PEBAM: A Profile-based Evaluation Method for Bias Assessment on Mixed Datasets

Mieke Wilms[1], Giovanni Sileno[1], and Hinda Haned[1]

[1]University of Amsterdam, Amsterdam, the Netherlands
mieke.wilms@student.uva.nl

Abstract. Bias evaluation methods focus either on individual bias or on group bias, where groups are defined based on protected attributes such as gender or ethnicity. More generally, however, descriptively relevant combinations of feature values in the data space (*profiles*) may serve also as anchors for biased decisions. This paper introduces therefore a semi-hierarchical clustering method for profile extraction from mixed datasets. It elaborates on how profiles can be used to reveal *historical*, *representational*, *aggregation* and *evaluation biases* in algorithmic decision-making models, taking as example the German credit data set. Our experiments show that the proposed profile-based evaluation method for bias assessment on mixed datasets (PEBAM) can reveal forms of bias towards profiles expressed by the dataset that are undetected when using individual- or group-bias metrics alone.

Keywords: Algorithmic Fairness · Bias Prevention · Bias Evaluation · Clustering · Domain Analysis

1 Introduction

The wider introduction of machine learning algorithms in decision-making processes feeds an ongoing debate over algorithmic decision-making (ADM). To prevent or correct ADM from taking biased decisions several fairness-aware machine learning algorithms have been proposed [8]. However, these algorithms are not always accessible to practitioners due to their 'black-box' nature [2]; they are highly dependent on the data preprocessing phase [8]; and, at more fundamental level, fairness and bias can be given technical meanings but cannot be captured by one single definition [14, 16]. Contemporary bias evaluation methods used for fairness analysis generally focus either on individual bias or on group bias, where groups are defined based on protected attributes such as gender or ethnicity, but this is not without drawbacks. For instance, analysing the German credit dataset—a real world dataset[1] collecting features of loan applicants and a credit risk label *good* or *bad* assigned to them—the group of young individuals (age below 25) obtains more often a false negative label than the group above 25, hence young individuals are discriminated when applying for a loan [11]. The

[1] Available at: https://www.kaggle.com/uciml/german-credit

simplest solution would be to take this sensitive attribute out of consideration, however there are multiple attributes that correlate with the “age” attribute (eg. “own house” [12]). From a more general standpoint, one may ask whether there exist relevant descriptive combinations of feature values in the data space, that we will call here *profiles*, which may act as anchors for (assessing the presence of) biased decisions. As an additional source of complexity, we need also to take into account that data is commonly presented in form of mixed datasets (ie. including both categorical and numeric features). Discretization of numeric dimensions, or embedding of categorical dimensions, add further complexity and potentially undesired effects. Given this context, we address the following research questions: *How can profiles be defined? How can we extract profiles from mixed datasets? How can profiles be used to assess biases? How does a profile-based assessment compare with existing individual- or group-based methods?* The goal of this paper is to develop and test a Profile-based Evaluation method for Bias Assessment of algorithmic decision-making on Mixed datasets (PEBAM). Our contribution is twofold: (i) we present an effective and computationally efficient method for profile extraction on mixed datasets based on clustering; (ii) we show how profiles can be utilized to evaluate various forms of biases—most of them associated to trained ADM models. The paper is structured as follows. Section 2 provide a brief overview of relevant concepts. Section 3 presents the proposed methodology. Section 4 elaborates on the experiments and results on the German credit dataset. A note on future work ends the paper.

2 Theoretical background

Types of bias Several types of bias have been identified in the literature (see eg. the 23 types in [14]), but for the scope of this research we will focus in particular on biases that can arise during a ML-product lifecycle (see eg. [16]). For instance, during the *data generation* process, we may have: *historical bias*, produced by the world as it is, and occurring even if data is perfectly measured and sampled; *representation bias*, occurring when the training data for the ML model under-represents parts of the population the algorithm will be used on. During the *model building* and *implementation* phases, we may have: *aggregation bias*, arising when a general model is used for all groups, while in reality different groups have a different mapping from input features to labels (eg. some ethnic groups can have different indicators for a disease than others); *evaluation bias*, occurring when the data on which the model is evaluated is a misrepresentation of the target population. These four types of bias do not cover all possible sources of bias, but they will be used as relevant examples about how to set up a profile-based evaluation.

Bias evaluation methods for algorithmic fairness In their extensive literature review, Mehrabi et al. [14] give an overview of the most widely used definitions of fairness within machine learning, providing 3 definitions focused on individual fairness, 6 on group fairness, in which groups are defined by protected attribute classes (eg. sex, ethnicity, etc.), 1 on subgroup fairness. In this work we will build upon two (group-fairness) measures. The first is *equal opportunity* [10], a criterion for fairness in binary algorithms. Reading the outcome $y = 1$ as the "advantaged" outcome, and A as the protected class attribute, we have:

Definition 1. ***Equal opportunity*** *A binary predictor $\hat{y}$ satisfies equal opportunity w.r.t. attribute A and ground truth y iff: $Pr\{\hat{y} = 1 | A = 0, y = 1\} = Pr\{\hat{y} = 1 | A = 1, y = 1\}$.*

The second is *contextual demographic (dis)parity* (CDD)—based on *conditional (non-)discrimination* by [12]—a measure found to be the most compatible with the decisions of the European Court of Justice on cases of discrimination [18].

