Information Sciences 581 (2021) 155–178

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Fair Clustering with Fair Correspondence Distribution

Woojin Lee [c], Hyungjin Ko [a], Junyoung Byun [a], Taeho Yoon [b], Jaewook Lee [a,*]

[a] Industrial Engineering, Seoul National University, 1 Gwanakro, Gwanak-gu, Seoul 08826, Republic of Korea
[b] Mathematical Sciences, Seoul National University, 1 Gwanakro, Gwanak-gu, Seoul 08826, Republic of Korea
[c] School of AI Convergence, Dongguk University-Seoul, 30 Pildong-ro 1-gil, Jung-gu, Seoul 04620, South Korea

ARTICLE INFO

Article history:
Received 31 May 2021
Received in revised form 29 July 2021
Accepted 5 September 2021
Available online 8 September 2021

Keywords:
Fair clustering
Support vector clustering
Fair distribution

ABSTRACT

In recent years, the issue of fairness has become important in the field of machine learning. In clustering problems, fairness is defined in terms of consistency in that the balance ratio of data with different sensitive attribute values remains constant for each cluster.

Fairness problems are important in real-world applications, for example, when the recommendation system provides targeted advertisements or job offers based on the clustering result of candidates, the minority group may not get the same level of opportunity as the majority group if the clustering result is unfair. In this study, we propose a novel distribution-based fair clustering approach. Considering a distribution in which the sample is biased by society, we try to find clusters from a fair correspondence distribution. Our method uses the support vector method and a dynamical system to comprehensively divide the entire data space into atomic cells before reassembling them fairly to form the clusters. Theoretical results derive the upper bound of the generalization error of the corresponding clustering function in the fair correspondence distribution when atomic cells are connected fairly, allowing us to present an algorithm to achieve fairness. Experimental results show that our algorithm beneficially increases fairness while reducing computation time for various datasets.

1. Introduction

With machine learning increasingly influencing decisions on important aspects of our lives, such as advertising, employment, and criminal justice, the demand for a fair machine learning model has also increased rapidly in recent years. While there is little doubt regarding the effectiveness and versatility of machine learning algorithms, we are increasingly focusing on the objectivity of algorithms. It requires not only high performance but also impartiality toward specified sensitive attributes in a population, such as sex, gender, or race.

A misconception about machine learning is that it is absolutely impartial. However, as machine learning relies on training data, algorithms may exhibit biases toward specific demographic groups because of biased samples. For example, facial recognition shows a better accuracy on white, male faces [9], and a model for criminal recidivism is likely to show a much higher false positive rate for black defendants than white defendants [1]. Additionally, gender stereotypes are reflected in word2vec embeddings [8].

* Corresponding author.
E-mail address: jaewook@snu.ac.kr (J. Lee).

https://doi.org/10.1016/j.ins.2021.09.010
0020-0255/

These fairness problems occur because the data provided by humans can be highly biased. The data can be biased for several reasons, such as those listed in [5]. First, labeling issues can cause data to be biased. Labeling is the process of manually assigning training data to class labels. For example, recruitment managers evaluate job seekers' capabilities and use them as training data. By using these data for training, a machine learning model might replicate the prejudices existing in a manager's decisions. Bias can also occur during data collection. Machine learning models trained from partial or non-representative data might discriminate against a sensitive group. For instance, if a single sample has an inappropriate representation of a protected class, the results might be biased against the protected class. Moreover, if the sample size of a protected group is much smaller than that of the other group, the model might perform appropriately in the minority group. Even when a sensitive attribute is removed from the training data, the model can discriminate against the protected group because different features might contain some information about sensitive attributes.

Anti-discrimination laws evaluate the fairness of a decision-making process using disparate treatment and disparate impact. A disparate treatment means that the decisions are based on sensitive attributes. In the case of machine learning, sensitive features should not be included in the training data. This notion of fairness is also called "unawareness." However, it has a limitation in that the sensitive variables can be highly related to other features. While disparate treatment focuses on machine learning input, disparate impact focuses on the output. A model has a disparate impact if its results discriminate against a protected group [13]. This criterion of fairness is also called "demographic parity" or "statistical parity" and it is widely used in the machine learning field.

To achieve demographic parity, a decision should be independent of sensitive attributes. For example, if the sensitive attribute represents gender in an employment problem, then males and females should have the same probability of being hired. Some studies argue that this condition is related to the "four-fifth rule" in disparate impact law [13]. However, this definition ignores a possible relationship between the ground truth label and sensitive attributes. To overcome the drawbacks of statistical parity, several studies [15,31,25] have focused on an alternative criterion called "equalized odds," also known as "separation." This criterion conditions a metric on the ground truth label. This requires equal false positive and false negative between the different groups.

Previous research on fair machine learning has focused primarily on supervised problems such as classification[15,30] and regression. Sensitive attributes are not explicitly used in making decisions; decisions that discriminate against protected classes should not be made. However, an unfair unsupervised learning can also cause problems in the real world. For example, when a recommendation system provides targeted advertisements or job offers based on the clustering result of candidates, the minority group may not get the same level of opportunity as the majority group if the clustering result is unfair. In this study, we examine the fairness problem in clustering, which focuses on preserving the balance of each cluster. A seminal study [11] first introduced the *balance* concept as a criterion for fair clustering, defining a clustering to be fair if the resulting clusters share a common ratio of data points representing individuals belonging to groups of sensitive attributes. In addition, they suggested the use of polynomial-time algorithms to obtain approximately balanced clustering by first finding an approximately optimal decomposition of a dataset into fairlets, which are minimal subclusters with the exact value of an intended balance. They then applied k-means or k-median algorithms to the set of fairlets. A majority of follow-up studies since then have taken a combinatorial approach to search for clustering with a balance closer to the desired level [7,16]. However, the computational complexity of such approaches are prohibitive unless both the number of clusters and the data dimensions are very small, even with the scalable algorithms in [3]. This seems to be primarily because of the NP-hardness when computing a fair clustering, which is truly optimal in terms of k-center costs [11]. Experiments have also shown that k-center costs tend to increase substantially to control the balance measure, and it is difficult to assess the quality of clustering results.

In this study, we propose a novel approach to solve the problem: we no longer view fair clustering as a combinatorial optimization problem, but instead take a distribution-based approach. A distribution-based approach assumes that a real dataset is generated from a probability distribution with a compactly supported density function p, and that each connected component of the sup-level set $\{\mathbf{x} : p(\mathbf{x}) \geqslant r\}$ represents a meaningful cluster. Distribution-based methodologies have been successful in clustering [6,22,19,4,10,14,24,12] as well as in performing other tasks, including classification [20] and denoising [17,18]. In this work, we use a similar framework. First, we approximate the underlying density function, and then use the presumed density to identify the corresponding clusters. Despite these similarities, this work has several significant novelties, as summarized below.

