Algorithm-Agnostic Feature Attributions for Clustering

Christian A. Scholbeck[1,2(✉)], Henri Funk[1,2], and Giuseppe Casalicchio[1,2]

[1] LMU Munich, Munich, Germany
[2] Munich Center for Machine Learning (MCML), Munich, Germany
{christian.scholbeck,henri.funk,
giuseppe.casalicchio}@stat.uni-muenchen.de

Abstract. Understanding how assignments of instances to clusters can be attributed to the features can be vital in many applications. However, research to provide such feature attributions has been limited. Clustering algorithms with built-in explanations are scarce. Common algorithm-agnostic approaches involve dimension reduction and subsequent visualization, which transforms the original features used to cluster the data; or training a supervised learning classifier on the found cluster labels, which adds additional and intractable complexity. We present FACT (feature attributions for clustering), an algorithm-agnostic framework that preserves the integrity of the data and does not introduce additional models. As the defining characteristic of FACT, we introduce a set of work stages: sampling, intervention, reassignment, and aggregation. Furthermore, we propose two novel FACT methods: SMART (scoring metric after permutation) measures changes in cluster assignments by custom scoring functions after permuting selected features; IDEA (isolated effect on assignment) indicates local and global changes in cluster assignments after making uniform changes to selected features.

Keywords: Interpretable clustering · explainable AI · feature attributions · algorithm-agnostic · effect · importance · FACT · SMART · IDEA

1 Introduction

Recent efforts have focused on making machine learning models interpretable, both via model-agnostic interpretation methods and novel interpretable model types [27], which is referred to as interpretable machine learning or explainable artificial intelligence in different contexts. Unfortunately, success in addressing cluster interpretability has been limited [3]. In the context of our paper, feature attributions (FAs) either provide information regarding the importance of features for assigning instances to clusters (overall and to specific clusters); or how isolated changes in feature values affect the assignment of single instances or

C. A. Scholbeck and H. Funk—Contributed equally.

L. Longo (Ed.): xAI 2023, CCIS 1901, pp. 217–240, 2023.
https://doi.org/10.1007/978-3-031-44064-9_13

the entire data set to each cluster. Interpretable clustering algorithms [3,23,31] provide some insight into the constitution of clusters, e.g., relationships between features within clusters, but often fall short of providing FAs. Furthermore, the range of interpretable clustering algorithms is limited. An alternative approach is to post-process the original data (e.g., via principal components analysis) and visualize the found clusters in a lower-dimensional space [17]. This obfuscates interpretations by transforming the original features used to cluster the data. A third option is to train a supervised learning (SL) classifier on the found cluster labels, which is interpreted instead. This adds additional and intractable complexity on top of the clustering by introducing an additional model.

Contributions: We present FACT[1] (feature attributions for clustering), a framework that is compatible with any clustering algorithm able to reassign instances to clusters (algorithm-agnostic), preserves the integrity of the data, and does not introduce additional models. As the defining characteristic of FACT, we propose four work stages: sampling, intervention, reassignment, and aggregation. Furthermore, we introduce two novel FACT methods: SMART (scoring metric after permutation) measures changes in cluster assignments by custom scoring functions after permuting selected features; IDEA (isolated effect on assignment) indicates local and global changes in cluster assignments after making uniform changes to selected features. FACT is inspired by principles of model-agnostic interpretation methods in SL, which detach the interpretation method from the model, thereby detaching the interpretation method from the clustering algorithm. In Fig. 1, we summarize how SMART and IDEA utilize select ideas from SL and how they innovate with new principles.

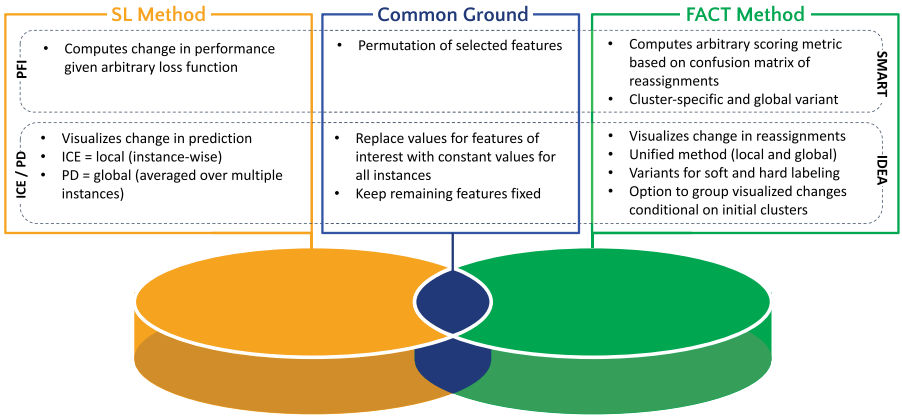


Fig. 1. Comparison of related concepts from SL (overlap in the center) with the clustering setting and novelties for FACT methods SMART and IDEA (right side).

[1] All presented methods are implemented in the R package `FACT` [13].

2 Notation and Preliminaries

2.1 Notation

We cluster a data set $\mathcal{D} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n}$ (where $\mathbf{x}^{(i)}$ denotes the i-th observation) into k clusters $\mathcal{D}^{(c)},\ c \in \{1, \ldots, k\}$. A single observation $\mathbf{x}$ consists of p feature values $\mathbf{x} = (x_1, \ldots, x_p)$. A subset of features is denoted by $S \subseteq \{1, \ldots, p\}$ with the complement set being denoted by $-S = \{1, \ldots, p\} \setminus S$. With slight abuse of notation, an observation $\mathbf{x}$ can be partitioned into $\mathbf{x} = (\mathbf{x}_S, \mathbf{x}_{-S})$, regardless of the order of elements within $\mathbf{x}_S$ and $\mathbf{x}_{-S}$. A data set $\mathcal{D}$ where all features in S have been shuffled jointly is denoted by $\tilde{\mathcal{D}}_S$. The initial clustering is encoded within a function f that - conditional on whether the clustering algorithm outputs hard or soft labels[2] - maps each observation $\mathbf{x}$ to a cluster c (hard label) or to k soft labels:

$$\text{Hard labeling: } f : \mathbf{x} \mapsto c,\ c \in \{1, \ldots, k\}$$
$$\text{Soft labeling: } f : \mathbf{x} \mapsto \mathbb{R}^k$$

For soft clustering algorithms, $f^{(c)}(\mathbf{x})$ denotes the soft label for the c-th cluster. This notation is also used to indicate the cluster-specific value within an IDEA vector (see Sect. 3.2).

2.2 Interpretations of Supervised Learning Models

In recent years, the interpretation of model output has become a popular research topic [28]. Existing techniques provide explanations in terms of FAs (e.g., a value indicating a feature's importance to the model or a curve indicating its effects on the prediction), model internals (e.g., beta coefficients for linear regression models), data points (e.g., counterfactual explanations [39]), or surrogate models (i.e., interpretable approximations to the original model) [27]. Many model-agnostic methods are based on identical work stages: First, a subset of observations is sampled which we intend to use for the model interpretation (sampling stage). This is followed by an intervention in feature values where the instances from the sampling stage are manipulated in certain ways (intervention stage). Next, we predict with the trained model and this new, artificial data set (prediction stage). This produces local (observation-wise) interpretations which can be further aggregated to produce global or semi-global interpretations (aggregation stage) [35]. These work stages can be considered a sensitivity analysis (SA) of the model.

[2] A vector of soft labels represents the propensity of an observation being assigned to each cluster. A convenient representation corresponds to a vector of pseudo probabilities $[0, 1]^k$. We refrain from labeling any algorithm as a hard or soft clustering algorithm because often an algorithm can output both hard and soft labels, e.g., k-means - traditionally considered a hard clustering algorithm - could output soft labels in the form of Euclidean distances to each cluster centroid.

