K. and Nürnberger, A.
Comprehensive Differentiation of Partitional Clusterings.
DOI: 10.5220/0011762000003467
In *Proceedings of the 25th International Conference on Enterprise Information Systems (ICEIS 2023) - Volume 2*, pages 243-255
ISBN: 978-989-758-648-4; ISSN: 2184-4992

Rensink, 2005). For example, while it might be easy for us to differentiate the number of computed clusters per clustering, it might be more difficult for us to identify modified data instances as well as data instances that have different cluster assignments. We may even think there are no differences at all simply because we have not noticed them. We follow the idea that changes should be communicated in order to ensure a sound understanding of the involved machine learning models (Kulesza et al., 2012; Kulesza et al., 2013; Kulesza et al., 2015). This also applies to the clustering differences. But these kind of differences must first be detected before they can be communicated properly. Enabling this detection in a flawless and detailed way is our major motivation.

The detection of clustering differences becomes even more crucial in interactive or human-in-the-loop clustering (Coden et al., 2017). In such clustering scenarios, the user guides the clustering algorithms and models by interacting with the computer systems and software applications that encompass them. This allows the integration of user knowledge, e. g., the user can correct automatically computed assignments of data instances to clusters. In turn, such user interaction can trigger changes in the involved clustering during the clustering process. The clustering might even change several times if we consider frequent user interaction. These clustering differences matter. They do not only allow the user to compare clusterings for evaluating their overall quality. They also enable the user to understand the consequences of her interactions. So in the end, it is possible that the user will be confused during the interactive clustering process when the differences or changes are not communicated properly. However, it must first be possible to detect these differences.

In this paper, our main contribution is the introduction of a novel partitional clustering difference model (Section 4). It allows the computational detection of differences between partitional clusterings. This paper is the first fundamental work following this specific approach. The novel partitional clustering difference model is based on a flexible partitional clustering representation which we also present in this paper (Section 3). Additionally, we demonstrate the applicability and practicability of the model by the means of exemplary but still prominent real-world use cases in planning and decision processes of the e-participation domain (Section 5). Furthermore, we present related work (Section 2), we conclude our research presented in this paper (Section 6), and we point to major directions for future work in this area (Section 7).

## 2 RELATED WORK

Generally, our research is in line with the recommendations of the ethics guidelines for trustworthy artificial intelligence published by the High-level Expert Group on Artificial Intelligence that was set up by the European Commission (High-Level Expert Group on AI, 2019). In their list of four ethical principles and seven key requirements for realizing trustworthy artificial intelligence, they emphasize human agency, human oversight, and explicability of machine learning algorithms and models. This clearly correlates to our motivation. However, they do not provide concrete methods and implementation options for any machine learning algorithm or model. The encouraging point is that the necessity of methods for interpretable machine learning and explainable artificial intelligence has long been recognized (Lipton, 2018; Abdul et al., 2018). However, this broad field is still considered to be very challenging so that related research activities continue to increase for years (Doshi-Velez and Kim, 2017). In particular, a lot of research activities in this area are being carried out for supervised learning scenarios, especially concerning the understanding of artificial deep neural networks, while efforts in the unsupervised and semi-supervised learning scenarios seem to be less frequent.

We focus on the differences between clusterings. We want to make clustering changes transparent. The work that examines the general comparison or differentiation of clusterings is clearly closely related research. In this regard, there are many measures that compare different clusterings (Wagner and Wagner, 2007). Commonly, these measures are classified into at least three groups: pair-counting measures, e. g., the Rand index (Rand, 1971), measures based on overlapping sets, e. g., the "Meilă-Heckerman-Measure" (Meilă and Heckerman, 2001), and information theoretic measures (Vinh et al., 2010), e. g., the variation of information (Meilă, 2003; Meilă, 2005; Meilă, 2007). These measures are sometimes used to evaluate the quality of clusterings in comparison to ground truth labels if these are available, and these measures are also used to check how similar or dissimilar clusterings are. Unfortunately, these measures do not consider the differences between at least two clusterings in relation to the data instances and their cluster assignments on a lower level, i. e., changes to cluster assignments of individual data instances cannot be determined by relying only on these measures. Instead, these measures allow a high-level comparison of clusterings. The similarity or dissimilarity of clusterings is only represented by a number which is the numerical value of the specific measure used. It remains unclear where the differences between the clusterings exactly are. We, however, consider a full description of differences between clusterings on the lowest level.

Other closely related research that addresses multiple different clusterings are meta clustering and consensus clustering methods. For example, the meta clustering algorithm (Caruana et al., 2006) generates various reasonable and qualitatively different clusterings on a base level. These are then presented to a user on a meta level, e. g., by clustering similar clusterings. In the end, this user is free to select the most appropriate (meta) clustering. Again, it remains unclear where the differences between the clusterings exactly are. This applies to the clusterings on the base and meta levels. The user also needs to determine the differences independently. Consensus clustering, e. g., cluster ensemble methods (Strehl and Ghosh, 2003), typically tries to find the maximum agreement between multiple clusterings in order to generate one single clustering that is supposed to be better than the individual ones, i. e., such methods focus on how much information is shared. This approach also does not detect the exact clustering differences on the lowest level. A detailed differentiation of clusterings from a user's perspective is not possible with this approach. We consider a different path by providing a formal model description of the plain difference between clusterings. We see this description as the foundation for communicating changes to a user. To the best of our knowledge, there is no prior work following this specific idea.

# 3 PARTITIONAL CLUSTERING REPRESENTATION

In order to be able to differentiate partitional clusterings, we need to define a partitional clustering representation first. For this purpose, we have the following requirements:

1. *Completeness:* The partitional clustering representation shall be as complete and comprehensive as possible so that the difference between two partitional clusterings can also be determined in detail as much as possible. This requirement is about not losing any information that might later be useful to the user.
2. *Flexibility:* The partitional clustering representation should be separated from the partitional clustering algorithm as much as possible so that it is beneficial for a wider range of clustering applications, i. e., the partitional clustering representation needs to be flexible to some extent.