We first define *fair distribution* by a probability distribution such that for any Borel set $B \in \mathbb{R}^d$, the value of $P(X \in B | X \in G_i)$ is the same (or almost the same allowed by adopted fairness standards) for all $i = 1, \cdots, \ell$, where $G_1, \cdots, G_\ell$ denote the groups of sensitive attributes. We assume that real data are generated from a distribution that has deviated from a fair correspondence distribution because of socioeconomic biases, and propose a new idea that a fair clustering can be obtained by tracing this fair correspondence distribution. Second, although it is difficult to directly approximate a fair density function, we show that detecting clusters corresponding to a fair correspondence distribution can be reduced to a discrete optimization problem through a theoretical analysis. This is done by restricting the location of modes in the search for a fair correspondence distribution, and then providing a generalization error bound on the proposed cluster assignment rule based on a finite set of modes. The development of this theory is based on Rademacher complexity and $\mathcal{H}$-divergence, which have never been used in the related literature. Third, we provide a flexible framework for fair clustering where a user can control to what extent the fairness constraint should be reflected in the resulting clusters. This implies that the method we suggest, unlike any pre-

existing fair clustering algorithms, satisfies the *correspondence principle*, that is, the algorithm would still successfully detect clusters based on distribution even if the fairness constraint is removed.

The contribution of our paper is threefold:

1. We first propose a novel distribution-based approach to a fair clustering problem. We suggest the concept of *fair correspondence distribution* and try to find fair clusters from this distribution.
2. We provide a theoretical background that derives the upper bound of the generalization error of the corresponding clustering function.
3. We present an algorithm to achieve fair clustering results based on the support vector method and a dynamical system. Experimental results show that the suggested method can successfully obtain fair clustering results in various datasets.

Up to the proposed methodology section, we develop some necessary theoretical constructs and explain our algorithm in more detail. Then, in the next section, we experimentally show the proposed algorithm's adequate performance on both synthetic and real datasets, with a significantly relaxed time consumption when compared to the state-of-the-art fairlet based methods.

2. Preliminaries

2.1. Fair clustering in terms of balance

Suppose we have N data points to be assigned to K different clusters $\mathfrak{C} = \{C_1, C_2, \ldots, C_K\}$. Each data point $\mathbf{x}_i$ has a sensitive attribute whose value may be, say, red or blue. The balance of a subset $S \subset D = \{\mathbf{x}_i\}_{i=1}^N$ is

$$\text{balance}(S) = \min\left\{\frac{|S_r|}{|S_b|}, \frac{|S_b|}{|S_r|}\right\},$$

where S_b and S_r denote the set of red and blue points in S, respectively. Then, the overall balance of the clustering is defined as

$$\text{balance}(\mathfrak{C}) = \min_{C \in \mathfrak{C}} \text{balance}(C).$$

Since the balance of a clustering is determined by taking the minimum over balances of all clusters, it is important to control the balance of the most biased cluster. Assuming $C_b \leqslant C_r$, the balance of the clustering is $\frac{|C_b|}{|C_r|}$; this indicates that every cluster $C \in \mathfrak{C}$ has a balance of at least $\frac{|C_b|}{|C_r|}$. [11] suggested the concepts of *balance* and used it as a criterion for fair clustering. We follow the literature and consider that the higher this measure is for each clustering result, the fairer is the clustering.

The purpose of fair clustering is to find $\mathfrak{C}$ that solves the following problem:

$$\begin{aligned} &\text{minimize} && F(\mathfrak{C}) \\ &\text{subject to} && \text{balance}(\mathfrak{C}) = \text{balance}(D), \end{aligned}$$

where F denotes the clustering cost function.

Previous approaches to fair clustering have attempted to solve this fairness constrained problem using combinatorial approach such as finding fairlets [11,3] or coresets [16]. However, these approaches entail a substantial computational cost to find an approximately optimal fairlet decomposition. In addition, in these approaches, a fairness constraint is achieved at the expense of the clustering objective.

In contrast to the previous methods focused on a combinatorial optimization approach, our method employs a fair correspondence distribution to solve the fair clustering problem, which is detailly described in the next section.

2.2. Multi-basin system and atomic cells

Given a dataset $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \subset \mathcal{X} = \mathbb{R}^d$, we introduce the notion of a *support density function*, of which a sup-level set well describes the support of the data distribution.

Definition 1. We define a *support density function* as a positive density function $p : \Re^d \to \Re^+$ where a level set of p for some $r > 0$ can be decomposed into several disjoint connected sets as

$$L_p(r) = \{\mathbf{x} \in \Re^n : p(\mathbf{x}) \geqslant r\} = C_1 \cup \cdots \cup C_{K_p(r)} \tag{1}$$

where $C_i, i = 1, \ldots, K_p(r)$ are disjoint connected sets corresponding to different clusters and $K_p(r)$ is the number of clusters determined by p and r.

Given a support density function p corresponding to the decision function, we consider the following generalized positive gradient system

$$\frac{d\mathbf{x}}{dt} = F(\mathbf{x}) = R(\mathbf{x})\nabla p(\mathbf{x}),$$

where $R(\mathbf{x})$ is a Riemannian metric on $\mathcal{X}$, or a smooth assignment of a symmetric, positive definite matrix over the sample space. This system is known to be *complete*, that is given any initial condition $\mathbf{x}(0) = \mathbf{x_0}$, it can be guaranteed that the fixed point $\mathbf{x}(t)$ at iteration t is uniquely determined for all $t \in \mathbb{R}$.

A point at which F vanishes is called an *equilibrium vector* (EV) of the system. An EV $\mathbf{x} \in \mathbf{R}^d$ such that the Jacobian $J_F(\mathbf{x})$ of F at $\mathbf{x}$ is positive-definite is called a *stable equilibrium vector* (SEV). If $\mathbf{x}$ is an EV such that $J_F(\mathbf{x})$ has precisely k negative eigenvalues and $d-k$ positive eigenvalues, then it is called an *index-k EV*. Given a SEV, $\mathbf{s}$, we consider its *basin of attraction* $B(\mathbf{s}) = \{\mathbf{x}(0) \in \mathbb{R}^d : \lim_{t\to\infty}\mathbf{x}(t) = \mathbf{s}\}$, which is the set of all initial conditions that converge to $\mathbf{s}$ under the negative gradient system. It is known that the system is *completely stable* under weak assumptions, that is, the data space $\mathbb{R}^d$ partitions into basins of attractions as follows:

$$\mathbb{R}^d = \bigcup_{i=1}^{M} \overline{B(\mathbf{s}_i)} \tag{2}$$

where $V = \{\mathbf{s}_1, \cdots, \mathbf{s}_M\}$ are the SEVs and $\overline{S}$ denotes the closure of a set S. [23].

As mentioned above, we aim to identify the connected components of the support of data distribution. This is achieved by identifying the SEVs and establishing a proper notion of adjacency between them. We refer to each basin $B(\mathbf{s}_i)$ as an "atomic cell", which is a sub-cluster of points tightly grouped around the modes of data distribution. We claim that these points should not be separated into different clusters because the points in close proximity to a mode are homogeneous and clearly reflect the intrinsic distribution structure. Hence, we search for a clustering such that each atomic cell belongs to the same cluster.