Established methods to determine FAs for SL models comprise the individual conditional expectation (ICE) [16], partial dependence (PD) [11], accumulated local effects (ALE) [2], local interpretable model-agnostic explanations (LIME) [33], Shapley values [26,37], or the permutation feature importance (PFI) [6, 9]. The functional analysis of variance (FANOVA) [18,34] and Sobol indices [36] of a high-dimensional model representation are powerful tools to quantify input influence on the model output in terms of variance but are limited by the requirement for independent inputs. Among the mentioned techniques, the following three are useful for the development of SMART and IDEA:

- **PFI:** Shuffling a feature in the data set destroys the information it contains. The PFI evaluates the model performance before and after shuffling and uses the change in performance to describe a feature's importance.
- **ICE:** The ICE function indicates the prediction of an SL model for a single observation $\mathbf{x}$ where a subset of values $\mathbf{x}_S$ is replaced with values $\tilde{\mathbf{x}}_S$ while we condition on the remaining features $\mathbf{x}_{-S}$, i.e., keep them fixed. For single features of interest, an ICE corresponds to a single curve.
- **PD:** The PD function indicates the expected prediction given the marginal effect of a set of features. The PD can be estimated through a point-wise aggregation of ICEs across all considered instances.

2.3 Interpretations for Clustering Algorithms

Unsupervised clustering has largely been ignored by this line of research. However, for high-dimensional data sets, the clustering routine can often be considered a black box, as we may not be able to assess and visualize the multidimensional cluster patterns found by the algorithm. It is, therefore, desirable to receive deeper explanations of how an algorithm's decisions can be attributed to the features. Interpretable clustering algorithms incorporate the interpretability criterion directly into the cluster search. One option is to find an interpretable tree-based clustering [5,10,12,14,15,24,25,30]. Interpretable clustering of numerical and categorical objects (INCONCO) [31] is an information-theoretic approach based on finding clusters that minimize minimum description length. It finds simple rule descriptions of the clusters by assuming a multivariate normal distribution and taking advantage of its mathematical properties. Interpretable clustering via optimal trees (ICOT) [3] uses decision trees to optimize a cluster quality measure. In [23] clusters are explained by forming polytopes around them. Mixed integer optimization is used to jointly find clusters and define polytopes.

The focus of this paper lies on algorithm-agnostic interpretations. In many cases, we wish to use a clustering algorithm that does not provide any explanations. Furthermore, even interpretable clustering algorithms often do not directly provide FAs, thus still requiring additional interpretation methods. Analogously to SL, we may define post-hoc interpretations (which are typically algorithm-agnostic) as ones that are obtained after the clustering procedure, e.g., by showing a subset of representative elements of a cluster or via visualization techniques

such as scatter plots [22]. In most cases, the data is high-dimensional and requires the use of dimensionality reduction techniques such as principal component analysis (PCA) before being visualized in two or three dimensions. PCA creates linear combinations of the original features called the principal components (PCs). The goal is to select fewer PCs than original features while still explaining most of their variance. PCA obscures the information contained in the original features by rotating the system of coordinates. For instance, interpretable correlation clustering (ICC) [1] uses post-processing of correlation clusters. A correlation cluster groups the data such that there is a common within-cluster hyperplane of arbitrary dimensionality. ICC applies PCA to each correlation cluster's covariance matrix, thereby revealing linear patterns inside the cluster. One can also use an SL algorithm to post-process the clustering outcome which learns to find interpretable patterns between the found cluster labels and the features. Although we may use any SL algorithm, classification trees are a suitable choice due to naturally providing decision rules on how they arrive at a prediction [4]. Although this is a simple approach that can produce FAs via model internals or model-agnostic interpretation methods, it introduces intractable complexity through an additional model.

An algorithm-agnostic option that bypasses these issues is a form of SA where data are deliberately manipulated and reassigned to existing clusters. The global permutation percent change (G2PC) [8] indicates the percentage of change between the cluster assignments of the original data and those from a permuted data set. A high G2PC indicates an important feature for the clustering outcome. The local permutation percent change (L2PC) [8] uses the same principle for single instances.

3 FACT Framework and Methods

We first define a distinction of various FAs for the clustering setting: A *local FA* indicates how a feature contributes to the cluster assignment of a single observation; a *global FA* indicates how a feature contributes to the cluster assignments of an entire data set; a *cluster-specific FA* indicates how a feature contributes to the assignments of observations to one specific cluster. We introduce four work stages for FACT methods:

- **Sampling:** We sample a subset of observations that were previously clustered and shall be used to determine FAs. The larger this subset, the better our FA estimates. The smaller, the faster their computation.
- **Intervention:** Next, we manipulate feature values for the subset of observations from the sampling stage. This can be a targeted intervention (e.g., replacing current values with a pre-defined value) or shuffling values.
- **Reassignment:** This new, manipulated data set is reassigned to existing clusters through soft or hard labels. For each observation from the sampling stage, we receive a vector of soft labels or a single hard label.

– **Aggregation:** The soft or hard labels from the reassignment stage are aggregated in various ways, e.g., they can be averaged (soft labels) or counted (hard labels) cluster-wise.

The only prerequisite is an existing clustering based on an algorithm that can reassign instances to existing clusters through soft or hard labels. Methods only differ with respect to the intervention and aggregation stages. Next, we present our two novel FACT methods SMART and IDEA.

3.1 Scoring Metric After Permutation (SMART)

The intervention stage consists of shuffling values for a subset of features S in the data set $\mathcal{D}$ (i.e., jointly shuffling rows for a subset of columns); the aggregation stage consists of measuring the change in cluster assignments through an appropriate scoring function h applied to a confusion matrix consisting of original cluster assignments and cluster assignments after shuffling. When comparing original cluster assignments and the ones after shuffling the data, we can create a confusion matrix (see Appendix A) in the same way as in multi-class classification. One option to evaluate the confusion matrix is to directly use a scoring metric suitable for multiple clusters, e.g., the percentage of observations changing clusters after the intervention as in G2PC (found in all non-diagonal elements of the confusion matrix, see Eq. (1) for a definition). If one is interested in a scoring metric specifically developed for binary confusion matrices, the alternative is to consider binary comparisons of cluster c versus the remaining clusters. The results of all binary comparisons can then be aggregated either through a micro or a macro-averaged score (see Appendix B). Established scoring metrics based on binary confusion matrices include the F1 score (see Appendix B), Rand [32], or Jaccard [21] index. The micro-averaged score (hereafter referred to as micro score) is a suitable metric if all instances shall be considered equally important. The macro-averaged score (hereafter referred to as macro score) suits a setting where all classes (i.e., clusters in our case) shall be considered equally important. In general terms, the scoring function maps a confusion matrix to a scalar scoring metric. A multi-cluster scoring function is defined as:

$$h_{\text{multi}} : \mathbb{N}_0^{k \times k} \mapsto \mathbb{R}$$

A binary scoring function is defined as:

$$h_{\text{binary}} : \mathbb{N}_0^{2 \times 2} \mapsto \mathbb{R}$$

Let $M \in \mathbb{N}_0^{k \times k}$ denote the multi-cluster confusion matrix and $M_c \in \mathbb{N}_0^{2 \times 2}$ the binary confusion matrix for cluster c versus the remaining clusters (see Appendix A for details). SMART for feature set S corresponds to:

$$\begin{aligned} \text{Multi-cluster scoring:} \quad & \text{SMART}(\mathcal{D}, \tilde{\mathcal{D}}_S) = h_{\text{multi}}(M) \\ \text{Binary scoring:} \quad & \text{SMART}(\mathcal{D}, \tilde{\mathcal{D}}_S) = \text{AVE}(h_{\text{binary}}(M_1), \ldots, h_{\text{binary}}(M_k)) \end{aligned}$$

where AVE averages a vector of binary scores, e.g., via micro or macro averaging. In order to reduce variance in the estimate from shuffling the data, one can shuffle t times and evaluate the distribution of scores. Let $\tilde{\mathcal{D}}_S^{(t)}$ denote the t-th shuffling iteration for feature set S. The SMART point estimate is given by:

$$\overline{\text{SMART}}(\mathcal{D}, \tilde{\mathcal{D}}_S) = \psi\left(\text{SMART}(\mathcal{D}, \tilde{\mathcal{D}}_S^{(1)}), \ldots, \text{SMART}(\mathcal{D}, \tilde{\mathcal{D}}_S^{(t)})\right)$$

where ψ extracts a sample statistic such as the mean or median.