First, we consider a matrix $D \in \mathbb{R}^{N\times M}$ that conforms to the complete parent data set available for learning the partitional clustering. The $N$ rows and $M$ columns of D represent the data instances and data features respectively. $D_{i,*}$ denotes the $i$-th data instance of D, $1 \le i \le N$, and $D_{*,j}$ denotes the $j$-th data feature of D, $1 \le j \le M$. Consequently, the row and column indices of D act as the data instance and data feature identifiers. We assume that there is exactly one parent data set. Furthermore, we allow the selection of a $n \times m$ submatrix X of D for learning the partitional clustering so that only specific data instances and data features can be used, $n \le N, m \le M$. Then we finally consider the partitional clustering representation $(r, c, C, W, Y, p)$:

- A vector $r \in \mathbb{N}^n$ representing the selected data instance identifiers. It describes the mapping from data instance identifiers to row indices of X. The $i$-th vector entry $r_i$ is the data instance identifier of $X_{i,*}$, $1 \le i \le n$. The row index $i$ of X does not necessarily have to match the data instance identifier. For example, $X_{1,*}$ could actually represent the eleventh data instance of the parent data set D, i. e. $X_{1,*} = D_{11,*}$, instead of the first one, i. e. $X_{1,*} \ne D_{1,*}$.
- A vector $c \in \mathbb{N}^m$ representing the selected data feature identifiers. It acts as the mapping from data feature identifiers to column indices of X. The $j$-th vector entry $c_j$ is the data feature identifier of $X_{*,j}$, $1 \le j \le m$. The column indices of X do not necessarily have to match the data feature identifiers.
- A matrix $C \in \{-1, 0, 1\}^{n\times n}$ conforming to the pairwise constraints between the data instances. The entry $C_{i,j}$ denotes a pairwise constraint between the $i$-th data instance and the $j$-th data instance, $1 \le i \le n, 1 \le j \le n$. We consider three distinct values for the entries: $C_{i,j} = -1$ indicates a cannot-link constraint, $C_{i,j} = 1$ indicates a must-link constraint, and there is no pairwise constraint between $i$ and $j$ when $C_{i,j} = 0$.
- A matrix $W \in \{w \in \mathbb{R} \mid w \ge 0\}^{n\times n}$ denoting the weights of the pairwise constraints. The entry $W_{i,j}$ controls the importance of the related pairwise constraint $C_{i,j}$, $1 \le i \le n, 1 \le j \le n$. For example, the larger the entry, the greater the importance.
- A matrix $Y \in \{y \in \mathbb{R} \mid 0 \le y \le 1\}^{n\times k}$ corresponding to the cluster assignments of all data instances. The number of clusters is denoted by $k \in \mathbb{N}$. The entry $Y_{i,j}$ represents the cluster assignment of the

data instance $i$ to the cluster $j$, $1 \leq i \leq n, 1 \leq j \leq k$. It holds $\sum_{j=1}^{k} \mathrm{Y}_{i,j} = 1, \forall i$. The column indices of Y act as the cluster identifiers.

- A tuple $p$ containing the partitional clustering parameters. The number and structure of the tuple entries depend on the applied clustering algorithm. For example, $p$ could contain the mean vectors and covariance matrices of multivariate Gaussian mixture components.

We categorize the aforementioned components into three groups: $\mathrm{r}, \mathrm{c}, \mathrm{C}$, and W belong to the *input* parameters, Y belongs to the *output* parameters, and $p$ belongs to the *model* parameters. These groups provide a complete and comprehensive representation of the clustering (requirement 1).

We consider our partitional clustering representation as a general and flexible union of different ways to formalize unsupervised and semi-supervised partitional clustering representations (requirement 2). First, if the semi-supervised learning scenario is not of interest, we can ignore the related components C and W. This depends on the learning scenario and the available expert knowledge or supervision. Second, the chosen matrix structure of Y allows us to consider soft clustering, i. e., $\mathrm{Y}_{i,j} \in [0,1]$, as well as hard clustering, i. e., $\mathrm{Y}_{i,j} \in \{0,1\}$. Third, the model parameters $p$ further increase the degree of flexibility. Theorem 3.1 describes the (storage) space complexity of the partitional clustering representation.

**Theorem 3.1.** *Let $pc = (\mathrm{r}, \mathrm{c}, \mathrm{C}, \mathrm{W}, \mathrm{Y}, p)$ be a partitional clustering according to the definition of the partitional clustering representation, let $f_{pc}$ and $f_p$ denote the (storage) space of pc and p respectively, and let $O(\cdot)$ denote asymptotic notation. Then $f_{pc} = O(\max(n^2, f_p))$ holds true.*

*Proof.* Since $pc$ is composed of $\mathrm{r}, \mathrm{c}, \mathrm{C}, \mathrm{W}, \mathrm{Y}$, and $p$, the (storage) space $f_{pc}$ equals the sum $f_{\mathrm{r}} + f_{\mathrm{c}} + f_{\mathrm{C}} + f_{\mathrm{W}} + f_{\mathrm{Y}} + f_p$ of the individual (storage) spaces. So we have $f_{pc} = n + m + nn + nn + nk + f_p$. Considering that $n \gg k$ in clustering tasks, we finally have $f_{pc} = O(\max(n^2, f_p))$. □

We emphasize that we consider exactly one parent data set for a comparison of different partitional clusterings. If we were to consider multiple parent data sets from different instance and feature spaces instead, the differentiation of the related partitional clusterings would be pointless. This means that a change to D will be reflected to all involved partitional clusterings. And finally, please note that possibly a lot of entries of C and W are zero. Hence, C and W can be represented as sparse matrices instead. Going further, the redundant constraints that follow the symmetric property $\mathrm{C}_{i,j} = \mathrm{C}_{j,i}$ can also be removed, i. e., C and W could be represented as upper or lower triangular matrices.

# 4 PARTITIONAL CLUSTERING DIFFERENCE MODEL

We now propose the novel partitional clustering difference model. It is represented by the tuple $(r', c', C', W', Y', p')$. This tuple representation is similar to the partitional clustering representation introduced in the previous section. The components $r', c', C', W', Y'$, and $p'$ semantically relate to their counterparts $\mathrm{r}, \mathrm{c}, \mathrm{C}, \mathrm{W}, \mathrm{Y}$, and $p$ of the partitional clustering representation. However, each component now describes a difference, i. e., given two partitional clusterings $pc_1 = (\mathrm{r}_1, \mathrm{c}_1, \mathrm{C}_1, \mathrm{W}_1, \mathrm{Y}_1, p_1)$ and $pc_2 = (\mathrm{r}_2, \mathrm{c}_2, \mathrm{C}_2, \mathrm{W}_2, \mathrm{Y}_2, p_2)$, $r'$ describes the difference between $\mathrm{r}_1$ and $\mathrm{r}_2$, and $c'$ describes the difference between $\mathrm{c}_1$ and $\mathrm{c}_2$ etc. These component-wise differences on the lowest level lead to a full description of the overall difference. This approach allows us to detect changes to the input parameters, the output parameters, and the model parameters.