We consider two SEVs, $\mathbf{s}_a$ and $\mathbf{s}_b$, to be *adjacent* to each other if $\overline{B(\mathbf{s}_a)} \cap \overline{B(\mathbf{s}_b)} \neq \varnothing$. Note that there exists a unique one-dimensional curve from $\mathbf{s}_a$ to $\mathbf{s}_b$ climbs up and then down the hill via an index-one EV, the so-called a *transition equilibrium vector* (TEV), $\mathbf{t} \in \partial B(\mathbf{s}_a) \cap \partial B(\mathbf{s}_b)$. With this definition, we construct a graph $G_r = (V_r, E_r)$ truncated at level r as in [23], where

1. The vertex set V_r consists of $\mathbf{s}_i \in V$ with $p(\mathbf{s}_i) > r$.
2. The edge set E_r consists of pairs $(\mathbf{s}_i, \mathbf{s}_j)$ of adjacent SEVs such that the corresponding TEV $\mathbf{t}$ satisfies $p(\mathbf{t}) > r$.

It is known [21] that under some weak assumptions, there exists a threshold level γ for which the graph G_r is connected if $r \leqslant \gamma$. We will exploit this condition to develop our clustering methodology based on atomic cells in the next section.

3. Proposed methodology

3.1. Problem settings

This subsection explains how the fair clustering problem is modeled in our paper. We first define *fair distribution* by a probability distribution such that for any Borel set $B \in \mathbb{R}^d$, the value of $P(X \in B | X \in G_i)$ is the same (or almost the same allowed by adopted fairness standards) for all $i = 1, \cdots, \ell$, where $G_1, \cdots, G_\ell$ denote the groups of sensitive attributes. To illustrate this definition in clustering, the fair distribution maintains a balance (almost) evenly across each cluster. Therefore, if a machine learning model (supervised or unsupervised) is built from samples generated by a fair distribution, the result of the model will be fair.

We assume a real data set $\{\mathbf{x}_i \in \mathcal{X} : i = 1, \ldots, N\}$ to originate from a sample data distribution $\mathcal{D}_S$. We view that the sample data distribution is the result of a deviation from a fair distribution $\mathcal{D}_F$ caused by socioeconomic biases. Note that the density of sample data distribution $\mathcal{D}_S$ can be approximated using a Gaussian kernel support density function. On the other hand, a fair correspondence distribution can be intractable because neither $\mathcal{D}_F$ nor its samples $\{\mathbf{x}_i \sim \mathcal{D}_F\}$ may exist in a real world scenario.

To make a fair distribution feasible and reflect the important characteristics of sample distribution, we propose the concept of *fair correspondence distribution*, which is a fair distribution that shares the dense areas near the sample data distribution modes of $\mathcal{D}_S$.

To search for a fair correspondence distribution, we construct a support density function for the sample distribution $\mathcal{D}_S$. Then, utilizing its associated multi-basin systems, we partition the input space into atomic cells. In particular, we intend the data points in each atomic cell computed from the support density function to result in the same cluster for a hierarchical fair clustering.

The purpose of this paper is to find a cluster labeling function that can reduce the generalization error risk on fair correspondence distribution $\mathcal{D}_F$ using the information from tractable real data samples $\{x_i \sim \mathcal{D}_S\}$.

The basic problem setting of is illustrated in Fig. 1. By considering a *fair correspondence distribution* with its atomic cells and their connectivity, we can obtain a fair clustering result. However, because we only have a biased sample distribution $\mathcal{D}_S$, we try to approximate the fair clustering using two different steps.

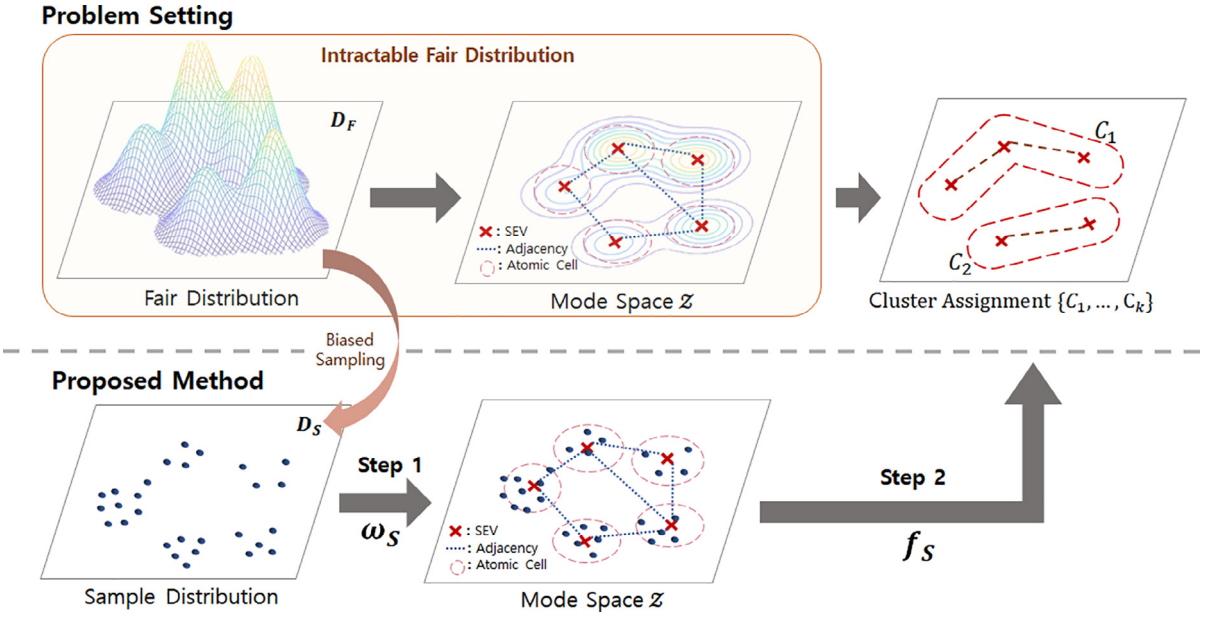


Fig. 1. Illustration of overall framework of our proposed method. **Up**: Problem settings of our paper. The mash plot denotes density function of fair correspondence distribution $\mathcal{D}_F$ while red $\times$ represents SEV, red regions denote basin cells corresponding to each SEVs (stable equilibrium vector), and the blue line represents adjacency between two SEVs. **Down**: Proposed method in our paper. The blue $\circ$ represents samples $\{x_1, \ldots, x_i\} \in \mathcal{D}_S$. (Best viewed in color.)

3.2. Overall framework

In this paper, we try to find a *fair* clustering, which is given in the form of (1), where p_F is the density of a fair correspondence distribution $\mathcal{D}_F$. We propose an indirect method, that does not model $\mathcal{D}_F$ explicitly but only extracts the corresponding clusters. This is done by adopting the frameworks from [22,21], which suggest a distribution-based clustering framework based on the idea of a dynamical system.