We can demonstrate the equivalency between directly applying the G2PC scoring metric to the confusion matrix and micro averaging F1 scores[3]. Given a multi-cluster confusion matrix M (see Appendix A), G2PC is defined as:

$$\begin{aligned}\text{G2PC}(M) &= \frac{1}{n}\left(\sum_{i=1}^{k}\sum_{j=1}^{k} \#_{ij} - \sum_{l=1}^{k} \#_{ll}\right) \\ &= \frac{1}{n}\left(n - \sum_{l=1}^{k} \#_{ll}\right) \\ &= 1 - \frac{1}{n}\sum_{l=1}^{k} \#_{ll} \end{aligned} \tag{1}$$

The micro F1 score is equivalent to accuracy (for settings where each instance is assigned a single label), so the following relation holds (refer to Appendix D for a detailed proof):

Theorem 1 (Equivalency between SMART with micro F1 and G2PC).

$$1 - \text{G2PC}(M) = \text{AVE}_{\text{MICRO}}(\text{F1}(M_1), \ldots, \text{F1}(M_k)) = \text{F1}_{\text{micro}}(M)$$

Proof sketch. In our utilization of confusion matrices, a "false classification" corresponds to a change in clusters after the intervention, and a "true classification" corresponds to an observation staying in the same cluster. It follows that accuracy (ACC) *represents the global percentage of observations staying in the initial cluster after the intervention stage:* $1 - \text{ACC}(M) = \text{G2PC}(M)$.

$\text{AVE}_{\text{MICRO}}(\text{F1}(M_1), \ldots, \text{F1}(M_k))$ *can be directly derived from the multi-cluster matrix* M *and is denoted by* $\text{F1}_{\text{micro}}(M)$. *Let* TP *denote the number of true positive labels,* FP *the number of false positives, and* FN *the number of false negatives. For multi-class classification problems,* FP = FN *and thus:*

$$\text{F1}_{\text{micro}}(M) = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \text{ACC}(M)$$

It follows that $1 - \text{G2PC}(M) = \text{F1}_{\text{micro}}(M)$. □

[3] Micro averaging refers to a strategy of aggregating binary comparisons where each instance is considered equally important. For the F1 score, the equivalency can be directly derived from the multi-cluster confusion matrix and involves summing up all diagonal elements (true positives) and remaining elements (false positives or false negatives). See Appendices B and D for details.

Micro F1 scores are unsuited for unbalanced classes in classification settings, as they treat each instance as equally important. From the direct dependency between G2PC and micro F1, it follows that for clusters that considerably differ in size (i.e., imbalanced clusters), G2PC does not accurately represent the importance of features, as it is dominated by larger clusters. SMART in turn allows more flexible interpretations than G2PC, e.g., by using macro F1 scores.

We can also directly evaluate binary comparisons of the found clusters to obtain cluster-specific FAs. Recall that a cluster-specific FA provides information regarding how a feature influences reassignments of instances to one specific cluster. Algorithms 1 and 2 describe the cluster-specific and global SMART algorithms, respectively. The algorithms are applied in Sects. 5 and 6. See Fig. 10 for visualized outcomes. Note that the resampling procedure to reduce the variance of estimates is optional and that global SMART can also involve binary comparisons (which requires running cluster-specific SMART), e.g., via macro averaging; we circumscribe all such different variants as the computation of the multi-cluster score h.

Algorithm 1. Cluster-Specific SMART

run clustering algorithm
for all iter $\in \{1, \ldots, t\}$ **do**
  shuffle columns S
  compute hard labels
  for all $c \in \{1, \ldots, k\}$ **do**
    create a binary confusion matrix
    compute score $h_c^{(\text{iter})}$ from confusion matrix
  end for
end for
for all $c \in \{1, \ldots, k\}$ **do**
  evaluate distribution of $\{h_c^{(\text{iter})}\}_{\text{iter} \in \{1,\ldots,t\}}$
end for

Algorithm 2. Global SMART

run clustering algorithm
for all iter $\in \{1, \ldots, t\}$ **do**
  shuffle columns S
  compute hard labels
  create a multi-cluster confusion matrix
  compute multi-cluster score $h^{(\text{iter})}$
end for
evaluate distribution of $\{h^{(\text{iter})}\}_{\text{iter} \in \{1,\ldots,t\}}$

3.2 Isolated Effect on Assignment (IDEA)

IDEA for soft labeling algorithms (sIDEA) indicates the soft label that an observation $\mathbf{x}$ with replaced values $\tilde{\mathbf{x}}_S$ is assigned to each c-th cluster. IDEA for hard labeling algorithms (hIDEA) indicates the cluster assignment of an observation $\mathbf{x}$ with replaced values $\tilde{\mathbf{x}}_S$. Both are described by the clustering (assignment) function f:

$$\text{IDEA}_{\mathbf{x}}(\tilde{\mathbf{x}}_S) = \text{sIDEA}_{\mathbf{x}}(\tilde{\mathbf{x}}_S) = \text{hIDEA}_{\mathbf{x}}(\tilde{\mathbf{x}}_S) = f(\tilde{\mathbf{x}}_S, \mathbf{x}_{-S})$$

sIDEA corresponds to a k-way vector:

$$\begin{aligned}\text{sIDEA}_{\mathbf{x}}(\tilde{\mathbf{x}}_S) &= \left(f^{(1)}(\tilde{\mathbf{x}}_S, \mathbf{x}_{-S}), \ldots, f^{(k)}(\tilde{\mathbf{x}}_S, \mathbf{x}_{-S})\right) \\ &= \left(\text{sIDEA}^{(1)}_{\mathbf{x}}(\tilde{\mathbf{x}}_S), \ldots, \text{sIDEA}^{(k)}_{\mathbf{x}}(\tilde{\mathbf{x}}_S)\right)\end{aligned}$$

Note that although IDEA is a local method, we typically compute it for a subset of observations selected in the sampling stage. The intervention stage consists of replacing $\mathbf{x}_S$ (for an observation $\mathbf{x}$) by $\tilde{\mathbf{x}}_S$. Algorithm 3 describes the computation of the local IDEA.

Algorithm 3. Local IDEA

```
run clustering algorithm
sample m vectors of feature values {x̃_S^(j)}_{j∈{1,...,m}}
for all i ∈ {1, ..., n} do
    for all j ∈ {1, ..., m} do
        generate hypothetical observation x = (x̃_S^(j), x_{-S}^(i))
        IDEA_{x^(i)}(x̃_S^(j)) = f(x)
    end for
end for
```

During the aggregation stage, we aggregate local IDEAs to a global function. For soft labeling algorithms, we can compute a point-wise average of soft labels for each cluster; for hard labeling algorithms, we can compute the fraction of hard labels for each cluster. The global IDEA is denoted by the corresponding data set $\mathcal{D}$. The global sIDEA corresponds to:

$$\text{sIDEA}_{\mathcal{D}}(\tilde{\mathbf{x}}_S) = \left(\frac{1}{n}\sum_{i=1}^{n}\text{sIDEA}^{(1)}_{\mathbf{x}^{(i)}}(\tilde{\mathbf{x}}_S), \ldots, \frac{1}{n}\sum_{i=1}^{n}\text{sIDEA}^{(k)}_{\mathbf{x}^{(i)}}(\tilde{\mathbf{x}}_S)\right) \tag{2}$$

where the c-th vector element is the average c-th element of local sIDEA vectors. The global hIDEA corresponds to:

$$\text{hIDEA}_{\mathcal{D}}(\tilde{\mathbf{x}}_S) = \left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_1(\text{hIDEA}_{\mathbf{x}^{(i)}}(\tilde{\mathbf{x}}_S)), \ldots, \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_k(\text{hIDEA}_{\mathbf{x}^{(i)}}(\tilde{\mathbf{x}}_S))\right) \tag{3}$$

where the c-th vector element is the fraction of hard label reassignments to the c-th cluster. Algorithm 4 describes the computation of the global IDEA. See Sects. 5 and 6 for applications of the local and global IDEA and Figs. 6, 7, and 11 for visualizations.