Definition 2. ***Conditional Demographic Disparity*** *Let R be a given set of attributes, A_r be the proportion of people belonging to a protected class in the advantaged group and with attribute $r \in R$, and let D_r be the proportion of people of protected class in the disadvantaged group with attribute r. A decision-making process exhibits conditional demographic disparity iff: $\forall r \in R : D_r > A_r$*

The conditions r in R should be *explanatory* [12], ie. they should hypothetically explain the outcome even in the absence of discrimination against the protected class (eg., different salaries between men and women might be due to different working hours). Under this view, R is derived from domain expert knowledge.

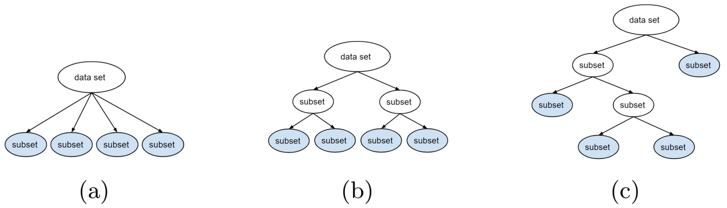
Clustering algorithm for mixed data In ADM one very often has to deal with mixed datasets, ie. datasets that consist of both categorical and numerical features. Various solutions have been proposed in the literature to the known difficulty to capture distributions on mixed datasets [15]; the present work will rely in particular on *k-medoids clustering* [5]. The main benefit of k-medoids clustering over k-means is that it is more robust to noise and outliers; we also do not have to come up with a measure to compute the mean for categorical features. On the other hand, the k-medoids clustering problem is NP-hard to solve exactly. For this reason, in our work we will make use of the heuristic Partitioning Around Medoids (PAM) algorithm [4].

3 Methodology

PEBAM (*Profile-based Evaluation for Bias Assessment for Mixed datasets*) is a method consisting of three main steps: (1) a profile selection—based on the iteration of clustering controlled by a measure of variability—to extract profiles representative of the target domain from an input dataset; (2) profiles are evaluated in terms of stability over repetitions of extractions; (3) a given ADM classification model is evaluated for bias against those profiles.

3.1 Profile-selection based on clustering

Informally, profiles can be seen as relevant descriptive elements that, as a group, act as a "summary" of the data space. Because individuals sharing to an adequate extent similar attributes should intuitively be assigned to the same profile,



(a) (b) (c)

Fig. 1: Schematic overview of the simple clustering algorithm with $k = 4$ (a), the hierarchical clustering algorithm with $k = 2$ and $l = 2$ (b) and a possible outcome of the semi-hierarchical clustering algorithm (c).

clustering can be deemed compatible with a profile selection task. We consider then three different clustering algorithms to implement profile selection: (i) *simple clustering*; (ii) a form of *hierarchical clustering* based on the iteration of the first; and (iii) a novel *semi-hierarchical clustering* method based on adding static and dynamic constraints to control the second. The first two algorithms will be used as baselines to evaluate the third one, and will be described succintly.

The **simple clustering** algorithm consists in the application of the chosen clustering algorithm (in our case, k-medoids, Sec. 2), and results in a flat partition of the sample space (see eg. Fig. 1a). The challenge is to determine the right number of clusters k. **Hierarchical clustering** consists in the nested iteration of the previous algorithm set with k clusters over l layers (see eg. Fig. 1b). The benefit of this method is that the resulting tree-based structure gives further insight on the basis of which feature clusters are created. The downside is that we have to tune an extra hyperparameter besides k: the number of layers l.

The **semi-hierarchical clustering** algorithm is an automatically controlled version of the second algorithm. It is based on performing iteratively two operations. First, we apply a clustering method with $k = 2$, i.e. at every step we divide input subsets into two new subsets (clusters). Then, we test each new subset to decide whether we should continue clustering by looking at the *variability* of the features it expresses, or at its cardinality. For variability of a feature f w.r.t. a dataset D we intend a measure between 0 and 1 that indicates to what extent the feature f is varying within D. A value close to 0 indicates that f is rather stable over D, and a value close to 1 indicates that f mostly varies in D (and so it is not relevant for describing it). The intuition behind using this measure is that stable (ie. not varying) features help to discriminate a profile over another one. For instance, coffees are characterized by a black (or dark brown) colour, so the colour feature is very stable to support discriminating coffee from other drinks; in contrast, colour is not a relevant feature to discriminate books from other objects. In general, any feature can then be: (1) an *irrelevant feature*, when the variability of a feature exceeds an upper bound c_u, is deemed not to be a characteristic property of the cluster; (2) a *relevant feature:* when the variability of a feature is smaller than a lower bound c_l, it means that the feature is strongly characteristic of that cluster. When all features expressed by a subset satisfy either case (1) or case (2), or the subset has less than a fixed amount of elements n_{stop}, there is no need to continue further clustering the input subset, and therefore it is selected as a *profile*. The resulting structure is then a binary tree, possibly unbalanced (see Fig. 1c), whose leaves are the selected profiles. The benefit of semi-hierarchical clustering over simple and hierarchical clustering is that we do not have to decide the numbers of clusters and layers in advance, but requires setting the variability thresholds values c_u and c_l, as well as the threshold cluster cardinality n_{stop}.