Each component of the partitional clustering difference model is represented by the ordered triple $(\mathrm{cm}, \mathrm{add}, \mathrm{rem})$. It entails the following three vectors: common entries or modified entries cm, added entries add, and removed entries rem. The vector add represents the entries that have been added to the involved component of $pc_2$, and the vector rem represents the entries that have been removed from the involved component of $pc_1$. In contrast, there are subtle differences in the interpretation of cm. Referring to $r'$, the vector cm represents the common data instance identifiers existing in both $\mathrm{r}_1$ and $\mathrm{r}_2$. The same applies to $c'$ and the common data feature identifiers. But when referring to $C'$ and $W'$, the vector cm represents the exact values of the differences between the modified constraints and weights respectively. The same applies to $Y'$, i. e., the vector cm represents the exact values of the differences between the modified cluster assignments of the data instances. In summary, by considering a component difference as the ordered triple $(\mathrm{cm}, \mathrm{add}, \mathrm{rem})$, we can carefully find out which exact difference or type of change exists. On the contrary, this approach may already seem complex or rather laborious because, for only one partitional clustering difference, we already have to deal with six ordered triples (one ordered triple for each partitional clustering difference component), and each ordered triple entry is represented by a vector which further increases the complexity. However, in this regard, we just aim for a complete, comprehensive, and fundamental model of partitional clustering differences at first. It is not about the direct or immediate interpretation of the partitional clustering difference model from a user's perspective.

We now briefly describe the computation of the individual component differences. Following our definition of the partitional clustering representation, we need to consider differences between vectors (r and c), matrices (C, W, and Y), and tuples ($p$). The computation of each component difference results in an ordered triple of the form $(\mathrm{cm}, \mathrm{add}, \mathrm{rem})$ as we described before. Referring to the difference $r'$ between the two vectors $\mathrm{r}_1$ and $\mathrm{r}_2$ of two partitional clusterings, cm then contains the entries that $\mathrm{r}_1$ and $\mathrm{r}_2$ have in common, add holds the entries that are contained in $\mathrm{r}_2$ but not in $\mathrm{r}_1$, and rem holds the entries that are contained in $\mathrm{r}_1$ but not in $\mathrm{r}_2$. The same applies to the computation of $c'$. The computation of the matrix differences $C', W'$, and $Y'$ is more sophisticated because it has to take the data instance identifiers, data feature identifiers, and cluster identifiers into account. We provide a novel, detailed algorithm for the computation of these special kind of matrix differences in figure 1.

This algorithm returns the ordered triple $(\mathrm{cm}, \mathrm{add}, \mathrm{rem})$ that represents the difference between two matrices $\mathrm{M}_1$ and $\mathrm{M}_2$. Finally, considering the tuple difference $p'$, the differences are computed for each component of the tuple analogously. We assume that the type and semantics of the model parameters $p$ are fixed.

We want to point out that it might be beneficial to consider a sparse vector representation for the vector cm that entails either common or modified values, especially if there are only subtle differences between the involved partitional clusterings. This would have a positive effect on the (storage) space of the partitional clustering difference model. The (storage) space complexity is described by Theorem 4.1. Additionally, if there is no difference between the components of the partitional clusterings at all, the vectors cm, add, and rem are empty vectors which we denote by $\varnothing$.

**Theorem 4.1.** *Let* $pcd = d(pc_1, pc_2) = (r', c', C', W', Y', p')$ *be the partitional clustering difference between the partitional clusterings* $pc_1 = (\mathrm{r}_1, \mathrm{c}_1, \mathrm{C}_1, \mathrm{W}_1, \mathrm{Y}_1, p_1)$ *and* $pc_2 = (\mathrm{r}_2, \mathrm{c}_2, \mathrm{C}_2, \mathrm{W}_2, \mathrm{Y}_2, p_2)$ *according to the definition of the partitional clustering difference model, let* $f_{pcd}$ *and* $f_{p'}$ *denote the (storage) space of pcd and* $p'$ *respectively, and let* $O(\cdot)$ *denote asymptotic notation. Then it follows* $f_{pcd} = O(\max({n_1}^2 + {n_2}^2, f_{p'}))$.

*Proof.* Since $pcd$ is composed of $r', c', C', W', Y'$, and $p'$, the (storage) space $f_{pcd}$ equals the sum $f_{r'} + f_{c'} + f_{C'} + f_{W'} + f_{Y'} + f_{p'}$ of the individual (storage) spaces. We have $f_{r'} = n_1 + n_2 - |\phi(\mathrm{r}_1) \cap \phi(\mathrm{r}_2)|, f_{c'} = m_1 + m_2 - |\phi(\mathrm{r}_1) \cap \phi(\mathrm{r}_2)|, f_{C'} = f_{W'} = n_1 n_1 + n_2 n_2 - |\phi(\mathrm{r}_1) \cap \phi(\mathrm{r}_2)|, f_{Y'} = n_1 k_1 + n_2 k_2 - |\phi(\mathrm{r}_1) \cap \phi(\mathrm{r}_2)|$, $\phi(\mathrm{v})$ returns a set containing the unique elements of the vector v. Considering that $n_1 \gg k_1$ and $n_2 \gg k_2$ in clustering tasks, this leads to $f_{pcd} = O(\max({n_1}^2 + {n_2}^2, f_p))$. □

Algorithm: Difference between two matrices of different sizes
Input:
1) Matrices $\mathrm{M}_1 \in \mathbb{R}^{s \times t}$ and $\mathrm{M}_2 \in \mathbb{R}^{u \times v}$
2) Vectors $\mathrm{r}_1 \in \mathbb{N}^s, \mathrm{c}_1 \in \mathbb{N}^t, \mathrm{r}_2 \in \mathbb{N}^u$, and $\mathrm{c}_2 \in \mathbb{N}^v$ that represent the row and column identifiers of $\mathrm{M}_1$ and $\mathrm{M}_2$