A useful interpretation for hard labeling algorithms can be obtained by visualizing the percentage of all labels per isolated intervention. The fraction of the most frequent hard label indicates the – as we call it – "certainty" of the global IDEA function for hard labeling algorithms (see Fig. 6 on the left).

Whether the global IDEA can serve as a good description of the feature effect on the reassignment depends on the heterogeneity of underlying local effects. If substituting a feature set by the same values for all instances results in similar reassignments for most instances, the global IDEA is a good interpretation instrument. Otherwise, further investigations into the underlying local effects are required.

Algorithm 4. Global IDEA

```
run clustering algorithm
sample m vectors of feature values {x̃_S^(j)}_{j∈{1,...,m}}
for all i ∈ {1,...,n} do
    compute IDEA_{x^(i)} (see Algorithm 3)
end for
for j ∈ {1,...,m} do
    for c ∈ {1,...,k} do
        if soft labeling algorithm then
            compute sIDEA_D^(c)(x̃_S^(j)) (see Eq. 2)
        else
            compute hIDEA_D^(c)(x̃_S^(j)) (see Eq. 3)
        end if
    end for
end for
```

Initial Cluster Effect on IDEA: If there is a certain within-cluster homogeneity, we ought to see similar shapes of local IDEA functions depending on the observations' initial cluster (before the intervention stage). Let c_{init} denote the initial cluster index. We receive one aggregate IDEA per initial cluster (we refrain from using the word "global" here, as there is a separate, global IDEA independent from the initial cluster), which reflects the aggregate, isolated effect of an intervention in the feature(s) of interest on the assignment to cluster c **per initial cluster** c_{init}:

$$\text{IDEA}_{\mathcal{D}^{(c_{\text{init}})}}(\tilde{\mathbf{x}}_S) = \left(\text{IDEA}^{(1)}_{\mathcal{D}^{(c_{\text{init}})}}(\tilde{\mathbf{x}}_S), \ldots, \text{IDEA}^{(k)}_{\mathcal{D}^{(c_{\text{init}})}}(\tilde{\mathbf{x}}_S)\right) \tag{4}$$

whose components correspond to (depending on the clustering algorithm output):

$$\text{sIDEA}^{(c)}_{\mathcal{D}^{(c_{\text{init}})}}(\tilde{\mathbf{x}}_S) = \frac{1}{n^{(c_{\text{init}})}} \sum_{i \,:\, \mathbf{x}^{(i)} \in \mathcal{D}^{(c_{\text{init}})}} \text{sIDEA}^{(c)}_{\mathbf{x}^{(i)}}(\tilde{\mathbf{x}}_S)$$

$$\text{hIDEA}^{(c)}_{\mathcal{D}^{(c_{\text{init}})}}(\tilde{\mathbf{x}}_S) = \frac{1}{n^{(c_{\text{init}})}} \sum_{i \,:\, \mathbf{x}^{(i)} \in \mathcal{D}^{(c_{\text{init}})}} \mathbb{1}_c(\text{hIDEA}_{\mathbf{x}^{(i)}}(\tilde{\mathbf{x}}_S))$$

where $n^{(c_{\text{init}})}$ corresponds to the number of observations within initial cluster c_{init}. This definition lends itself to a convenient visualization per initial cluster, which we showcase in Fig. 7.

4 Additional Notes on FACT

How to Generate Feature Values for Interventions: A simple option is to use a feature's sample distribution, i.e., all observed values. In classical SA of model output [34], one typically intends to explore the feature space as thoroughly as possible (space-filling designs). In SL, there are valid arguments against space-filling designs due to potential model extrapolations, i.e., predictions in areas where the model was not trained with enough data [19,29]. In clustering, the absence of model performance issues allows us to fill the feature space as extensively as possible, e.g., with unit distributions, random, or quasi-random (also referred to as low-discrepancy) sequences (e.g., Sobol sequences) [34]. In fact, assigning unseen data to the clusters serves our purpose of visualizing the decision boundaries between the clusters determined by the clustering algorithm.

Generating Feature Values for SMART and IDEA: For SMART, we evaluate a fixed data set and jointly shuffle values of the feature set S. For IDEA, we can either use observed values or strive for a more space-filling design. More values result in better FAs but higher computational costs.

Reassigning versus Reclustering: FACT aims to explain a given clustering of the data. The found clustering outcome is treated as "a snapshot in time", similarly to how explanations in SL are conditional on a trained model. FACT methods are therefore akin to model-agnostic interpretation methods in SL. It follows that we need a reassignment of instances to pre-found clusters instead of a reclustering (running the clustering algorithm from the ground up). Reclustering artificial data would result in a "concept drift" and different clusters, thus being counterproductive to our goals.

In Fig. 2 (left), we create an artificial data set using the Cartesian product of the original bivariate data that forms 3 clusters and reassign the artificially created observations to the found clusters of a cluster model fitted on the original bivariate data (grid lines). The right plot visualizes a reclustering of the same artificial data set, resulting in clearly visible changes in the shape and position of the clusters.

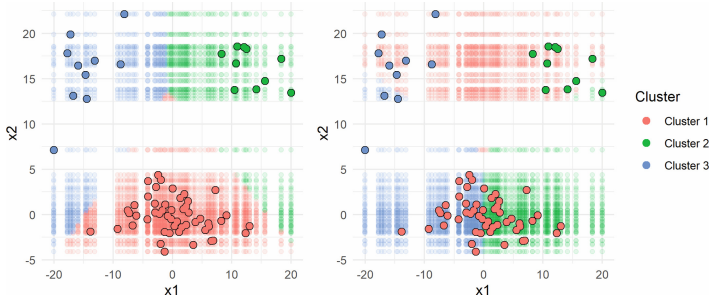


Fig. 2. Observations (solid points) and Cartesian product (transparent grid) reassigned (left plot) and reclustered (right plot).

How the FACT Framework is Algorithm-Agnostic: How to reassign instances differs across clustering algorithms. For instance, in k-means we assign an instance to the cluster with the lowest Euclidean distance; in probabilistic clustering such as Gaussian mixture models we select the cluster associated with the largest probability; in hierarchical clustering, we select the cluster with the lowest linkage value, etc. [8]. In other words, although the implementation of the reassignment stage differs across algorithms (the computation of soft or hard labels), FACT methods stay exactly the same. For FACT to be truly algorithm-agnostic, we develop variants to accommodate both soft and hard labeling algorithms.

Limitations: FACT is not suited for evaluating the quality of the clustering, i.e., whether clusters have a high within-cluster homogeneity and high between-cluster heterogeneity. Furthermore, we need an appropriate assignment function that assigns instances to existing clusters and which may frequently not be available. Particularly IDEA is limited by computational constraints for large data sets. Hence, we introduce a sampling stage for FACT, where only a subset of clustered observations can be selected to estimate FAs.