In quantitative terms, when *numerical features* are not significant, we expect that their distribution should approximate a uniform distribution. Let us assume we have a numerical non-constant feature X; we can normalize X between 0 and 1 via $(X - X_{min})/(X_{max} - X_{min})$, and then compute the sample standard deviation s from the normalized samples. Theoretically, for a random variable $U \sim Uniform(0,1)$, we have $\mu = 1/2$, and $Var(U) = \mathbb{E}[(U-\mu)^2] = \int_0^1 (x - 1/2)^2 \mathrm{d}x = 1/12$. Thus, the standard deviation of a random variable uniformly distributed is $\sqrt{1/12} \approx 0.29$. Therefore, if the sample standard deviation s approximates 0.29, we can assume that the feature X is uniformly distributed across the given cluster and is therefore not a unique property of species within the cluster. On the other hand, when the standard deviation is close to zero, this means that most sample points are close to the mean. This indicates that feature X is very discriminating for that specific cluster. To obtain a measure of variability, we need to compute the standardized standard deviation $s_s = s/0.29$.

For *categorical features*, we consider the variability measure proposed in [1]. Let X a n-dimensional categorical variable consisting of q categories labelled as $1, 2, ..., q$. The relative frequency of each category $i = 1, ..., q$ is given by $f_i = n_i/n$, where n_i is the number of samples that belongs to category i and $n = \sum_{i=1}^{q} n_i$. Let $\vec{f} = (f_1, f_2, ..., f_q)$ be the vector with all the relative frequencies. We define the variability of X as: $v_q = 1 - ||\vec{f}||_q$. Allaj [1] shows that the variability is bounded by 0 and $1 - 1/q$, where an outcome close to $1 - 1/q$ associates to high variability. We can also compute the standardized variability: $v_{q,s} = \frac{v_q}{1-1/\sqrt{q}}$, such that the variability lies between 0 and 1 for all number of categories q where again a variability close to 1 implies a high variability and hence a non-characteristic feature. A variability close to 0 indicates that the feature is highly characteristic for that sample set.

3.2 Evaluation of clustering methods for profile selection

Given a clustering method, we need to evaluate whether it is working properly with respect to the profile selection task. Unfortunately, since this task is an unsupervised problem, we do not have access to the ground truth, but we can still focus on a related problem: *Is our method stable?* The **stability** of a clustering method towards initialization of the clusters can be tested by running the algorithm multiple times with different initial cluster settings (in our case, different datapoints selected as the initial medoids), and check whether we end up with the same sets of clusters (hereby called *cluster combinations*). When the stability analysis results in two or more different cluster combinations, we want to know how similar these combinations are, and for doing this, we will introduce a measure of inter-clustering similarity.

Inter-clustering similarity is a similarity score that tells to what extent different outcomes of clustering are similar, by comparing how many elements clusters belonging to the two clustering outputs have in common with each other. This score can be computed by comparing the distribution of the elements over the clusters of two combinations. We present the process through an example:

Example: Let us consider a dataset with 20 data points. Suppose the clustering algorithm returns the following two different cluster combinations C^1 and C^2: where $C^1 = (c_1^1, c_2^1, c_3^1) = ((1,4,6,7,13,17,18,20),(3,5,11,12,15,16),(2,8,9,10,14,19))$, and $C^2 = (c_1^2, c_2^2, c_3^2) = ((1,4,6,7,13,17,18,20),(3,11,12,14,15,16,19),(2,5,8,9,10))$. For every cluster c_i^1 in C_1 we compute its overlap with each cluster c_j^2 in C_2. For instance, for the first cluster of C^1, $c_1^1 = (1,4,6,7,13,17,18,20)$ the max overlap is 1, since c_2^1 of C^2 is exactly the same. For the second cluster c_2^1 we have:

$$\max\Big(\frac{|c_2^1 \cap c_1^2|}{|c_2^1|}, \frac{|c_2^1 \cap c_2^2|}{|c_2^1|}, \frac{|c_2^1 \cap c_3^2|}{|c_2^1|}\Big) = \max(0/6, 5/6, 1/6) = 0.83$$

Applying the same calculation on c_3^1 returns 0.67. The similarity score of cluster-combination C^1 with respect to cluster-combination C^2 is given by the mean over all the three maximum overlap values, which in this case is 0.83.

3.3 Profile-based evaluation of a given classifier

By means of profiles, we can assess the *historical*, *representational*, *aggregation* and *evaluation biases* (Sec. 2) of a certain ADM classification model.