Output: Matrix difference $M'$ between $\mathrm{M}_1$ and $\mathrm{M}_2$
Remark:
1) $\mathrm{set}(\mathrm{v})$ returns a set with the elements of vector v
2) $\mathrm{replace}(\mathrm{a}, \mathrm{b})$ replaces all $a_i$ of vector a with $a_i$'s index in vector b
3) $\mathrm{vec}(\mathrm{M})$ returns the matrix M as a vector in row-major order
4) $\mathrm{M}[\mathrm{i};\mathrm{j}]$ denotes the submatrix of the matrix M formed from the $r$ rows given by the row indices vector $\mathrm{i} = [i_1, i_2, \ldots, i_r]$ and the $c$ columns given by the column indices vector $\mathrm{j} = [j_1, j_2, \ldots, j_c]$
5) $\mathrm{sparse}(\mathrm{v})$ returns a sparse vector representation of the (dense) vector v

Method:
1) Compute common row and column identifiers
 1.1) Row identifiers $\mathrm{r} := \mathrm{set}(\mathrm{r}_1) \cap \mathrm{set}(\mathrm{r}_2)$
 1.2) Column identifiers $\mathrm{c} := \mathrm{set}(\mathrm{c}_1) \cap \mathrm{set}(\mathrm{c}_2)$
2) Compute common row and column indices
 2.1) Row indices $\mathrm{i}_1$ of $\mathrm{M}_1 := \mathrm{replace}(\mathrm{r}, \mathrm{r}_1)$
 2.2) Column indices $\mathrm{j}_1$ of $\mathrm{M}_1 := \mathrm{replace}(\mathrm{c}, \mathrm{c}_1)$
 2.3) Row indices $\mathrm{i}_2$ of $\mathrm{M}_2 := \mathrm{replace}(\mathrm{r}, \mathrm{r}_2)$
 2.4) Column indices $\mathrm{j}_2$ of $\mathrm{M}_2 := \mathrm{replace}(\mathrm{c}, \mathrm{c}_2)$
3) Compute modified, added, and removed entries
 3.1) $\mathrm{mod} := \mathrm{sparse}(\mathrm{vec}(\mathrm{M}_2[\mathrm{i}_2;\mathrm{j}_2] - \mathrm{M}_1[\mathrm{i}_1;\mathrm{j}_1]))$ (use of sparse is optional)
 3.2) $\mathrm{add} := \mathrm{vec}($entries of $\mathrm{M}_2$ excluding $\mathrm{M}_2[\mathrm{i}_2;\mathrm{j}_2])$
 3.3) $\mathrm{rem} := \mathrm{vec}($entries of $\mathrm{M}_1$ excluding $\mathrm{M}_1[\mathrm{i}_1;\mathrm{j}_1])$
4) Return $M' = (\mathrm{mod}, \mathrm{add}, \mathrm{rem})$

Figure 1: Matrix difference algorithm for computing $C', W'$, and $Y'$ of the partitional clustering difference model.

We provide a detailed example for determining a partitional clustering difference between two partitional clusterings in Example 4.1 to better illustrate the previous descriptions.

**Example 4.1.** *Let* D *denote the parent data set for learning partitional clusterings as stated in* (1).

$$\mathrm{D} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 3 \\ 2 & 2 & 4 \end{bmatrix} \quad (1)$$

*Furthermore, consider the two specific partitional clusterings $pc_1$ and $pc_2$ that used* D *as stated in* (2) *and* (3) *respectively. Figure 2 depicts $pc_1$ and $pc_2$.*

$$pc_1 = ([1,2,3],[1,2],0_{3\times 3},0_{3\times 3},\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},(2)) \quad (2)$$

$$pc_2 = ([1,2,4],[1,2],0_{3\times 3},0_{3\times 3},\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},(2)) \quad (3)$$

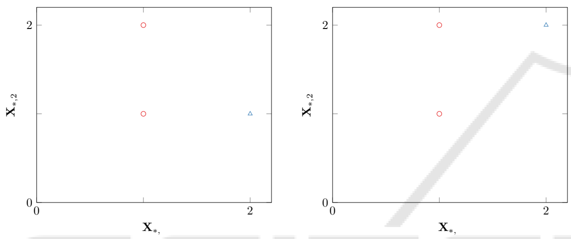

Figure 2: Two exemplary partitional clusterings $pc_1$ (left) and $pc_2$ (right) with two clusters each (indicated by marker shape and color).

*Both partitional clusterings have only one model parameter $k = 2$ that explicitly specifies the number of clusters. Then the difference of $pc_1$ and $pc_2$ yields the partitional clustering difference $pcd = d(pc_1, pc_2) = (r', c', C', W', Y', p')$. The difference of the data instance identifiers $r'$ is stated in* (4)*, and the difference of the data feature identifiers $c'$ is stated in* (5).

$$r' = d(\mathrm{r}_1, \mathrm{r}_2) = ([1,2],[4],[3]) \quad (4)$$

$$c' = d(\mathrm{c}_1, \mathrm{c}_2) = ([1,2],\varnothing,\varnothing) \quad (5)$$

*These equations demonstrate that the data instance with identifier 4 has been used to learn $pc_2$ (but not $pc_1$), the data instance with identifier 3 has been used to learn $pc_1$ (but not $pc_2$), the first two data instances of* D *have been used to learn both $pc_1$ and $pc_2$, and that both partitional clusterings have been learned using the first two data features only. The difference $C'$ is then stated in* (6).

$$\begin{aligned} C' &= d(\mathrm{C}_1, \mathrm{C}_2, \mathrm{r}_1, \mathrm{c}_1, \mathrm{r}_2, \mathrm{c}_2) \\ &= (0,0,0,0],[0,0,0,0,0],[0,0,0,0,0]) \end{aligned} \quad (6)$$

*No values have been modified, but there are added and removed values because of the difference $r'$. The difference $W'$ is computed analogously. Finally, the differences $Y'$ and $p'$ are stated in* (7) *and* (8) *respectively.*

$$Y' = d(\mathrm{Y}_1, \mathrm{Y}_2, \mathrm{r}_1, \mathrm{c}_1, \mathrm{r}_2, \mathrm{c}_2) = ([0,0,0,0],[0,1],[0,1]) \quad (7)$$

$$p' = d(p_1, p_2) = (0) \quad (8)$$

We explicitly point out that the computation of the partitional clustering difference model will differentiate clusterings even if only the cluster identifiers have been swapped between partitional clusterings. In this case, all clusters could still contain the same data instances as before. Although this is formally a difference, such a difference might not be of interest. Then this could be corrected by incorporating model parameters like the cluster centers to check for the equality of the clusters, or a mapping from cluster identifiers to column indices of the cluster assignments matrix Y could be added to the partitional clustering representation as we did with the data instance and data feature identifiers. But we will not focus on this specific aspect in this paper any further.