5 Simulations

5.1 Flexibility of SMART - Micro F1 versus Macro F1

In this simulation, we illustrate that the micro F1 score and therefore also the G2PC proposed in [8] is not useful for imbalanced cluster sizes. We also demonstrate the advantages of our more flexible SMART approach, which allows us to use the macro F1 score instead, a scoring metric better suited for imbalanced cluster sizes. We simulate a data set with two features consisting of 4 differently sized classes (see Fig. 3), where each class follows a different bivariate normal distribution. 60 instances are sampled from class 3 while 20 instances are sampled from each of the remaining classes. To capture the latent class variable, c-means is initialized at the 4 centers. The right plot in Fig. 3 displays the perfect cluster

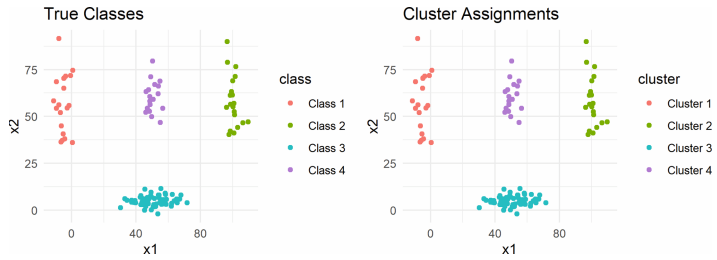


Fig. 3. Visualization of the data and the perfect clustering of c-means.

assignments found by c-means. We can see that x_1 is the defining feature of the clustering for 3 out of 4 clusters, i.e., for the clusters enumerated by 1, 2, and 4. Our goal is to analyze the c-means clustering model to discover which of the two features were more important for the clustering outcome.

We now compare the macro F1 score and micro F1 score (see Appendix B) for x_1 and x_2. Both features have micro F1 median scores of 0.58, suggesting equal importance for x_1 and x_2. Recall that the micro F1 score corresponds to 1 - G2PC (see Theorem 1). This implies that G2PC is unable to identify a meaningful feature importance ranking for x_1 and x_2 in this case. Macro F1 on the other hand is different for both features ($x_1 = 0.43, x_2 = 0.64$), indicating that x_1 is more important. Note that the F1 score is a similarity index. A low F1 score indicates a high feature importance, i.e., a high dissimilarity between the clustering outcome based on the original data and the clustering outcome after the feature of interest has been shuffled. These results stem from the fact that micro F1 accounts for each instance with equal importance (by globally counting true and false positives, see Appendix B). Cluster 3 is over-represented with three times as many instances as the remaining clusters. The macro F1 score accurately captures this by treating each cluster as equally important, regardless of its size.

5.2 Global versus Cluster-Specific SMART

Next, we demonstrate that even when using the macro F1 score for imbalanced clusters, the results may obfuscate the importance of features to specific clusters, which is where cluster-specific SMART becomes the method of choice. We simulate three visibly distinctive classes (left plot in Fig. 4) where each class follows a bivariate normal distribution with different mean and covariance matrices. 50 instances are sampled from class 2, and 20 instances are sampled from class 1 and class 3 each. We initialize c-means at the 3 mean values. As shown in Fig. 4, the cluster assignments capture all three classes almost perfectly, except for one instance of class 2 being assigned to cluster 1 and one to cluster 3.

We compare the global macro F1 (which weights the importance of clusters equally) to the cluster-specific F1 score. With a global macro F1 median of 0.62 for x_1 and 0.66 for x_2, there is no difference between the importance of both

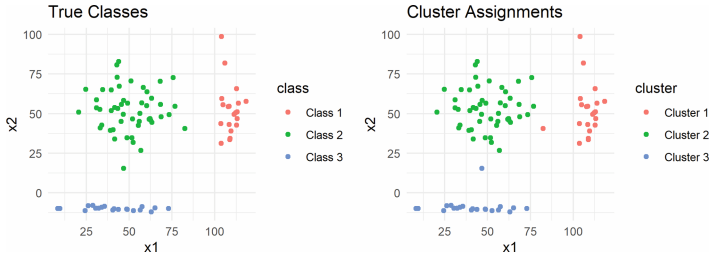


Fig. 4. Three classes with different distributions clustered by c-means. True classes (left) and clusters (right) almost perfectly match.

features for the overall clustering. In contrast, cluster-specific SMART offers a more detailed view of the contributions of each feature to the clustering outcome. Both features, x_1 and x_2, have an equal regional feature importance of 0.73 in forming cluster 2. For cluster 3, feature x_2 is considerably more important with a macro F1 score of 0.26, compared to 0.86 for feature x_1. Vice versa, feature x_1 is the defining feature of cluster 1 with a score of 0.24. In comparison, the importance of x_2 for cluster 1 is 1.0, implying that the permutation of feature x_2 had no effect on the assignment criteria for cluster 1.

5.3 How to Interpret IDEA

Here, we demonstrate how IDEA can visualize isolated, univariate effects of features on the cluster assignments of multi-dimensional data; how the heterogeneity of local effects influences the explanatory power of the global IDEA; and how grouping IDEA curves by initial cluster assignments reveals similar effects. We draw 50 instances from three multivariate normally distributed classes. To make them differentiable for the clustering algorithm, the classes are generated with an antagonistic mean structure. The covariance matrix of the three classes is sampled using a Wishart distribution (see Appendix C for details). The left plot in Fig. 5 depicts the three-dimensional distribution of the classes. We intend class 3 to be dense and classes 1 and 2 to be less dense but large in hypervolume. We initialize c-means at the 3 centers and optimize via the Euclidean distance. Figure 5 visualizes the perfect clustering. Figure 6 (left) displays an hIDEA plot for x_1 (see Sect. 3.2), indicating the majority vote of cluster assignments when exchanging values of x_1 by the horizontal axis value for all observations.

The curves in Fig. 6 (right) represent the cluster-specific components of the sIDEA function (local and global). Note that this refers to the effect of observations being reassigned to the c-th cluster and not the initial cluster effect, which we demonstrate below. The bandwidths represent the local IDEA curve ranges that were averaged to receive the respective global IDEA. We can see that - on average - x_1 has a substantial effect on the clustering outcome. The lower the value of x_1 that is plugged into an observation, the more likely it is assigned to cluster 1, while for larger values of x_1 it is more likely to be assigned to cluster

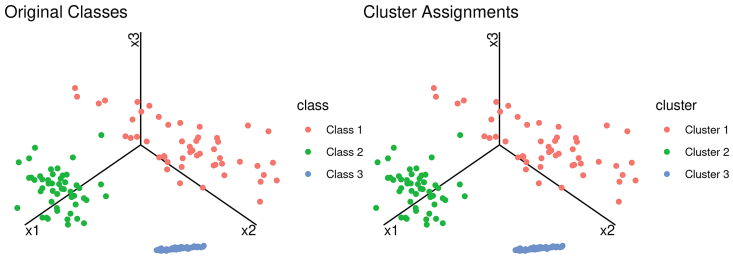


Fig. 5. Sampled classes (left plot) versus clusters (right plot).

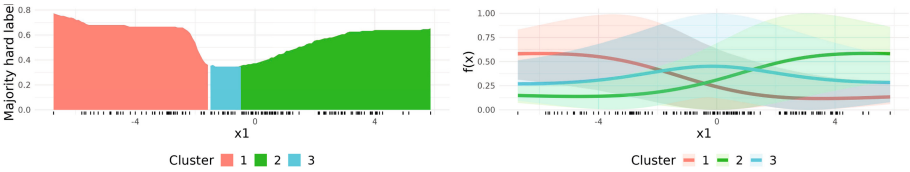


Fig. 6. Left: A plot indicating "certainty" of the global hIDEA function. On average, replacing x_1 by the axis value results in an observation being assigned to the color-indicated cluster. The vertical distance indicates how many observations are assigned to the majority cluster. **Right:** Cluster-specific global sIDEA curves. Each curve indicates the average soft label of an observation being assigned to the c-th cluster if its x_1 value is replaced by the axis value. The bandwidths visualize the distribution of local sIDEA curves that were vertically averaged to the respective global, cluster-specific sIDEA.

2. For $x_1 \approx 0$, observations are more likely to be assigned to cluster 3. The large bandwidths indicate that the clusters are spread out, and plugging in different values of x_1 into an observation has widely different effects across the data set. Particularly around $x_1 \approx 0$, where cluster 3 dominates, the average effect loses its meaning due to the underlying local IDEA curves being highly heterogeneous. In this case, one should be wary of the interpretative value of the global IDEA. We proceed to investigate the heterogeneity of the local sIDEA curves for cluster 3 (see Fig. 7 on the left). The flat shape of the cluster-specific global sIDEA indicates that x_1 has a rather low effect on observations being assigned to cluster 3. However, the cluster-specific local sIDEA curves reveal that individual effects cancel each other out when being averaged.