While we have data $\{\mathbf{x}_i\}$ in input space $\mathcal{X}$, let $\mathcal{Z} = \{\mathbf{s}_i : \mathbf{s}_i \in \mathcal{X},\ i = 1, \cdots, k\}$ refer to a mode space, i.e. a collection of a finite elements from the input space. The mode space $\mathcal{Z}$ is induced from $\mathcal{X}$ by using the representation function $\omega \in \mathcal{W} = \{\omega : \mathcal{X} \to \{\mathbf{s}_1, \ldots, \mathbf{s}_k\} = \mathcal{Z}\}$ where for any $\mathbf{x} \in \mathcal{X}, \omega(\mathbf{x}) = \mathbf{s}_i$ for some i. This means a representation function $\omega \in \mathcal{W}$ partitions the entire input space $\mathcal{X}$ into k cells. Our choice of $\mathcal{Z}$ would be the set of SEVs, with each of them representing a basin, or an atomic cell. Additionally, ω maps each $\mathbf{x} \in \mathcal{X}$ to a SEV which attracts $\mathbf{x}$. Note that Eq. 2 guarantees that ω is defined almost everywhere on $\mathcal{X}$.

For a distribution $\mathcal{D}$ with a support density $p(\cdot)$, the hypotheses $h \in \mathcal{H} = \{f \circ \omega : \omega \in \mathcal{W}\}$ are then formed by the process $\langle \mathcal{X}, \mathcal{D} \rangle \to \omega\left\langle \{\mathbf{s}_i\}_{i=1}^k, \mathcal{D} \right\rangle \to f\mathcal{Y}_K$ where $\mathcal{Y}_K = \{1, 2, \ldots, K\}$ (mono-label case) or $\mathcal{Y}_K = \{0, 1\}^K$ (multi-label case) and $f : \{\mathbf{s}_i\}_{i=1}^k \to \mathcal{Y}_K$ on $\mathcal{D}$ denotes a cluster labeling function that assigns a cluster label of a support density $p(\cdot)$ to $\mathbf{s}_i$. Note that in this setting, once the representation function ω is determined, the clustering problem is translated into a discrete labeling problem in the mode space. One of the distinguishing features of this process is that for the same input $\mathbf{x} \in \mathcal{X}, (f \circ \omega)(\mathbf{x}) = f(\omega(\mathbf{x}))$ may take the different values depending on whether $\mathbf{x} \sim \mathcal{D}_S$ or $\mathbf{x} \sim \mathcal{D}_F$. This feature leads to a sharp difference between a traditional machine learning setting and our fair clustering setting.

The basic framework of our method is illustrated at the bottom of Fig. 1. First, we approximate the density function $p_N(x)$ using the data samples. Then, using the presumed density function and its dynamical system, we partition the input space into atomic cells. We illustrate this as step 1 in Fig. 1, which uses the representation function ω. Then, in step 2, consider the adjacency of each atomic cell and the density value of each TEV to assign each cell to a final cluster using the cluster assignment function f.

In the following paragraphs, we will search for clustering given by $\mathcal{D}_F$ under a restriction that the modes of distribution should be equal to those of $\mathcal{D}_S$. Given this assumption, it is possible to reduce the clustering problem to a discrete setting, which allows us to develop a concise algorithm while still pursuing the main philosophy of our presented theory. This restriction also reflects our intuition that data points from a high-density region of $\mathcal{D}_S$ should be eventually grouped into the same cluster.

Because we need to find $\mathcal{D}_f$ only using data samples from $\mathcal{D}_s$, we put a restriction that we will find a $\mathcal{D}_f$ that shares the mode space with $\mathcal{D}_s$. This means we will only consider the $\mathcal{D}_f$ that learns the same mode space $\mathcal{Z}$ with $\mathcal{D}_s$, because there could be various $\mathcal{D}_f$s that satisfying optimal fair clustering.

One distinguishing feature of our framework is that for the same input data $\mathbf{x} \in \mathcal{X}$, $(f \circ \omega)(\mathbf{x}) = f(\omega(\mathbf{x}))$ may take different values depending on whether $\mathbf{x} \sim \hat{\mathcal{D}}_S$ or $\mathbf{x} \sim \hat{\mathcal{D}}_F$. This is because the cluster labeling function f takes not only the set of modes $\{s_i\}$ but also the distribution $\mathcal{D}$ as an input.

Given the density function p for $\mathcal{D}$, the labeling is done by lowering the truncation level r in the graph structure G_r described in the previous section, and grouping adjacent atomic cells when a new edge is added to the edge set E_r. Equivalently, clusters are formed hierarchically by merging atomic cells in a descending order of values of density p at the corresponding TEVs. Fig. 2 illustrates why the clustering result may change with data distribution through a simple one-dimensional case with three atomic cells. If $\mathcal{D} = \hat{\mathcal{D}}_S$, with density represented by the blue curve in Fig. 2, atomic cells $B(\mathbf{s_2})$ and $B(\mathbf{s}_3)$ are grouped into a single cluster since the density value at the TEV between them (which is a local minimum in this case) is greater than r. However, if $\mathcal{D} = \hat{\mathcal{D}}_F$ such that the red density curve is used, then $B(\mathbf{s}_1)$ and $B(\mathbf{s}_2)$ are grouped together instead of the second and third cells.

3.3. Theoretical framework

This subsection provides the core theory behind our proposed methodology. We provide a generalization error bound for the proposed clustering method.

We utilize the concepts of $\mathcal{H}$-divergence and Rademacher complexity in the theoretical development. Unlike the existing works on fair clustering, which usually address the problem via combinatorial observations, we take a novel approach by inferring the fair correspondence distribution $\mathcal{D}_F$. This distribution-based framework allows us to incorporate ideas from computational learning theory, which have never been used in the fair clustering literature.

3.3.1. A generalization error bound

In this section, we provide a new generalization error bound for fair clustering. Consider a clustering task where $\mathcal{X}$ is an input space and $S = \{\mathbf{x}_i \in \mathcal{X} : i = 1, \ldots, N_s\}$ represents the given unlabeled data of N samples. In the fair clustering settings, we have two distinct distributions over $\mathcal{X}$: a sample distribution $\mathcal{D}_S$ according to the sample density $p_S(\mathbf{x})$ and a ground truth fair correspondence distribution $\mathcal{D}_F$, according to the true fair support density $p_F(\mathbf{x})$.