## 5 USE CASES

In this section, we apply the proposed partitional clustering difference model. By this means, we demonstrate the usefulness and potential of this novel model. We also demonstrate its benefits in comparison to the partitional clustering representation. Overall, our objective is to motivate for the necessity of the partitional clustering difference model in specific situations. At the same time, we emphasize that this is fundamental work conducted more on a conceptual level.

We concentrate on exemplary but still prominent real-world use cases in the area of planning and decision processes (Pahl-Weber and Henckel, 2008; Blotevogel et al., 2014). These processes play a crucial role in the e-participation domain where people are allowed to voice their opinions and ideas in different areas such as landscape planning or city budgeting (Briassoulis, 1997). Overall, such planning and decision processes can last several days, weeks, months, or even years. During that time, special phases exist where people are allowed to participate. In the end, participants write and submit contributions that should be assessed by public administrations. The public administration workers need to make decisions, e. g., they aggregate ideas for a new building project, or they accept or reject general complaints.

We consider the following scenario for the use cases: A public administration worker needs to analyze a data set of 1590 contributions submitted by citizens. These contributions consist of textual data (content of the contribution), time-oriented data (creation time), and spatial data (longitude and latitude representing the contribution's reference point). Table 1 shows an exemplary contribution. The data set originates from a real past participation phase of a

Table 1: An exemplary contribution.

| Content | Timestamp | Longitude | Latitude |
|---|---|---|---|
| The cobblestones are in a very poor condition. They cause a high level of noise pollution. Even the current speed limit does not help here, especially since this is ignored by many drivers. | 2021-05-23T09:25 | 13.4577 | 52.5128 |

planning and decision process. The contributions report city noise sources. The city intends to take action against the most common noise sources. Therefore, the public administration worker's objective is to find partitions of similar contributions. Generally, the motivation behind this is that similar contributions can be assessed and dealt with in a fairly similar way which would reduce the amount of work for the public administration worker. Such a partitioning helps to aggregate the different topics or complaints submitted by the citizens. The public administration worker is assisted by a machine learning system that is able to cluster the contributions by incorporating the *k*-means algorithm and the instance-based pairwise-constrained *k*-means algorithm (only in the fourth use case). The public administration worker could cluster the contributions one by one and compare the own results to the clustering proposed by the machine learning system, or the public administration worker could explore different clusterings by experimenting with parameters of the machine learning system. We point out that we consider the *k*-means algorithm for demonstration purposes only. We could have used other partitional clustering algorithms such as *k*-medoids or *k*-medians due to our flexible partitional clustering representation.

The proposed partitional clustering difference model can be used in various situations and for different reasons when clustering the contributions. We focus on the following specific use cases: (1) Debug the clustering algorithm, (2) Change the number of clusters, (3) Detect changes in the data set, and (4) Create constraints. These reflect common interactions (Bae et al., 2020). The machine learning system would compute the difference between clusterings.

## 5.1 Debug the Clustering Algorithm

This use case is about the traceability of each iteration of the clustering algorithm, e. g., when the initial clustering of the contributions is learned, or every time the public administration worker wants the machine learning system to re-compute the clustering. Sometimes, such a profound understanding is necessary, especially if the computed clustering is adopted by the public administration worker (with or without further user-made adjustments) in order to make major and possibly impactful decisions as we mentioned in the introduction of this paper. Additionally, public administration workers are laypersons in the field of machine learning. At least a brief understanding of how the machine learning system works in practice can be useful when it comes to trusting and accepting the computed clusterings.

In this use case, the public administration worker can examine the clustering at each iteration of the learning process which refers to analyzing a sequence of partitional clusterings. Our proposed partitional clustering representation can be used for this purpose to gain comprehensive and complete insights. We acknowledge that this procedure is generally not a novel approach. In fact, such a method is already used for teaching or educational purposes at least. But nevertheless, there is still a downside that we emphasize: the public administration worker would have to compare the partitional clusterings on her own. Then, for example, it might be difficult to grasp the exact differences between the initial and final clusterings. On the contrary, the partitional clustering difference model allows a new perspective. The public administration worker can apply the partitional clustering difference model to find out how the clustering of the contributions changes either step by step or by examining the difference between non-sequential partitional clusterings.

Figure 3 depicts partitional clusterings of the first, second last, and last iterations of the clustering algorithm[1] and the partitional clustering differences between them. While the sequence of partitional clusterings allows a general overview of the clustering for every iteration, the sequence of partitional clustering differences explicitly shows how many contributions

[1] We used the textual content of the contributions. First, we removed non-alphanumeric characters from the textual content. Then we tokenized the textual content, converted it to lower case, and, finally, we stemmed the results using the Porter stemmer algorithm. Second, we used the resulting tokens to create a term-document matrix with term frequency–inverse document frequency (TF–IDF) weights. Third, we performed latent semantic indexing on this matrix to derive ten concepts. We randomly picked three initial cluster centers from the data set of contributions.