Initial Cluster Effect: It seems likely that observations belonging to a single cluster in the initial clustering run would behave similarly once their feature values are changed. We color each sIDEA curve by the original cluster assignment (see Fig. 7 on the right) and add the corresponding aggregate curves. Our assumption - that observations within a cluster behave similarly once we make isolated changes to their feature values - is confirmed. The formal definition of this initial cluster effect is given by Eq. (4).

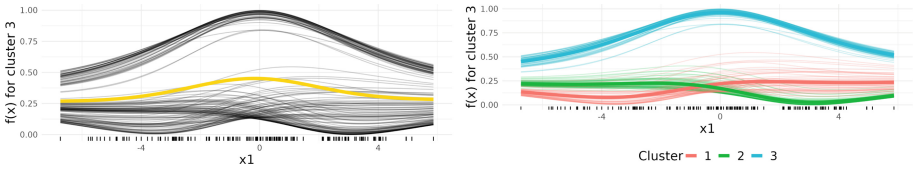


Fig. 7. Left: Cluster-specific IDEA (local and global), indicating effects on the soft labels for observations to be assigned to cluster 3. The black lines represent local effects; the yellow line the global effect. **Right:** sIDEA curves colored by initial cluster assignment. The thin curves represent local effects; the thick curves represent aggregate effects. We can see similar effects of replacing the values of x_1 on the soft labels, depending on what initial cluster an observation is part of.

5.4 IDEA Recovers Distribution Found by Clustering Algorithms

This simulation demonstrates how the global sIDEA can "recover" the distributions found by the clustering algorithm. We simulate 4 features and cluster the data into 3 clusters with FuzzyDBSCAN [20]. We illustrate soft labels for assignments to a single cluster in Fig. 8. The upper triangular plots display true bivariate marginal densities of features. The lower triangular plots display the corresponding bivariate global sIDEA estimates. Matching pairs of densities and sIDEA estimates "mirror" each other on the diagonal line. The diagonal plots visualize univariate marginal distributions (grey area) versus the corresponding estimated univariate global sIDEA curve (black line). The location and shape of sIDEA plots approximate the true marginal distributions. Note that for the correlated pairs (x_1, x_2) and (x_3, x_4), we recover the direction of the correlation.

6 Real Data Application

The Wisconsin diagnostic breast cancer (WDBC) data set [7] consists of 569 instances of cell nuclei obtained from breast mass. Each instance consists of 10 characteristics derived from a digitized image of a fine-needle aspirate. For each characteristic, the mean, standard error and "worst" or largest value (mean of the three largest values) is recorded, resulting in 30 features of the data set. Each nucleus is classified as malignant (cancer, class 1) or benign (class 2). We cluster the data using Euclidean optimized c-means. Figure 9 visualizes the projection of the data onto the first two PCs. The clusters cannot be separated with two PCs, and the visualization is of little help in understanding the influence of the original features on the clustering outcome.

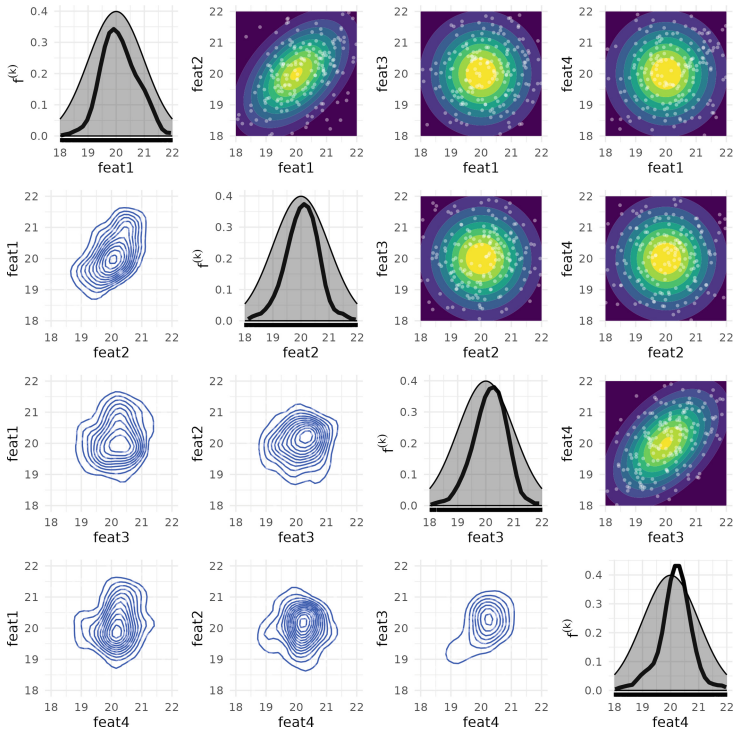


Fig. 8. Comparison of true bivariate marginal densities of features (upper triangular plots) with corresponding global bivariate sIDEA (lower triangular plots) and true univariate marginal densities of features (diagonal plots, grey area) with corresponding global univariate sIDEA (diagonal plots, black line). (Color figure online)

6.1 Aggregate FA for Each Cluster (SMART)

We first showcase how SMART can serve as an approximation of the actual reclustering. Measured on the latent target variable, the initial clustering run has an F1 score of 0.88. We then recluster the data, once with the 4 most important and once with the 4 least important features. Dropping the 26 least important features only reduces the F1 score by 0.03 to 0.85 (measured using the latent target). In contrast, using the 4 least important features reduces the F1 score by 0.55 to 0.33 and thus alters the clustering in a major way. This demonstrates that assigning new instances to existing clusters can serve as an efficient method for feature selection. To showcase the grouped feature importance, we jointly shuffle features and compare their importance in Fig. 10. Note that we use the natural logarithm of SMART here for better visual separability and to receive a natural ordering of the feature importance (due to F1 being a similarity index), where a larger bar indicates a higher importance and vice versa.

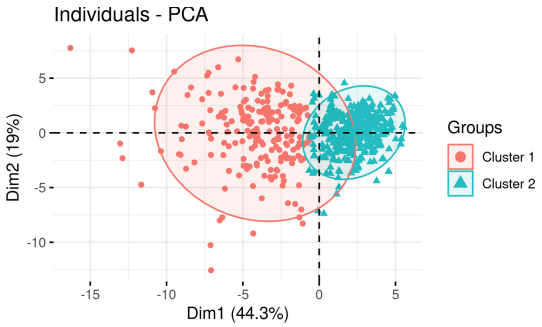


Fig. 9. First and second PCs of WDBC data with clusters of real target values.

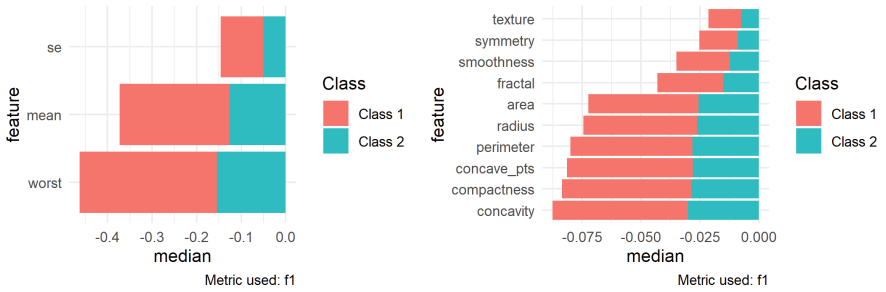


Fig. 10. Grouped SMART (using the natural logarithm) per cluster for groups of categories (left plot) and groups of characteristics (right plot) in the WDBC data set.