Utilizing these concepts and following the notations in [26], and given a true fair cluster labeling function $c_F = f \circ \omega_F$, and a hypothesis set $\mathcal{H} = \{f \circ \omega : \omega \in \mathcal{W}\}$, the generalization error (or risk) of a K-cluster label hypothesis $h \in \mathcal{H}$ in a fair correspondence distribution $\mathcal{D}_F$ is defined by

$$\begin{aligned}\mathcal{R}_{\mathcal{D}_F}(f \circ \omega) &= \mathcal{R}_{\mathcal{D}_F}(f \circ \omega, f \circ \omega_f) \\ &= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_F}\left[\|f \circ \omega(\mathbf{x}) - f \circ \omega_f(\mathbf{x})\|_K\right]\end{aligned}$$

where $\|\cdot\|_K$ is the Hamming distance that measures the number of different components in two vectors. If we consider $\hat{\mathcal{D}}_S^N$ as $\hat{\mathcal{D}}_F^N$ samples of size N drawn independently according to the respective distributions $\mathcal{D}_S$ and $\mathcal{D}_F$, then the empirical error of $h \in \mathcal{H}$ in a fair correspondence distribution $\mathcal{D}_F$ is defined by

$$\hat{\mathcal{R}}_{\hat{\mathcal{D}}_F^N}(f \circ \omega, f \circ \omega_F) = \frac{1}{N} \sum_{\mathbf{x}_i \in \hat{\mathcal{D}}_F^N} 1_{f \circ \omega(\mathbf{x}_i) \neq f \circ \omega_F(\mathbf{x}_i)}$$

For a true sample cluster labeling function $c_s = f \circ \omega_s$, $\mathcal{R}_{\mathcal{D}_S}$ and $\hat{\mathcal{R}}_{\hat{\mathcal{D}}_S^N}$ are similarly defined.

To provide a rigorous model of fair clustering, we define the $\mathcal{W}$ divergence which measures the discrepancy between two different distributions by

$$d_{\mathcal{W}}(\mathcal{D}_S, \mathcal{D}_F) = \sup_{\omega, \omega' \in \mathcal{W}} |\mathcal{R}_{\mathcal{D}_S}(f \circ \omega, f \circ \omega') - \mathcal{R}_{\mathcal{D}_F}(f \circ \omega, f \circ \omega')|$$

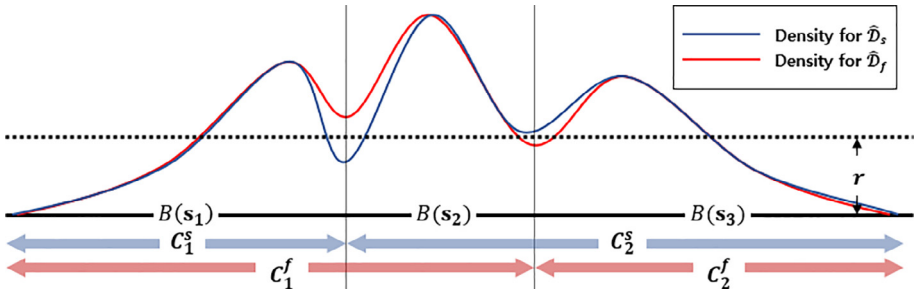


Fig. 2. Illustration of how cluster labeling may change with density functions. The blue and red curves each refer to empirical sample and fair correspondence distributions $\hat{\mathcal{D}}_S$ and $\hat{\mathcal{D}}_F$ respectively, and $B(\mathbf{s}_i)$ denotes atomic cells. Clustering results when $k = 2$, with assignment functions $f(\cdot, \hat{\mathcal{D}}_S)$ and $f(\cdot, \hat{\mathcal{D}}_F)$, are denoted by $\mathfrak{C}^s$ and $\mathfrak{C}^f$, respectively.

We are now ready to derive the following generalization error risk in our fair clustering setting.

Theorem 1. *Let $\mathcal{D}_S$ and $\mathcal{D}_F$ be the sample and the fair correspondence distributions with support densities $p_S(\cdot)$ and $p_F(\cdot)$, respectively. Let the ground truth cluster labeling function be $c_F = f \circ \omega_F$ induced from $\mathcal{D}_F$ and the optimal cluster labeling function in the sample distribution be $c_S = f \circ \omega_S$ induced from $\mathcal{D}_S$. Then for any hypothesis $f \circ \omega$ where $\omega \in \mathcal{W}$, the following inequality holds:*

$$\mathcal{R}_{\mathcal{D}_F}(f \circ \omega) \leqslant \mathcal{R}_{\mathcal{D}_S}(f \circ \omega) + \frac{1}{2} d_{\mathcal{W}}(\mathcal{D}_S, \mathcal{D}_F) + \min\left\{\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_S}[\|f \circ \omega_S(\mathbf{x}) - f \circ \omega_F(\mathbf{x})\|_K], \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_F}[\|f \circ \omega_S(\mathbf{x}) - f \circ \omega_F(\mathbf{x})\|_K]\right\} \tag{3}$$

Since we have restricted fair correspondence distributions to those that share modes with $\mathcal{D}_S$, the third term in Eq. (3) vanishes because $\omega_S = \omega_F$. The first term is the sample error caused by a discrepancy between the ground-truth sample density and the approximate support density. This is expected to be small with a large sample size, as the estimated density converges asymptotically to the true support density. The second term is a discrepancy measure between the two distributions, and can be interpreted as an intrinsic deviation of $\mathcal{D}_S$ from $\mathcal{D}_F$.

In summary, Theorem 1 tells us that the cluster labeling by a suitable combination of atomic cells represented by modes in $\mathcal{Z}$ could detect clusters induced by a fair density, with error bounded above by the right-hand side of Eq. (3), while the third term drops under the restriction $\omega_S = \omega_F$. This justifies our proposed method, which uses a greedy optimization of balance through an amalgamation of atomic cells in the mode space.

3.3.2. A margin-based generalization error bound

We next present margin-based generalization bounds in a fair clustering setting. Following the notations and definitions in [26], we first define Rademacher complexity to measure the capacity of a hypothesis space by its ability to fit random data.

Definition 2 (*Rademacher complexity*). For a real-valued function class $\mathcal{H} : \mathcal{X} \to [a, b]$ and a sample $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ generated by a distribution $\mathcal{D}_S$, the empirical Rademacher complexity of $\mathcal{H}$ with respect to S is the random variable

$$\hat{\mathfrak{R}}_{\mathcal{D}_S}(\mathcal{H}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}} \frac{1}{N}\sum_{i=1}^{N} \sigma_i h(\mathbf{x}_i)\right] \tag{4}$$

where $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_N\}$ are independent uniform $\{\pm 1\}$-valued Rademacher random variables. The *Rademacher complexity* of $\mathcal{H}$ is the expectation of the empirical Rademacher complexity over all samples of size m:

$$\mathfrak{R}_N(\mathcal{H}) = \mathbb{E}_{\mathcal{D}_S}\left[\hat{\mathfrak{R}}_{\mathcal{D}_S}(\mathcal{H})\right] = \mathbb{E}_{S\boldsymbol{\sigma}}\left[up_{h\in\mathcal{H}} \frac{1}{N}\sum_{i=1}^{N} \sigma_i h(\mathbf{x}_i)\right] \tag{5}$$