Historical bias arises on the dataset used for training. We measure it using conditional demographic disparity (Def. 2); however, the resulting value may be wrong if an attribute r is not a relevant characteristic for grouping individuals. Therefore, we consider a *profile-conditioned demographic disparity*, differing from [12, 18] in as much each profile label c_i acts as attribute r. In this way we capture behavioural attitudes (w.r.t. assigning or not an advantageous label) of the labeller-oracle towards elements belonging to that profile.

For **Representational bias**, by clustering the dataset around profiles, we may get insights if there are parts of the population which are overrepresented and parts which are underrepresented. A domain expert can compare the resulting distribution of profiles with the expected distribution for the population on which the decision-making algorithm is going to be used. When these two distributions differ much from each other, one can *bootstrap sampling* from profiles to get a more correct distribution in the training dataset. If no domain expertise is available, one can consider using this method during deployment. By collecting the data where the algorithm is applied on, once the dataset is sufficiently large (eg. about the size of the training dataset), one can divide the samples of the collected dataset over the different known profiles based on distance towards the medoids associated to profiles. If the distribution of the collected dataset over the profiles relevantly differs from the distribution of the training dataset, we can conclude that there is representation bias. Alternatively, one can repeat the profile selection on the collected data, and evaluate how much they differ from the ones identified in the training dataset.

In order to evaluate **Aggregation bias**, we need a good metric to evaluate the performance of the trained model under assessment. Our goal is to evaluate the model against all profiles, i.e. to test whether the model works equally well on individuals with different profiles (i.e. individuals from different clusters). We start from the definition of equal opportunity (Def. 1), but we reformulate it in a way that equal opportunity is computed with respect to profiles instead of the protected class:

Definition 3 (Equal opportunity w.r.t. a single profile). *We say a binary predictor $\hat{y}$ satisfies equal opportunity with respect to a profile C equal to i and outcome/ground truth y iff:* $Pr\{\hat{y} = 1|C = i, y = 1\} = Pr\{\hat{y} = 1|C \neq i, y = 1\}$

Definition 4 (General equal opportunity). *A binary predictor $\hat{y}$ satisfies general equal opportunity iff it satisfies equal opportunity with respect to all profiles $C \in \{1, .., k\}$ and ground truth y. In formula:* $Pr\{\hat{y} = 1|C = 1, y = 1\} = \ldots = Pr\{\hat{y} = 1|C = k, y = 1\}$

In some cases, it might be that getting a wrong prediction for a ground truth or positive outcome label occurs more often than with a negative outcome label, and may be more valuable (eg. in some medical disease treatment); looking at distinct values of y may give insights on the overall functioning of the model.

Evaluation bias arises when the model is evaluated on a population which differs in distribution from the data that was used for training the model. It has been shown [8] that fairness-preserving algorithms tend to be sensitive to fluctuations in dataset composition, i.e. the performance of the algorithms is affected by the train-test split. To ensure that we do not have evaluation bias, we run a Monte Carlo simulation of the decision-making algorithm. This means that we make M different train-test splits of the dataset. For each train-test split, we train the decision-making algorithm on the train set and use the test set for evaluation. For the model-evaluation, we use general equal opportunity (Def. 4). This gives us insights on which profiles are more sensitive towards train-test splitting (and thus to evaluation bias).

4 Experiments and results

We conducted two experiments to evaluate PEBAM.[2] In the first, we considered a small artificial dataset to test the processing pipeline in a context in which we knew the ground truth. In the second, we focused on the German credit dataset, used in multiple researches on fairness [8, 7, 11, 13]. This dataset contains 8 attributes (both numerical and categorical) of 1000 individuals, including an association of each individual to a credit risk score (good or bad). For the sake of brevity, we will limit our focus here on the German credit dataset. All Python-code that we run for obtaining the results is publicly available.[3]

[2] Experimental setup: Intel Core i7-10510u, 16 GB RAM, Windows-10 64-bit.

[3] `https://github.com/mcwilms/PEBAM`

algorithm	l	k	# clusters	# cluster comb.	freq. most occurring comb.	running time
simple	1	20	20	15	43	3.956 ± 0.376 s
simple	1	30	30	50	9	8.170 ± 0.777 s
simple	1	40	40	17	26	12.068 ± 0.875 s
hierarchical	2	4	16	31	24	1.119 ± 0.078 s
hierarchical	2	5	25	21	28	1.152 ± 0.077 s
hierarchical	2	6	36	18	32	1.298 ± 0.113 s
hierarchical	3	3	27	54	12	1.144 ± 0.047 s
hierarchical	4	2	16	18	29	1.450 ± 0.083 s
hierarchical	5	2	32	39	20	1.614 ± 0.071 s
semi-hierarchical	-	-	34^*	47	23	2.264 ± 0.112 s