6.2 Visualizing Marginal Feature Effects (IDEA)

We now visualize isolated univariate and bivariate effects of features on assignments. Figure 11 plots the global IDEA curve for three features `concavity_worst`, `compactness_worst`, and `concave_points_worst`. The transparent areas indicate the regions where the local curve mass is located. A rug on the horizontal axis shows the distribution of the corresponding feature. For all three features, larger values result in observations being assigned to cluster 1, while lower values result in observations being assigned to cluster 2. The distribution of cluster-specific local IDEA curves is wide, reflecting voluminous clusters. All features have a strong univariate effect on the cluster assignments, which indicates a large importance of each feature to the constitution of each cluster.

Figure 11 (right) plots the two-dimensional sIDEA for `compactness_worst` and `compactness_mean`. The color indicates what cluster the observations are assigned to on average when `compactness_worst` and `compactness_mean` are replaced by the axis values. The transparency indicates the magnitude of the soft label, i.e., the "certainty" in our estimate. On average, the observations are assigned to cluster 2 when adjusting both features to lower values and to cluster 1 when adjusting both features to higher values.

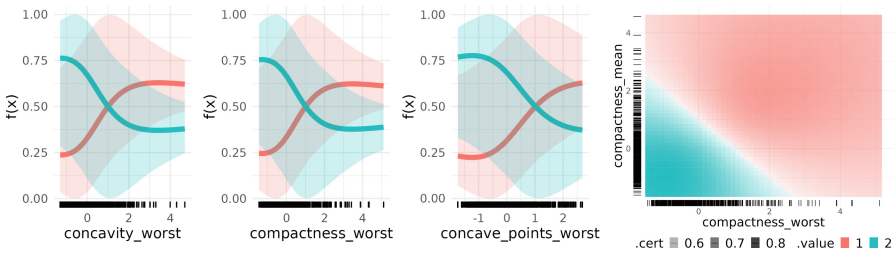


Fig. 11. Left: Univariate global sIDEA plots for the features `concavity_worst`, `compactness_worst`, and `concave_points_worst`. **Right:** Two-dimensional sIDEA for the features `compactness_worst` and `compactness_mean`. On average, an observation is assigned to cluster 1 for large values of both features, while it is assigned to cluster 2 for low values of both features.

7 Conclusion

This research paper proposes FACT, a framework to produce FAs which is compatible with any clustering algorithm able to reassign instances through soft or hard labels, preserves the integrity of the data, and does not introduce additional models. FACT techniques provide information regarding the importance of features for assigning instances to clusters (overall and to specific clusters); or how isolated changes in feature values affect the assignment of single instances or the entire data set to each cluster. We introduce two novel FACT methods: SMART and IDEA. SMART is a general framework that outputs a single global value for each feature indicating its importance to cluster assignments or one value for each cluster (and feature). IDEA adds to these capabilities by visualizing the structure of the feature influence on cluster assignments across the feature space for single observations and the entire data set.

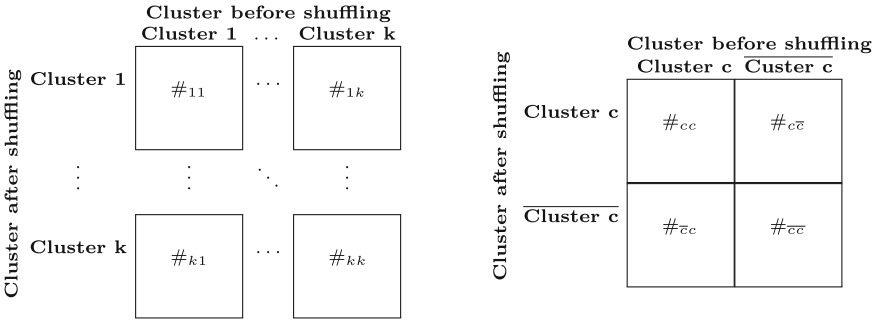
Although explaining algorithmic decisions is an active research topic in SL, it is largely ignored for clustering algorithms. The FACT framework provides a new impetus for algorithm-agnostic interpretations in clustering. With SMART and IDEA, we hope to establish a foundation for the future development of FACT methods and spark more research in this direction.

A Confusion Matrix for SMART

Transferring the concept of confusion matrices from classification tasks, a "true" classification would correspond to an observation staying within the same cluster after the intervention, and a "false" classification would result in a reassignment to a different cluster.

For the multi-cluster matrix on the left, let TP denote the sum of all true positives from all binary comparisons of cluster c versus the remaining clusters, FP the sum of all false positives, and FN the sum of all false negatives. It follows that $\sum_{l=1}^{k} \#_{ll} = \text{TP}$ and $n - \sum_{l=1}^{k} \#_{ll} = \text{FP} = \text{FN}$.

Table 1. Multi-cluster and binary confusion matrices for SMART.



For the binary matrix on the right, let TP_c denote all true positives of cluster c versus the remaining clusters, FP_c all false positives, FN_c all false negatives, and TN_c all true negatives. It follows that $\#_{cc} = \mathrm{TP}_c$, $\#_{c\bar{c}} = \mathrm{FP}_c$, $\#_{\bar{c}c} = \mathrm{FN}_c$, and $\#_{\bar{c}\bar{c}} = \mathrm{TN}_c$.

B Scores

F_β score: Balances false positives and false negatives. The F_β score of cluster c versus the remaining ones corresponds to:

$$F_{\beta,c} = \frac{\left(\beta^2+1\right)\cdot P_c \cdot R_c}{\beta^2 \cdot P_c + R_c}, \quad \text{where} \quad P_c = \frac{\#_{cc}}{\#_{cc}+\#_{\bar{c}c}} \quad \text{and} \quad R_c = \frac{\#_{cc}}{\#_{cc}+\#_{c\bar{c}}}$$

The F_1 (which we refer to as F1) score simplifies to:

$$F_{1,c} = 2\frac{P_c \cdot R_c}{P_c + R_c}$$

Given a multi-cluster confusion matrix M, let ϕ_c be an arbitrary binary scoring function dependent on TP, FP, FN, and TN. $\mathcal{S}_{\text{macro}}$ denotes the multi-cluster macro score that treats each cluster with equal importance. $\mathcal{S}_{\text{micro}}$ denotes the multi-cluster micro score that treats each instance with equal importance:

$$\mathcal{S}_{\text{macro}}(M) = \frac{1}{k}\sum_{c=1}^{k} \phi\left(\mathrm{TP}_c, \mathrm{FP}_c, \mathrm{FN}_c, \mathrm{TN}_c\right)$$

$$\mathcal{S}_{\text{micro}}(M) = \phi\left(\sum_{c=1}^{k}\mathrm{TP}_c, \sum_{c=1}^{k}\mathrm{FP}_c, \sum_{c=1}^{k}\mathrm{FN}_c, \sum_{c=1}^{k}\mathrm{TN}_c\right)$$

C Wishart Distribution

We sample the covariance matrix M from the Wishart distribution with $M \sim \mathrm{Wishart}_3(3, \Sigma)$. Σ is constructed using $\Sigma_{\text{Class 1}} = 0.6I_3$, $\Sigma_{\text{Class 2}} = 0.3I_3$, and

$\Sigma_{\text{Class 3}} = 0.15I_3$, where I_3 refers to the 3×3 identity matrix. As a result, the variance of class 1 is the largest, the variance of class 3 is the lowest, and the variance of class 2 lies between the variances of classes 1 and 3.

D Proofs

Proof (Theorem 1).