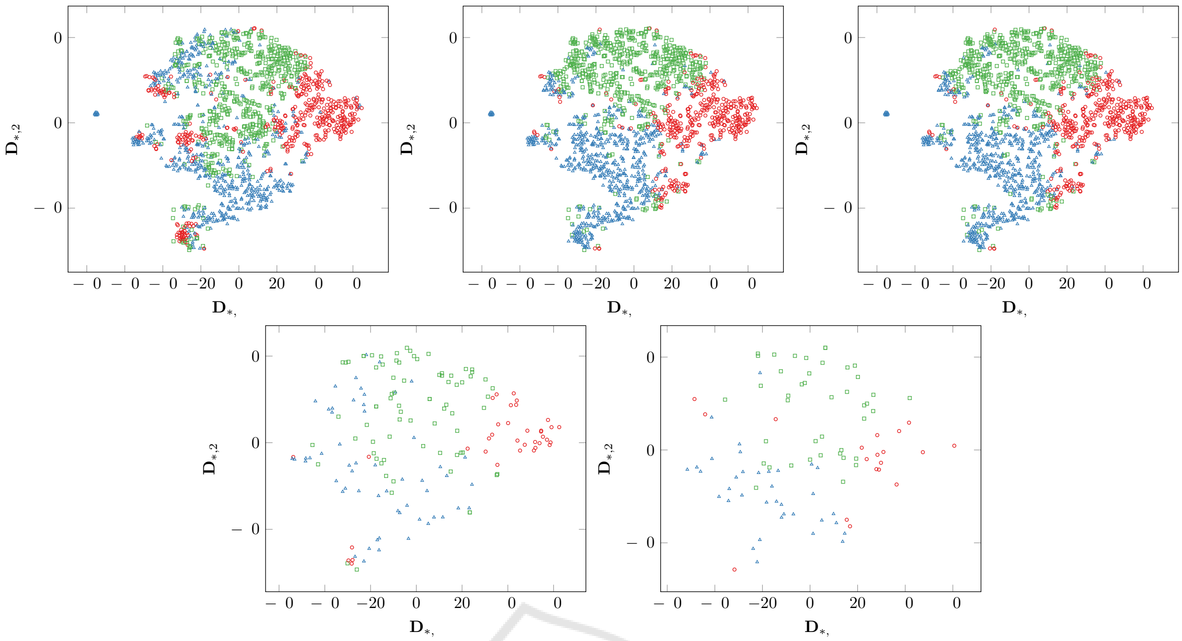

Figure 3: Partitional clusterings (top, from left to right) and related partitional clustering differences (bottom, from left to right) between these partitional clusterings with three clusters (indicated by marker shape and color). The term-frequency–inverse document frequency (TF–IDF) vector representations of the contributions' contents used for clustering have been reduced to two dimensions using a t-distributed stochastic neighbor embedding (t-SNE) for demonstration purposes only.

are assigned to different clusters in comparison to the previous iteration. Of course, it is still difficult to specify the exact quantity of affected contributions by only relying on this specific visualization. This clearly depends on the overall number of contributions. But from iteration to iteration, there should be fewer differences visible according to how the clustering algorithm works. Figure 3 confirms this. Thus, the public administration worker can get more insight into how the clustering algorithm generally works by applying the partitional clustering difference model. Please note again that we provide only sample visualizations for demonstration purposes. The visual encoding of the partitional clustering difference model is not the focus of this paper. However, this does not change the fact that this model contains all the information needed to communicate the exact differences between each iteration. The model could also be the foundation for deriving further metrics such as the exact quantity of changes.

## 5.2 Change the Number of Clusters

During the clustering task, the public administration worker might increase or decrease the number of clusters in order to compare appropriate clusterings of the contributions. This can be seen as an optimization step of a clustering model parameter from an expert's perspective. But this can also be seen as some kind of experimentation with algorithm or user interface settings from a layperson's perspective. The public administration worker might alternately increment or decrement the number of clusters just to get an idea of the effects on the clustering result computed by the machine learning system. Either way, this specific user interaction will most likely affect the cluster assignments of some contributions, i. e., some contributions keep their previous cluster assignments, and others get new cluster assignments. It is important for the public administration worker to notice these differences, especially when the public administration worker tries to build a mental model of the underlying concept that represents the clustering. But it can be challenging to first grasp and then evaluate this concept if the clustering changes without communicating the differences.

Figure 4 shows a small sample of 51 contributions for demonstration purposes only. It illustrates how the cluster assignments of these contributions differ from each other when the public administration worker decreases the number of clusters used by the clustering algorithm[2] from $k = 3$ to $k = 2$. The simple juxtaposi-

[2]Again, we focused on the textual content, and we applied the same preprocessing steps as in the first use case. We used $k$-means++ seeding for sequentially choosing the initial cluster centers.

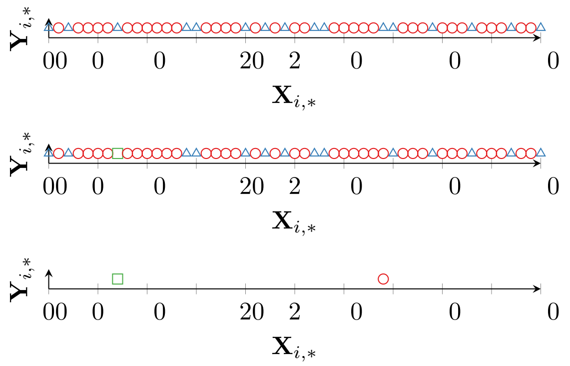


Figure 4: Two partitional clustering assignments (top and middle) of the contributions from 100 to 150 (from left to right) to a maximum of three clusters (differentiated by marker shape and color) and the difference (bottom).

tion of both partitional clustering assignments (before and after the change) together with the sorting of the contributions allows the public administration worker to search for differences. The public administration worker should be able to identify the differences by scanning through the whole list of contributions and their assignments. This is not a novel idea, and such a visual juxtaposition can be easily generated even without our introduced partitional clustering representation. But there are still issues left. The public administration worker can overlook differences, or the public administration worker may require more efforts to find these. This work is already laborious for 51 contributions. Generally, this depends at least on the number of contributions, the number of clusters (before and after the change), and the user's cognitive abilities. Then this is exactly where the partitional clustering difference model assists the public administration worker in finding and analyzing the differences between the cluster assignments more efficiently. For example, instead of communicating possibly large lists that contain the previous and current cluster assignments of the individual contributions to the public administration worker, only the specific contributions that actually changed their assignment can be presented to the public administration worker. Then the public administration worker does not have to search for these differences because they have already been detected computationally. A condensed programmatic output of the partitional clustering difference for this example is shown in Figuere 5. This output can be the foundation for a new visualization that communicates the differences.

```
PCD = (r', c', C', W', Y', p'):
 r' = ..., c' = ...,
 C' = ..., W' = ...,
 p' = ...,
 Y' = (mod, add, rem)
    = ([(107, 2->3), ..., (134, 2->1)],
       [], [])
```

Figure 5: Condensed programmatic output of the partitional clustering difference between the two clusterings shown in Figure 4. The output partially lists the affected contributions and their new cluster assignments, e. g., the contribution 107 now belongs to cluster 3.