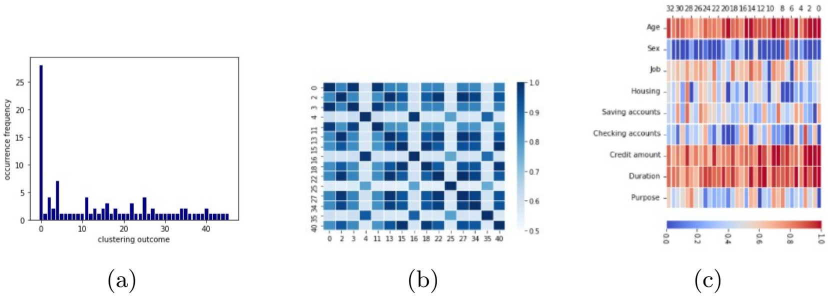
Table 1: Stability analysis of the cluster combinations for all three clustering algorithms applied on the German credit data set with varying parameter settings. *For the semi-hierarchical clustering the number of clusters is not fixed, we reported the number of clusters for the cluster combination that occurs most.

4.1 Evaluation of clustering for profile selection

We evaluate the three clustering algorithm for profile selection specified in Sec. 3.1 following the method described in Sec. 3.2. For all three clustering algorithms, we used a k-medoids clustering algorithm with the Gower distance [3, 9]. The simple algorithm and the hierarchical clustering require the tuning of hyperparameters as k (number of clusters), and l (number of layers), and therefore have a fixed number of final clusters (in this context seen as profiles). Since we do not know the correct number of profiles in advance, we tried several hyperparameters. The semi-hierarchical clustering algorithm needs instead three other parameters: c_u, c_l (upper and lower bounds of variability), and n_{stop} (the threshold for cluster cardinality). For our experiments, we chose to set $c_u = 0.9$, $c_l = 0.1$, and n_{stop} to 5% of the size of the dataset (ie. 50 for the German credit dataset).

As a first step, we test if clustering methods are *stable* enough. Table 1 gives a summary of the stability analysis performed for all three clustering methods on the German credit dataset, reporting (when relevant) the parameters l and c k, the number of clusters (eg. k^l), the number of different outcomes after running the clustering algorithm 100 times with different random initializations, how often the most occurring cluster combinations occurs in these 100 runs, and the mean running time. Amongst other things, Table 1 shows that, when running the stability analysis with semi-hierarchical clustering, the German credit dataset produces 47 different cluster combinations. However, several combinations occur only once, and only one of the combination (number 0, the first one) occurs significantly more often than the other combinations, see Fig. 2a.

As a second step, we compute the *inter-clustering similarity* to test if the different cluster combinations are adequately similar. Fig. 2b shows the inter-clustering similarity of the different cluster combinations we obtain on the German credit dataset via the semi-hierarchical clustering algorithm, showing only the cluster combinations that occur more than once. A dark blue tile means that two clusters are very similar (max 100 percent), and a white tile means that they have 50 percent or less of the clusters in common.



(a) (b) (c)

Fig. 2: Stability and variability analysis of the semi-hierarchical clustering algorithm applied on the German credit dataset: (a) frequency of each clustering outcome (or cluster combination) obtained over 100 runs; (b) inter-clustering similarity for cluster combinations that occur more than once; (c) variability of the different features for each profile in the most recurrent cluster combination.

As a confirmation that the algorithms end up on profiles which are descriptive attractors, we compute the feature *variability* of each cluster (supposedly a profile) within the most occurring clustering outcome. The variability plot of Fig. 2c shows for each profile (columns) the variability of the features (rows), where dark blue indicates a low variability and dark red a high variability. In tendency, qualitative features becomes stable, whereas numerical features show still a certain variability at the level of profiles. For each profile, however, the majority of features becomes stable.

4.2 Profile-based Evaluation of Bias

We now apply the bias evaluation methods described in Sec. 3.3 with the profiles obtained by applying the semi-hierarchical clustering algorithm on classifiers trained on the German credit data set via three commonly used machine learning algorithms: *logistic regression classifier*, *XGboost classifier*, and *support vector machine (SVM) classifier* (eg. [2]).[4] Following the standard practice of removing protected attributes (gender, ethnicity, etc.) as input features during training, we do not use the feature "Sex" provided by the German credit dataset for training the classifiers.

For **Representational bias**, Fig. 3 gives an overview of the presence of the identified profiles within the German credit dataset. We see that not all profiles are equally frequent; this is not necessarily an error, as long as this profile distribution is a good representation of the data on which ADM will be applied in practice. Expert knowledge or actual data collection can be used to test this assumption.

To assess **Historical bias**, we first test for (general) *demographic disparity* with respect to protected attributes. A disadvantaged group with attribute x

[4] Note however that the same approach will apply with any choice of profile selection method or of ML method used to train the classifier.

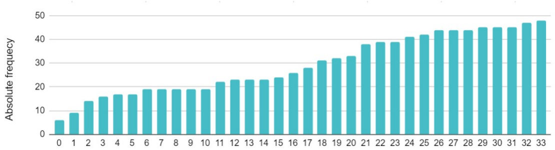


Fig. 3: Absolute frequencies of profiles obtained after performing the semi-hierarchical clustering algorithm on the German credit dataset.