We define $\Lambda_{\mathcal{D}_S}(\mathcal{W})$ for a representation function $\omega \in \mathcal{W}$ by

$$\Lambda_{\mathcal{D}_S}(\mathcal{W}) = \left\{\mathbf{x} \mapsto 1_{y=f\circ\omega(\mathbf{x})} : \mathbf{x} \sim \mathcal{D}_S,\ y \in \mathcal{Y}_K,\ \omega \in \mathcal{W}\right\}.$$

The cluster label associated to point $\mathbf{x}$ is $f \circ \omega(\mathbf{x})$ which is the one resulting in the largest score $1_{y=f\circ\omega(\mathbf{x})}$. The *margin* $\xi^{\omega}_{\mathcal{D}_S}(x,\ y)$ at a sample cluster labeled example $(\mathbf{x},\ y)$ with $\mathbf{x} \sim \mathcal{D}_S$ is defined by

$$\xi^{\omega}_{\mathcal{D}_S}(\mathbf{x},\ y) = \begin{cases} 1 & \text{if } y = f \circ \omega(\mathbf{x}) \\ -1 & \text{otherwise} \end{cases}$$

Thus, $f \circ \omega$ misclassifies $(\mathbf{x},\ y)$ iff $\xi^{\omega}_{\mathcal{D}_S}(\mathbf{x},\ y) \leqslant 0$.

Let $S = \left\{(\mathbf{x}_i^s, y_i^s) \in \mathcal{D}_S \times \mathcal{Y}_K :\ i = 1, \ldots, m\right\}$ represent the cluster labeled data of m samples. For any $0 < \rho < 1$, we define the *empirical margin loss* for multi-class classification by

$$\hat{\mathcal{R}}_{S,\rho}(f \circ \omega) := \frac{1}{N}\sum_{i=1}^{N} \Psi_\rho\left(\xi^{\omega}_{\mathcal{D}_S}(\mathbf{x}_i,\ y_i)\right) \leqslant \frac{1}{m}\sum_{i=1}^{N} 1_{\xi^{\omega}_{\mathcal{D}_S}(\mathbf{x}_i,\ y_i)\leqslant\rho} = \hat{\mathcal{R}}_{\mathcal{D}_S}(f \circ \omega)$$

where Ψ_ρ is the margin loss function defined by

$$\Psi_\rho(x) = \begin{cases} 0 & \text{if } \rho \leqslant x \\ 1 - x/\rho & \text{if } 0 \leqslant x \leqslant \rho \\ 1 & \text{if } x \leqslant 0 \end{cases}$$

This upper bound is a fraction of the sample data points that have been misclassified or correctly classified with a confidence less than or equal to ρ.

Now, we are ready to present the following margin bound for fair clustering of a mono-label $\mathcal{Y}_K$ in the probably approximately correct (PAC) learning framework.

Theorem 2. *For any $\delta > 0$, with probability at least $1-\delta$, the following fair clustering generalization bound holds for all hypotheses $f \circ \omega$ where $\omega \in \mathcal{W}$:*

$$\mathcal{R}_{\mathcal{D}_F}(f \circ \omega) \leqslant \hat{\mathcal{R}}_{S,\rho}(f \circ \omega) + \frac{4K}{\rho}\mathfrak{R}_N\big(\Lambda_{\mathcal{D}_S}(\mathcal{W})\big) + 2\sqrt{\frac{\log\frac{2}{\delta}}{2N}} + d_{\mathcal{W}}\big(\hat{\mathcal{D}}_S^N, \hat{\mathcal{D}}_F^N\big) + 4\sqrt{\frac{\hat{d}\log\frac{eN}{2\hat{d}}}{N}} \tag{6}$$

where $\hat{d} = \mathrm{VCdim}(\{f \circ \omega : \omega \in \mathcal{W}\})$.

Here, we can drop the third term in Eq. (6) since a common representation function ω minimizes the fourth term when $\omega_S = \omega_F$ or ω_S and ω_F are sufficiently close to each other.

3.4. Algorithm

As shown in Fig. 1, we can divide the clustering procedure into two phases.

3.4.1. First phase

In the first phase, we approximate the density function and partition the input space into atomic cells. Specifically, we apply a margin-based one-class support vector method [28] to estimate the support density function of the $\mathcal{D}_S$. This is done by solving the problem

$$\begin{aligned} \min_{\substack{w \in \mathcal{H} \\ \boldsymbol{\xi}, \rho}} \quad & \tfrac{1}{2}||w||^2 + \tfrac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho, \\ \text{subject to} \quad & (w \cdot \Phi(\mathbf{x}_i)) \geqslant \rho - \xi_i, \\ & \xi_i \geqslant 0,\ \rho \geqslant 0, \end{aligned}$$

where $\Phi : \mathcal{X} \to \mathcal{H}$ is a feature map into an inner product space, $\xi = (\xi_1, \cdots, \xi_N)$ are slack variables, $\nu \in (0, 1]$ is a parameter and ρ is an additional variable to be optimized. For a Gaussian kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = e^{-q_N||\mathbf{x}_i - \mathbf{x}_j||^2}$ with width parameter q_N, the dual of this problem turns out to be equivalent to the following [29]:

$$\begin{aligned} \max \quad & W = \sum_{i=1}^{N}\beta_i - \sum_{i,j=1}^{N}\beta_i\beta_j e^{-q_N||\mathbf{x}_i - \mathbf{x}_j||^2} \\ \text{subject to} \quad & 0 \leqslant \beta_j \leqslant C_N, \quad \sum_j \beta_j = 1, \end{aligned} \tag{7}$$

where $j = 1, \ldots, N$. If we let its solution be $\bar{\beta}_i, i = 1, \ldots, N$, then the decision function is given by

$$\begin{aligned} s_N(\mathbf{x}) \quad &= 1 - 2\sum_{i=1}^{N}\bar{\beta}_i e^{-q_N||\mathbf{x} - \mathbf{x}_i||^2} \\ &\quad + \sum_{i,j=1}^{N}\bar{\beta}_i\bar{\beta}_j e^{-q_N||\mathbf{x}_i - \mathbf{x}_j||^2}. \end{aligned}$$

From this we obtain the trained Gaussian kernel support density function given by

$$p_N(\mathbf{x}) = (q_N/\pi)^{n/2}\sum_{i=1}^{N}\bar{\beta}_i e^{-q_N||\mathbf{x} - \mathbf{x}_i||^2}, \tag{8}$$

which converges asymptotically to the true sample support density under some mild conditions.

Theorem 3. *Let a sample $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be generated by a distribution $\mathcal{D}_S$ with a sample support density $p_S(\mathbf{x})$. Assume the following conditions are satisfied,*

$$\lim_{N\to\infty} q_N = \infty \quad \text{and} \quad \lim_{N\to\infty} NC_N^2 q_N^{d/2} = 0.$$

Then the estimate $p_N(\mathbf{x})$ of the form

$$p_N(\mathbf{x}) = (q_N/\pi)^{n/2}\sum_{i=1}^{N}\bar{\beta}_i e^{-q_N||\mathbf{x} - \mathbf{x}_i||^2}, \tag{9}$$

converges asymptotically to $p_S(\mathbf{x})$.