## 5.3 Detect Changes in the Data Set

The contributions submitted to the public administration can change in planning and decision processes of the e-participation domain. There are various reasons for this circumstance. For example, the data set of contributions might not always be complete when the public administration worker starts the clustering process, which means that new contributions can arrive later. Public administrations sometimes start to analyze the contributions even though the participation phase is still running. This can happen when the data set of contributions is large and the assignments of parts of the contributions must be controlled manually by the public administration. So in order to cope with the data set volume, the public administration worker may want to start early with clustering the contributions available at this specific point in time. This problem becomes more prominent when the user-driven clustering process takes multiple hours or even days with possible breaks in between while new contributions can still be submitted by the participants. Another example is the deletion or editing of some contributions after the initial submission. This can be done by the owners or submitters of the contributions. For example, participants sometimes correct the location the contribution points to, assuming that such information is collected at all, or the participants sometimes edit the content after the initial submission. Such a change should be taken into account because the contribution could suddenly portray a completely different meaning or complaint. Furthermore, the machine learning system could learn a completely different clustering by taking the added, removed, and modified contributions into account. This new clustering and the reasons for the re-computation, i. e., the changes to the contributions, should then be communicated to the public administration worker. The public administration worker should be able to differentiate the proposed two clusterings before and after the changes to the data set of contributions.

Another perspective and motivation for the partitional clustering difference model is that the changes to the contributions cannot be controlled by the public administration worker who wants to cluster the data set. This missing control means that some kind of detection and notification could be useful to inform the public administration worker about the change or difference in the data set because it might affect the overall clustering result when known contributions are suddenly missing, have been changed, or when new contributions reveal new relationships or ideas. While the public administration worker could possibly just ignore deleted contributions in the current clustering, added contributions must still be assessed and put into the correct cluster by the public administration worker.

Figure 6 illustrates a sample of 100 contributions from our real-world data set at two different points in time during the participation phase. It immediately becomes clear that it is challenging to identify all differences. This is especially true when there are no further hints to suggest what to look for. This problem is not restricted to this exemplary visualization that focuses on the spatial data of the contributions. We could also arrange the contributions side by side while focusing on the textual content (before and after the changes), and the identification of differences would probably be even more challenging. But by using the partitional clustering difference model instead, the public administration worker can easily identify the differences between the data sets because it tracks the exact changes to the data instances and data features used to actually learn a partitional clustering. In this use case, it detects added, removed, and modified contributions. So overall, the number of contributions changed because there are more additions of contributions than removals. One edit occurred. Again, the public administration worker can retrieve this data set difference by analyzing the partitional clustering difference. Such a simple quantification can still be done without our model. However, the public administration worker is also able to retrieve which exact contributions changed, have been removed, or are completely new.

## 5.4 Create Constraints

The automatically computed assignments of contributions to clusters are not always correct so that the public administration worker wants to integrate some corrections. In this use case, the public administration worker adds a few pairwise constraints to make some corrections to the partitional clustering computed by the clustering algorithm, i. e., some contributions shall belong to the same cluster if possible because they represent a similar concern that the clustering algorithm did not detect. Based on this new information, the machine learning system can re-compute the clustering with a potentially better quality. In turn, this new clustering can be presented to the public administration worker. But the addition of pairwise constraints not only influences the clustering assignments. The clustering algorithm, in this use case the instance-based pairwise-constrained $k$-means algorithm, also has to re-compute the transitive closure of the pairwise constraints every time a pairwise constraint is added, i. e., the clustering algorithm derives new pairwise constraints, or it needs to remove existing ones. This is also true when the public administration worker deletes some pairwise constraints between the contributions. Even if the public administration worker explicitly adds only one pairwise constraint, other pairwise constraints can also be affected. It is possible that multiple new must-link constraints are added implicitly, although the public administration worker has explicitly added only one must-link constraint. This is not only a difference triggered by the public administration worker but also a difference triggered by the clustering algorithm. Such a distinction can be important. These effects might be clear to an expert user with background knowledge in machine learning. But a layperson like the public administration worker might not know about the properties of pairwise constraints. However, the effects on the relations between contributions should be made clear, especially if it can affect the clustering result. The public administration worker should investigate implicitly added constraints in order to improve the understanding of the relations between the involved contributions. Furthermore, the larger the number of contributions the more challenging it is for the public administration worker to manually keep track of the implicitly changed pairwise constraints.

Table 2 lists five must-link constraints that the public administration worker added explicitly. Based on these, three more must-link constraints have been added implicitly by the clustering algorithm. The partitional clustering difference model can keep track of these differences. The public administration worker can then inspect the new proposed relations between the contributions by considering the implicitly as well as the explicitly added pairwise constraints.

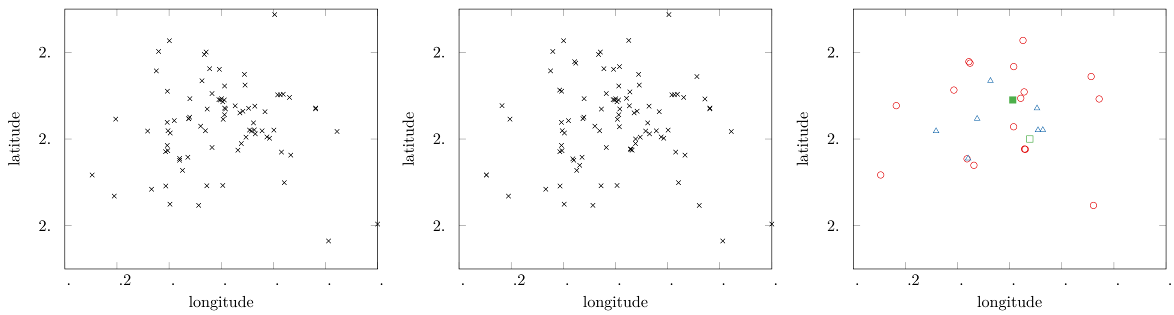


Figure 6: The data set of contributions with a sample of 100 contributions at two different points in time during the participation phase (left and middle), and the difference between these data sets (right). It shows added contributions (red, circular marker), removed contributions (blue, triangular marker), and one modified contribution (green, rectangular marker; the filled marker style represents the contribution before the modification).

Table 2: Explicitly and implicitly added must-link constraints between the contributions $x_i$ and $x_j$.

| | Constraints | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Explicit | | | | | Implicit | | |
| $x_i$ | 1 | 1 | 34 | 65 | 98 | 1 | 2 | 2 |
| $x_j$ | 2 | 34 | 66 | 67 | 99 | 66 | 34 | 66 |

## 6 CONCLUSION

Our work focused on revealing the differences between partitional clusterings. We introduced a novel partitional clustering difference model for the differentiation of two partitional clusterings. It is equally suitable for unsupervised and semi-supervised clustering because it can store information about differences between pairwise constraints that typically represent some form of supervision. In general, this partitional clustering difference model keeps track of all changes to the input, output and model parameters of the involved partitional clusterings. Consequently, it does not only track differences between clustering assignments but also between input parameters like the data instances and data features used to learn the partitional clusterings. The exact clustering differences become transparent.