(DG_x) is the group with risk-label 'bad' and attribute x, whereas an advantaged group with attribute x (AG_x) is the group with risk-label 'good' and attribute x. Denoting with A the proportion of people from the full dataset belonging to the protected class (female) in the advantaged group over all people (female and male) in the advantaged group, and D for the proportion of people belonging to the protected class in the disadvantaged group over all people, we find:

$$D = \frac{\#DG_f}{\#DG_{f+m}} = 0.36 > 0.29 = \frac{\#AG_f}{\#AG_{f+m}} = A$$

and hence we conclude that, at aggregate level, the German credit data exhibit demographic disparity. We now will do the same computation for each profile subset of the German credit dataset, to test whether there is *demographic disparity for profiles*. We write A_c for the proportion of people belonging to the protected class (female) in the advantaged group of cluster c over all people in the advantaged group of cluster c, and D_c for the proportion of people belonging to the protected class in the disadvantaged group of cluster c over all disadvantaged people in cluster c. Table 2 shows that there are 3 profiles (7, 20 and 28) that show demographic disparity. The fact that profiles show demographic disparity indicates that it might be possible that for some profiles other (not-protected) attributes correlate with the protected attribute, and so the protected attribute can indirectly be used in the training-process of the model.

In the computation of the (dis)-advantage fraction of Table 2 we still looked at the protected group *female*, however, we can also compute the measures A^*_c (D^*_c) as the fraction of (dis)advantaged individuals in a profile c over the total individuals within that profile (without distinguishing the protected class in it):

$$A^*_c = \frac{\#AG_c}{\#AG_c + \#DG_c} \qquad D^*_c = \frac{\#DG_c}{\#AG_c + \#DG_c}$$

By doing so, we get an indication of how informative a profile is for belonging to the (dis)advantaged group. Table 3 shows the fraction of advantaged and disadvantaged individuals for each profile. Note that there are profiles for which the majority of the samples is clearly advantaged (eg. 0, 1, 2, ...), a few have some tendency towards disadvantaged outcomes (eg. 3, 15), but in comparison could be put together with other profiles that have no clear majority (eg. 9, 10, ...). Plausibly, for profiles exhibiting a mixed distribution of the risk label, there may be factors outside the given dataset that determine the label. Since

c	D_c	A_c	c	D_c	A_c	c	D_c	A_c	c	D_c	A_c	c	D_c	A_c
0	0.00	0.00	**7**	**0.36**	**0.25**	14	0.00	0.19	21	0.00	0.03	**28**	**0.96**	**0.90**
1	0.00	0.00	8	0.00	0.00	15	1.00	1.00	22	0.00	0.00	29	0.00	0.00
2	0.00	0.00	9	0.00	0.00	16	1.00	1.00	23	0.00	0.00	30	1.00	1.00
3	0.00	0.00	10	1.00	1.00	17	1.00	0.86	24	0.00	0.00	31	1.00	1.00
4	0.00	0.07	11	0.00	0.00	18	0.00	0.00	25	0.00	0.03	32	0.00	0.00
5	1.00	1.00	12	0.00	0.00	19	0.06	0.06	26	1.00	1.00	33	0.00	0.09
6	0.00	0.08	13	0.00	0.00	**20**	**0.24**	**0.00**	27	0.00	0.00			

Table 2: Fractions of disadvantaged (D) and advantaged (A) individuals with protected attribute *female* in each profile c.

c	D^*	A^*	c	D^*	A^*	c	D^*	A^*	c	D^*	A^*	c	D^*	A^*
0	0.00	1.00	**7**	**0.58**	**0.42**	14	0.09	0.91	21	0.05	0.95	**28**	**0.52**	**0.48**
1	0.11	0.89	**8**	**0.58**	**0.42**	**15**	**0.62**	**0.38**	**22**	**0.36**	**0.64**	**29**	**0.40**	**0.60**
2	0.14	0.86	**9**	**0.47**	**0.53**	**16**	**0.38**	**0.62**	23	0.15	0.85	30	0.09	0.91
3	**0.62**	**0.38**	**10**	**0.53**	**0.47**	17	0.25	0.75	24	0.37	0.63	31	0.22	0.78
4	0.12	0.88	11	0.18	0.82	18	0.19	0.81	25	0.20	0.71	32	0.13	0.87
5	0.12	0.88	12	0.35	0.65	**19**	**0.50**	**0.50**	**26**	**0.45**	**0.55**	33	0.06	0.94
6	0.32	0.68	**13**	**0.52**	**0.48**	**20**	**0.52**	**0.48**	27	0.14	0.86			

Table 3: Fractions of disadvantaged (D^*) and advantaged (A^*) individuals in each profile c.