This would serve as a theoretical basis for justifying our density-based methodology.

Since we assume the modes of $\mathcal{D}_S$ and $\mathcal{D}_F$ to be equal, we can estimate the representation function $\hat{\omega}$ by learning the real samples. After estimating $\hat{\omega}$, we obtain k atomic cells $\{\mathbf{s}_i\}_{i=1}^{k}$ and l TEVs $\{\mathbf{t}_j\}_{j=1}^{l}$.

3.4.2. Second phase

The second phase involves grouping k atomic cells into K final clusters. If we have estimated density function $\hat{p}$, we can merge the cells hierarchically in the descending order of $\hat{p}(\mathbf{t}_j)$s. However, although we have assumed the mode space $\{\mathbf{s}_i\}$ of $\mathcal{D}_F$ to be equal to $\mathcal{D}_S$, the density function of the fair correspondence distribution p_f is intractable. To overcome this difficulty, we require the density function of fair correspondence distribution p_f to be similar to the density function of the estimated sample distribution $\hat{p}_s$, with a discrepancy between the two coming from unknown, immutable socioeconomic biases. We resort to our theoretical result that fair clustering assignment based on atomic cells can address this discrepancy, and suggest a fair labeling method by adding of fair constraints to the estimated sample density values with a hyperparameter, which would then serve as a guideline for greedy merging at each step.

The estimated value of the (unnormalized) density of fair clustering at $\mathbf{t}_j$ is as follows:

$$\hat{p}_f(\mathbf{t}_j) = \hat{p}_s(\mathbf{t}_j) - \lambda \cdot \left[\text{balance}(C_{j_+} \cup C_{j_-}) - \min(\text{balance}(C_{j_+}), \text{balance}(C_{j_-}))\right], \tag{10}$$

where j_+ and j_- denote the indices of adjacent SEVs corresponding to $\mathbf{t}_j$, while C_{j_+} and C_{j_-} are interim sub-clusters (unions of atomic cells) to which $\mathbf{s}_{j_+}$ and $\mathbf{s}_{j_-}$ belong, respectively. λ is a hyperparameter that determines the degree to which the fair constraint is reflected in the algorithm. We update $\hat{p}_f(\mathbf{t}_j)$ after every merging as the sub-cluster assignment is modified. The fair constraint term detects an increase in balance when sub-clusters $\mathbf{s}_{j_+}$ and $\mathbf{s}_{j_-}$, which are adjacent to TEV $\mathbf{t}_j$, are merged.

3.4.3. Pseudocode

In this section, an algorithm representing our proposed framework in pseudocode is demonstrated for clarity. Algorithm 1 shows the construction of graph G_r, while Algorithm 1 represents labeling with a balance update. A Fair SVC is implemented in Algorithm 1 and 2. After learning the representation function $\hat{\omega}$ to partition the input space into k-clustered regions with various Gaussian width parameters, $\hat{f}$ is learned to assign atomic cells to cluster labels.

Since our method relies on hypercubes and numerical integration on these hypercubes, and need to solve the quadratic programming to obtain the support density function, the algorithm does not scale well with the dataset dimention d. Therefore, it is limited to low-dimensional problems.

Our implementation in Matlab is available at https://github.com/wj926/Fair_SVC.

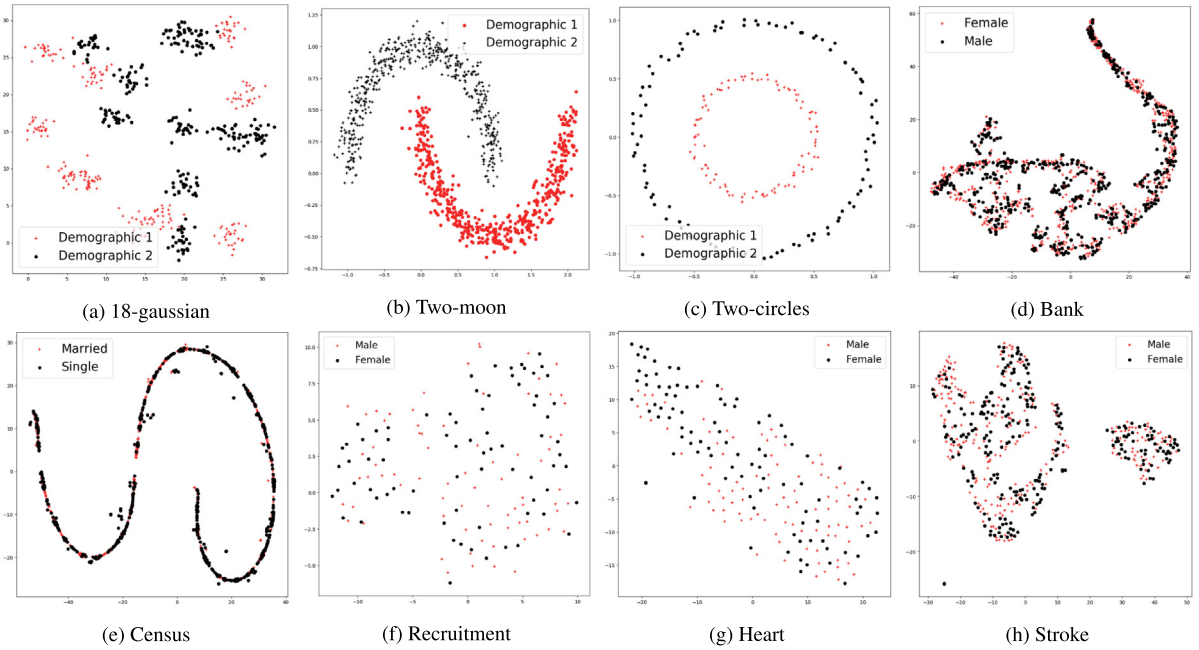


Fig. 3. Visualization of the original data distribution of synthetic and real datasets in a two-dimensional data space. To visualize high-dimensional real datasets, dimension reduction of the data set was performed using t-SNE. The sensitive variable is represented by two different markers.

Algorithm 1. Construction of Graph G_r

Data: $D = \{\mathbf{x}_i = (x_i^1, \ldots, x_i^d)^T, z_i\}_{i=1}^N \times \mathbb{R}^d \cup \{0, 1\}$ where $\mathbf{z}$ denotes the sensitive variable

Result: Weighted graph $G_r = (V, E)$

1. Initialization

Train a support density function p. Set $a^m = \min_i x_i^m$ and $b^m = \max_i x_i^m,\ m = 1, \ldots, d$.