Recall the definition of G2PC with respect to a multi-cluster confusion matrix M (see Table 1 in Appendix A):

$$\text{G2PC}(M) = \frac{1}{n}\left(\sum_{i=1}^{k}\sum_{j=1}^{k}\#_{ij} - \sum_{l=1}^{k}\#_{ll}\right) = \frac{1}{n}\left(n - \sum_{l=1}^{k}\#_{ll}\right) = 1 - \frac{1}{n}\sum_{l=1}^{k}\#_{ll}$$

Let TP denote the number of true positive labels, FP the number of false positives, and FN the number of false negatives. The sum of diagonal elements corresponds to TP:

$$\sum_{l=1}^{k}\#_{ll} = \text{TP}$$

It follows that:

$$\text{G2PC}(M) = 1 - \frac{\text{TP}}{n}$$

TP divided by the absolute number of instances equals the percentage of "correctly classified instances" (the number of instances staying within the same cluster after the intervention in our case) which corresponds to accuracy (ACC):

$$\frac{1}{n}\sum_{l=1}^{k}\#_{ll} = \frac{\text{TP}}{n} = \text{ACC}(M)$$

It follows that:

$$\text{G2PC}(M) = 1 - \text{ACC}(M) \Leftrightarrow 1 - \text{G2PC}(M) = \text{ACC}(M) \tag{5}$$

The following relation holds by definition for the micro F1 score [38]:

$$\text{F1}_{\text{micro}}(M) = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})}$$

For multi-class classification it holds that FP = FN, as every false positive for one class is a false negative for another class. With $n = \text{TP} + \text{FP}$, it follows that:

$$\text{F1}_{\text{micro}}(M) = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{n} = \text{ACC}(M) \tag{6}$$

From Eqs. (5) and (6), we have:

$$1 - \text{G2PC}(M) = \text{F1}_{\text{micro}}(M)$$

□

References

1. Achtert, E., Böhm, C., Kriegel, H.P., Kröger, P., Zimek, A.: Deriving quantitative models for correlation clusters. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2006, pp. 4–13. Association for Computing Machinery, New York, NY, USA (2006)
2. Apley, D.W., Zhu, J.: Visualizing the effects of predictor variables in black box supervised learning models. J. R. Stat. Soc. Ser. B **82**(4), 1059–1086 (2020)
3. Bertsimas, D., Orfanoudaki, A., Wiberg, H.: Interpretable clustering via optimal trees. ArXiv e-prints (2018). arXiv:1812.00539
4. Bertsimas, D., Orfanoudaki, A., Wiberg, H.: Interpretable clustering: an optimization approach. Mach. Learn. **110**(1), 89–138 (2021)
5. Blockeel, H., Raedt, L.D., Ramon, J.: Top-down induction of clustering trees. In: Proceedings of the Fifteenth International Conference on Machine Learning, ICML 1998, pp. 55–63. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998)
6. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
7. Dua, D., Graff, C.: UCI machine learning repository (2019). http://archive.ics.uci.edu/ml
8. Ellis, C.A., Sendi, M.S.E., Geenjaar, E.P.T., Plis, S.M., Miller, R.L., Calhoun, V.D.: Algorithm-agnostic explainability for unsupervised clustering. ArXiv e-prints (2021). arXiv:2105.08053
9. Fisher, A., Rudin, C., Dominici, F.: All models are wrong, but many are useful: learning a variable's importance by studying an entire class of prediction models simultaneously. J. Mach. Learn. Res. **20**(177), 1–81 (2019)
10. Fraiman, R., Ghattas, B., Svarc, M.: Interpretable clustering using unsupervised binary trees. Adv. Data Anal. Classif. **7**(2), 125–145 (2013)
11. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**(5), 1189–1232 (2001)
12. Frost, N., Moshkovitz, M., Rashtchian, C.: ExKMC: Expanding explainable k-means clustering. ArXiv e-prints (2020). arXiv:2006.02399
13. Funk, H., Scholbeck, C.A., Casalicchio, G.: FACT: Feature Attributions for CluSTering (2023). https://CRAN.R-project.org/package=FACT. R package version 0.1.0
14. Gabidolla, M., Carreira-Perpiñán, M.A.: Optimal interpretable clustering using oblique decision trees. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2022. pp. 400–410. Association for Computing Machinery, New York, NY, USA (2022)
15. Ghattas, B., Michel, P., Boyer, L.: Clustering nominal data using unsupervised binary decision trees: comparisons with the state of the art methods. Pattern Recognit. **67**, 177–185 (2017)
16. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. J. Comput. Graph. Stat. **24**(1), 44–65 (2015)
17. Hinneburg, A.: Visualizing clustering results. In: Liu, L., Özsu, M.T. (eds.) Encyclopedia of Database Systems, pp. 3417–3425. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-39940-9_617
18. Hooker, G.: Generalized functional anova diagnostics for high-dimensional functions of dependent variables. J. Comput. Graph. Stat. **16**(3), 709–732 (2007)

19. Hooker, G., Mentch, L., Zhou, S.: Unrestricted permutation forces extrapolation: variable importance requires at least one more model, or there is no free variable importance. Stat. Comput. **31**(6), 82 (2021)
20. Ienco, D., Bordogna, G.: Fuzzy extensions of the DBScan clustering algorithm. Soft. Comput. **22**(5), 1719–1730 (2018)
21. Jaccard, P.: The distribution of the flora in the alpine zone. New Phytol. **11**(2), 37–50 (1912)
22. Kinkeldey, C., Korjakow, T., Benjamin, J.J.: Towards supporting interpretability of clustering results with uncertainty visualization. In: EuroVis Workshop on Trustworthy Visualization (TrustVis) (2019)
23. Lawless, C., Kalagnanam, J., Nguyen, L.M., Phan, D., Reddy, C.: Interpretable clustering via multi-polytope machines. ArXiv e-prints (2021). arXiv:2112.05653
24. Liu, B., Xia, Y., Yu, P.S.: Clustering through decision tree construction. In: Proceedings of the Ninth International Conference on Information and Knowledge Management, CIKM, pp. 20–29. Association for Computing Machinery, New York, NY, USA (2000)
25. Loyola-González, O., et al.: An explainable artificial intelligence model for clustering numerical databases. IEEE Access **8**, 52370–52384 (2020)
26. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS 2017, pp. 4768–4777. Curran Associates Inc., Red Hook, NY, USA (2017)
27. Molnar, C.: Interpretable Machine Learning (2019). https://christophm.github.io/interpretable-ml-book/
28. Molnar, C., Casalicchio, G., Bischl, B.: Interpretable machine learning - a brief history, state-of-the-art and challenges. In: Koprinska, I., et al. (eds.) ECML PKDD 2020 Workshops, pp. 417–431. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-65965-3_28
29. Molnar, C., et al.: General pitfalls of model-agnostic interpretation methods for machine learning models. In: Holzinger, A., Goebel, R., Fong, R., Moon, T., Müller, K.R., Samek, W. (eds.) xxAI 2020. LNCS, vol. 13200, pp. 39–68. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-04083-2_4
30. Moshkovitz, M., Dasgupta, S., Rashtchian, C., Frost, N.: Explainable k-means and k-medians clustering. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 7055–7065. PMLR (2020)
31. Plant, C., Böhm, C.: INCONCO: interpretable clustering of numerical and categorical objects. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2011, pp. 1127–1135. Association for Computing Machinery, New York, NY, USA (2011)
32. Rand, W.M.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971)
33. Ribeiro, M.T., Singh, S., Guestrin, C.: "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 2016, pp. 1135–1144. Association for Computing Machinery, New York, NY, USA (2016)
34. Saltelli, A., et al.: Global Sensitivity Analysis: The Primer. John Wiley & Sons Ltd, Chichester (2008)
35. Scholbeck, C.A., Molnar, C., Heumann, C., Bischl, B., Casalicchio, G.: Sampling, intervention, prediction, aggregation: a generalized framework for model-agnostic

interpretations. In: Cellier, P., Driessens, K. (eds.) ECML PKDD 2019. CCIS, vol. 1167, pp. 205–216. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43823-4_18
36. Sobol, I.: Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. Math. Comput. Simul. **55**(1), 271–280 (2001)
37. Strumbelj, E., Kononenko, I.: An efficient explanation of individual classifications using game theory. J. Mach. Learn. Res. **11**, 1–18 (2010)
38. Takahashi, K., Yamamoto, K., Kuchiba, A., Koyama, T.: Confidence interval for micro-averaged F1 and macro-averaged F1 scores. Appl. Intell. **52**(5), 4961–4972 (2022)
39. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harvard J. Law Technol. **31**(2) (2018)