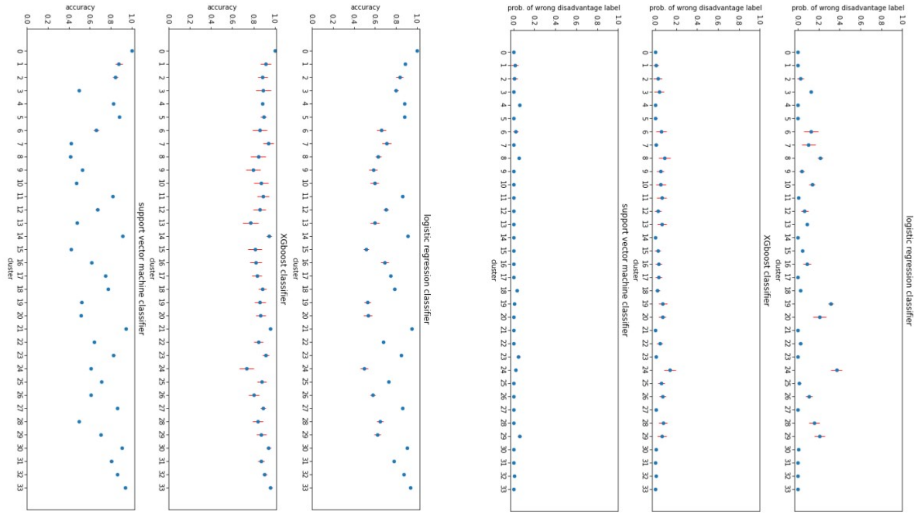
the ADM models also do not have access to these external features, it may be relevant to evaluate performance on these profiles to evaluate this hypothesis.

For the **Aggregation bias** we look at the blue dots in Fig. 4a, which indicate the mean performances of the algorithm over training the algorithm 100 times on different train-test splits. Looking at performance over each profile gives us a visual way to see to what extent general equal opportunity (Def. 4) is satisfied; we consider the average to provide a more robust indication. We see that the XGboost classifier performs the best of the three algorithms with respect to predicting the labels correctly, however we also observe some difference in performance depending on profile. In contrast, the SVM classifier has very low probabilities of getting an unjustified disadvantage label (Fig. 4b), while the probability of getting a correct label is not very high.

For the **Evaluation bias**, we look at the performance ranges of the different classification methods (visualized in terms of standard deviations). We see that the SVM classifier is the least sensitive towards the train-test split. The logistic regression classifier is already slightly more sensitive, however the XGboost classifier is by far the most sensitive towards the train-test split. All three algorithms are equally sensitive towards small profiles as much as larger profiles.

5 Conclusion

The paper introduced PEBAM: a new method for evaluating biases in ADM models trained on mixed datasets, focusing in particular on profiles extracted through a novel (semi-hierarchical) clustering method. Although we have proven the feasibility of the overall pipeline, several aspects need further consolidation, as for instance testing other measures of variability (eg. to be compared with



(a)

(b)

Fig. 4: (a) Probability of getting a correct prediction label. (b) Probability of getting a disadvantage ('bad') label when the true label is the advantage ('good'), for *logistic regression classifier*, *XGboost classifier*, and *SVM classifier* on the different profiles.

entropy-based forms of clustering, eg. [6]), similarity scores, and distance measures. Yet, the method was already able to find biases that were not revealed by most used bias evaluation methods, since they would not test for biased decisions against groups of individuals that are regrouped by non-protected attributed values only. For instance, profile 7, exhibiting demographic disparity against women as historical bias, refers to applicants with little saving/checking accounts and renting their house, who are asking credit for cars (see Appendix for details). Why, *ceteris paribus* (all other things being the same), men are preferred to women for access to credit for buying cars, if not in presence of a prejudice?

At a technical level, although the proposed semi-hierarchical clustering algorithms has shown a shorter running time than the baseline on the German credit dataset, the PAM algorithm does not scale well to larger datasets. Tiwari et al. propose BanditPAM as alternative for PAM [17], a method that reduces the complexity of each PAM iteration from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$. When using PEBAM on large datasets one might consider using BanditPAM over PAM. This will be investigated in future work.

Acknowledgments Giovanni Sileno was partly funded by the Dutch Research Council (NWO) for the HUMAINER AI project (KIVI.2019.006).