The partitional clustering difference model is valuable for clustering comparison tasks. A user cannot always be sure that no clustering differences actually exist just because none were found by the user. The novel partitional clustering difference model detects all differences without error and with no human efforts instead. We demonstrated the potential of the partitional clustering difference model by applying it to different prominent real-world use cases in the e-participation domain. Nonetheless, future work is still needed.

## 7 FUTURE WORK

There is significant potential for future work due to the novelty of the proposed partitional clustering difference model and its potential. The related ideas involve different research areas. First, it should be investigated how the partitional clustering difference model can be efficiently communicated to a user. We provided some visualizations in this paper but for demonstration purposes only. We need to decide which information should be shown to and which information should be hidden from the user. Thus, the research and application of proper visualization techniques for the differentiation of partitional clusterings is an important part of possible future work. In this regard, general visualization techniques like juxtaposition, explicit encoding, and superposition (Gleicher et al., 2011) should be investigated further. Especially ways to visually encode at least parts of a partitional clustering difference should be studied and developed. Second, we would like to examine how the model can be combined with the standard evaluation measures mentioned in Section 2. This concerns both the application of existing standard evaluation measures and the formulation of new measures in order to express the magnitude of the differences. Third, we would like to conduct user studies with laypersons to evaluate the appropriateness of the partitional clustering difference model at least in the described use cases. For this purpose, intelligent user interfaces need to be researched and developed that integrate the partitional clustering difference model in the clustering process. Overall, this is an interdisciplinary topic.

## REFERENCES


Abdul, A., Vermeulen, J., Wang, D., Lim, B. Y., and Kankanhalli, M. (2018). Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 582:1–582:18, New York, NY, USA. ACM.

Bae, J., Helldin, T., Riveiro, M., Nowaczyk, S., Bouguelia, M.-R., and Falkman, G. (2020). Interactive clustering: A comprehensive review. *ACM Computing Surveys*, 53(1):1:1–1:39.

Basu, S., Banerjee, A., and Mooney, R. J. (2004). Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 333–344, Lake Buena Vista, Florida, USA. Society for Industrial and Applied Mathematics.

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Advanced Applications in Pattern Recognition. Springer US, New York, NY, USA.

Bilenko, M., Basu, S., and Mooney, R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering. In *21st International Conference on Machine Learning*, page 11, Banff, Alberta, Canada. ACM Press.

Blotevogel, H. H., Danielzyk, R., and Münter, A. (2014). Spatial planning in germany. In *Spatial planning systems and practices in Europe*. Routledge Taylor & Francis Group, London, UK and New York, NY, USA.

Briassoulis, H. (1997). How the others plan: Exploring the shape and forms of informal planning. *Journal of Planning Education and Research*, 17(2):105–117.

Caruana, R., Elhawary, M., Nguyen, N., and Smith, C. (2006). Meta clustering. In *6th International Conference on Data Mining*, pages 107–118, Hong Kong, China. IEEE.

Coden, A., Danilevsky, M., Gruhl, D., Kato, L., and Nagarajan, M. (2017). A method to accelerate human in the loop clustering. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, Proceedings, pages 237–245, Houston, Texas, USA. Society for Industrial and Applied Mathematics.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning.

Gleicher, M., Albers, D., Walker, R., Jusufi, I., Hansen, C. D., and Roberts, J. C. (2011). Visual comparison for information visualization. *Information Visualization*, 10(4):289–309.

High-Level Expert Group on AI (2019). Ethics guidelines for trustworthy AI. Report, European Commission, Brussels.

Kaufman, L. and Rousseeuw, P. J. (1987). Clustering by means of medoids. In *Statistical Data Analysis Based on the L1-Norm and Related Methods*, pages 405–416. Elsevier Science, Amsterdam, North-Holland; New York, NY., USA.

Kim, B., Khanna, R., and Koyejo, O. O. (2016). Examples are not enough, learn to criticize! Criticism for interpretability. In *Advances in Neural Information Processing Systems 29*, pages 2280–2288. Curran Associates, Inc., Barcelona, Spain.

Kulesza, T., Burnett, M., Wong, W.-K., and Stumpf, S. (2015). Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pages 126–137, Atlanta, Georgia, USA. ACM.

Kulesza, T., Stumpf, S., Burnett, M., and Kwan, I. (2012). Tell me more? the effects of mental model soundness on personalizing an intelligent agent. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1–10, Austin, Texas, USA. ACM.

Kulesza, T., Stumpf, S., Burnett, M., Yang, S., Kwan, I., and Wong, W.-K. (2013). Too much, too little, or just right? Ways explanations impact end users' mental models. In *2013 IEEE Symposium on Visual Languages and Human Centric Computing*, pages 3–10, San Jose, CA, USA. IEEE.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, CA, USA. The Regents of the University of California.

Meilă, M. (2003). Comparing clusterings by the variation of information. In *Learning Theory and Kernel Machines*, Lecture Notes in Computer Science, pages 173–187, Berlin, Heidelberg. Springer.

Meilă, M. (2005). Comparing clusterings: An axiomatic view. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 577–584, Bonn, Germany. ACM Press.

Meilă, M. (2007). Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895.

Meilă, M. and Heckerman, D. (2001). An experimental comparison of model-based clustering methods. *Machine Learning*, 42(1):9–29.

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

Pahl-Weber, E. and Henckel, D., editors (2008). *The planning system and planning terms in Germany*. Academy for Spatial Research and Planning, Hanover, DE.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Schubert, E. and Rousseeuw, P. J. (2019). Faster k-medoids clustering: Improving the PAM, CLARA, and CLARANS algorithms. In *Similarity Search and*

*Applications*, Lecture Notes in Computer Science, pages 171–187, Cham. Springer International Publishing.

Simons, D. and Rensink, R. (2005). Change blindness: Past, present, and future. *Trends in cognitive sciences*, 9:16–20.

Strehl, A. and Ghosh, J. (2003). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617.

Vinh, N. X., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854.

Wagner, S. and Wagner, D. (2007). Comparing clusterings – An overview. Technical report, Karlsruhe.

Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.