References

1. Allaj, E.: Two simple measures of variability for categorical data. Journal of applied statistics **45**(8), 1497–1516 (2018)
2. Belle, V., Papantonis, I.: Principles and practice of explainable machine learning. Frontiers in Big Data **4** (2021)
3. Ben Ali, B., Massmoudi, Y.: K-means clustering based on Gower similarity coefficient: A comparative study. In: 2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO). pp. 1–5. IEEE (2013)
4. Budiaji, W., Leisch, F.: Simple k-medoids partitioning algorithm for mixed variable data. Algorithms **12**(9), 177– (2019)
5. Caruso, G., Gattone, S., Fortuna, F., Di Battista, T.: Cluster analysis for mixed data: An application to credit risk evaluation. Socio-economic planning sciences **73** (2021)
6. Cheng, C.H., Fu, A.W., Zhang, Y.: Entropy-based subspace clustering for mining numerical data. In: Proceedings of the 5th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. p. 84–93. KDD '99, Association for Computing Machinery, New York, NY, USA (1999)
7. Feldman, M., Friedler, S., Moeller, J., Scheidegger, C., Venkatasubramanian, S.: Certifying and removing disparate impact. In: Proceedings of the 21th ACM SIGKDD International Conference on knowledge discovery and data mining. pp. 259–268. KDD '15, ACM (2015)
8. Friedler, S., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E., Roth, D.: A comparative study of fairness-enhancing interventions in machine learning. In: Proceedings of the Conference on fairness, accountability, and transparency. pp. 329–338. FAT '19, ACM (2019)
9. Gower, J.C.: A general coefficient of similarity and some of its properties. Biometrics **27**(4), 857–871 (1971)
10. Hardt, M., Price, E., Price, E., Srebro, N.: Equality of opportunity in supervised learning. In: Advances in Neural Information Processing Systems (NIPS) (2016)
11. Kamiran, F., Calders, T.: Classifying without discriminating. In: Proc. 2nd IEEE Int. Conf. on Computer, Control and Communication (2009)
12. Kamiran, F., Žliobaitė, I., Calders, T.: Quantifying explainable discrimination and removing illegal discrimination in automated decision making. Knowledge and Information Systems **35**(3), 613–644 (2013)
13. Kamishima, T., Akaho, S., Sakuma, J.: Fairness-aware learning through regularization approach. In: 2011 IEEE 11th International Conference on Data Mining Workshops. pp. 643–650. IEEE (2011)
14. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. ACM Comput. Surv. **54**(6) (jul 2021)
15. Pleis, J.: Mixtures of Discrete and Continuous Variables: Considerations for Dimension Reduction. Ph.D. thesis, University of Pittsburgh (2018)
16. Suresh, H., Guttag, J.: A framework for understanding sources of harm throughout the machine learning life cycle. In: Equity and Access in Algorithms, Mechanisms, and Optimization. EAAMO '21 (2021)
17. Tiwari, M., Zhang, M.J., Mayclin, J., Thrun, S., Piech, C., Shomorony, I.: Banditpam: Almost linear time k-medoids clustering via multi-armed bandits. In: Advances in Neural Information Processing Systems (NIPS) (2020)
18. Wachter, S., Mittelstadt, B., Russell, C.: Why fairness cannot be automated: Bridging the gap between EU non-discrimination law and AI. Computer Law & Security Review **41** (2021)

A Profiles on the German credit dataset

The following table reports the profiles selected on the German credit dataset by applying the semi-hierarchical clustering proposed in the paper, as described by their medoids:

Profile	Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Purpose	Sample
0	26	male	2	rent	moderate	unknown	3577	9	car	859
1	37	male	2	own	unknown	unknown	7409	36	business	868
2	39	male	3	own	little	unknown	6458	18	car	106
3	26	male	2	own	little	little	4370	42	radio/TV	639
4	31	male	2	own	quite rich	unknown	3430	24	radio/TV	19
5	38	female	2	own	unknown	unknown	1240	12	radio/TV	135
6	43	male	1	own	little	little	1344	12	car	929
7	36	male	2	rent	little	little	2799	9	car	586
8	39	male	2	own	little	little	2522	30	radio/TV	239
9	31	male	2	own	little	moderate	1935	24	business	169
10	33	female	2	own	little	little	1131	18	furniture/equipment	166
11	26	male	1	own	little	moderate	625	12	radio/TV	220
12	23	male	2	own	unknown	moderate	1444	15	radio/TV	632
13	42	male	2	own	little	little	4153	18	furniture/equipment	899
14	29	male	2	own	unknown	unknown	3556	15	car	962
15	37	female	2	own	little	moderate	3612	18	furniture/equipment	537
16	27	female	2	own	little	little	2389	18	radio/TV	866
17	26	female	2	rent	little	unknown	1388	9	furniture/equipment	582
18	29	male	2	own	little	unknown	2743	28	radio/TV	426
19	53	male	2	free	little	little	4870	24	car	4
20	36	male	2	own	little	little	1721	15	car	461
21	38	male	2	own	little	unknown	804	12	radio/TV	997
22	29	male	2	own	little	moderate	1103	12	radio/TV	696
23	43	male	2	own	unknown	unknown	2197	24	car	406
24	27	male	2	own	little	little	3552	24	furniture/equipment	558
25	30	male	2	own	little	moderate	1056	18	car	580
26	24	female	2	own	little	moderate	2150	30	car	252
27	34	male	2	own	little	unknown	2759	12	furniture/equipment	452
28	24	female	2	rent	little	little	2124	18	furniture/equipment	761
29	34	male	2	own	little	moderate	5800	36	car	893
30	34	female	2	own	little	unknown	1493	12	radio/TV	638
31	30	female	2	own	little	unknown	1055	18	car	161
32	35	male	2	own	little	unknown	2346	24	car	654
33	35	male	2	own	unknown	unknown	1979	15	radio/TV	625