2. Constructing the vertices V and decomposing data points into several disjoint sets

$k = 0, V = \emptyset$; // a set of stable equilibrium points;

for *each data point* $\mathbf{x}_i \in D$, $i = 1, \ldots, N$ **do**

  numerically integrate the gradient system starting from $\mathbf{x}_i$ until it reaches an SEP $\mathbf{x}_i^*$

  if $\mathbf{x}_i^* \notin V$ **then**

    make $\langle \mathbf{s}_{k+1} \rangle$; $\mathbf{s}_{k+1} \leftarrow \mathbf{x}_i^*$, $V \leftarrow \{\mathbf{s}_{k+1}\} \cup V$; $\mathbf{x}_i \in \langle \mathbf{s}_{k+1} \rangle$ and $k \leftarrow k + 1$;

  else

    find $\mathbf{s} \in V$ such that $\mathbf{x}_i^*$ and $\mathbf{x}_i \in \langle \mathbf{s} \rangle$

  end

end

3. Finding the equilibrium points of system

Divide the region with $a^m \le x^m \le b^m,\ m = 1, \ldots, d$ into several hypercubes with length $(b^m - a^m)/\ell$, where ℓ is the user-defined step length.

Randomize the order of data points and initialize $visit(U) = False$ for each hypercube U.

for *each data point* $\mathbf{x}_i \in D$ **do**

  find the hypercube U enclosing $\mathbf{x}_i$;

  if $visit(U) = False$ **then**

    find the solution of $\nabla p(\mathbf{x}) = 0$ starting from $\mathbf{x}_i$; $visit(U) = True$;

  end

end

4. Constructing the edges E

Identify the index-one saddle points $\mathbf{t}_i$, with $p(\mathbf{t}_i) > r,\ i = 1, \ldots, l$, from the equilibrium points obtained in 2 by checking the eigenvalues of the negated Hessian $-\nabla^2 p(\mathbf{t}_i)$.

Let $T = \emptyset$; // a set of TEVs

for $i = 1$ *to* l **do**

  Find a unit length eigenvector $\mathbf{v}_i$ corresponding to a negative eigenvalue of $-\nabla^2 p(\mathbf{t}_i)$. Set $\mathbf{x}_i^+ = \mathbf{t}_i + \epsilon \mathbf{v}_i$ and $\mathbf{x}_i^- = \mathbf{t}_i - \epsilon \mathbf{v}_i$ for some small $\epsilon > 0$.

  Numerically integrate the gradient system starting from $\mathbf{x}_i^+$ and $\mathbf{x}_i^-$ until they approach the SEPs $\mathbf{s}_i^+$ and $\mathbf{s}_i^-$, respectively.

  if $\mathbf{s}_i^+ ! = \mathbf{s}_i^-$ **then**

    Set $\langle \mathbf{s}_i^+, \mathbf{s}_i^- \rangle \in E$ with $d_E(\mathbf{s}_i^+, \mathbf{s}_i^-) = p(\mathbf{t}_i)$;

    $T \leftarrow \{\mathbf{t}_i\} \cup T$;

  end

end

Algorithm 2. Labeling with Balance Update

Data: support function s and its associated weighted graph $G_r = (V, E)$ where $\mathbf{s}_i, i = 1, \ldots, k$ is the set of SEVs and $\mathbf{t}_j$, $j = 1, \ldots, l$ is the set of TEVs

Result:

1. Given a number of clusters K
2. Start with initial sub-clusters $C_1 = \langle \mathbf{s}_1 \rangle), \ldots, C_k = \langle \mathbf{s}_k \rangle$
3. In the initial step, the distance between two clusters w.r.t the corresponding TEV is defined as $d(C_{j_+}, C_{j_-}) = \hat{p}_f(\mathbf{t}_j) = s(\mathbf{t}_j) - \lambda[\text{balance}(C_{j_+} \cup C_{j_-}) - \min(\text{balance}(C_{j_+}), \text{balance}(C_{j_-}))]$, where $\mathbf{s}_j^+ \in C_{j_+}, \mathbf{s}_j^- \in C_{j_-}$, $j_+, j_- \in 1, \ldots, k$, for all j
4. Rearrange the index $j = 1, \ldots, l$ in such a way that $d(C_{1_+}, C_{1_-}) < d(C_{2_+}, C_{2_-}) < \cdots < d(C_{l_+}, C_{l_-})$
5. Set $I = \{1, \ldots, k\}$, $J = \{1, \ldots, l\}$ and $m = 1$ **while** $m \leq k - K$ **do**
   - $C_{k+m} = C_{1_+} \cup C_{1_-}$;
   - $I \leftarrow \{k + m\} \cup (I \setminus \{1_+, 1_-\})$;
   - $J \leftarrow J \setminus \{1\}$;
   - **for** $j = 2, \ldots, |J|$ **do**
     - **if** $C_{j_+} \cup C_{j_-} \subset C_{k+m}$ **then**
       - $J \leftarrow J \setminus \{j\}$;
     - **end**
   - **end**
   - Find all the neighboring clusters of C_{1_+}, C_{1_-} according to the reduced J and replace C_{1_+} or C_{1_-} with C_{k+m}; recalculate $d(C_{j_+}, C_{j_-})$ for corresponding j;
   - Rearrange the index $j = 1, \ldots, |J|$ in the same way as 4;

end

4. Experiments

In this section, our proposed algorithm and comparative methods are experimentally applied to both synthetic and real datasets. We have used nine datasets for our evaluation.

4.1. Datasets

4.1.1. Synthetic datasets

To visualize how our model connects basin cells fairly and to evaluate its performance, we apply several algorithms including the proposed one on three well-known 2D synthetic datasets: **Two-circles**, **Two-moons**, and **18-Gaussian** as shown in Fig. 3. The samples were generated using the open source library scikit learn functions. We assign sensitive attribute values to the generated datasets in such a way that traditional clustering methods without fairness may result in a low balance. The ratio of the values of the sensitive variables is always set to 1:1 for simplicity.

4.1.2. Real datasets

We used three widely-used standard datasets for the evaluation of fair clustering algorithms: **bank**, **census**, and **diabetes**. The **bank**[1] dataset has 4,521 data points and represents marketing campaigns of a Portuguese banking institution. "marital" is used as a sensitive variable, which has three elements. We use only two of them except "divorced". The **census**[2] dataset has 32,561 points and is extracted from the 1994 US Census database. It contains two sensitive variables, but we only use "sex". The **diabetes**[3] dataset was extracted from a database of encounters of 10 years of clinical care of patients with diabetes at

[1] https://archive.ics.uci.edu/ml/datasets/Bank+Marketing.
[2] https://archive.ics.uci.edu/ml/datasets/adult.
[3] https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008.

Table 1
Description of synthetic datasets.

Dataset	Num. of points	Sensitive variable
Two-circles	200	0 for inner circle with radius 0.5 1 for outer circle with radius 1
Two-moon	1000	0 for upper moon 1 for lower moon
18-gaussian	500	As shown in Fig. 3

Table 2
Description of standard real datasets.

Dataset	features	Sensitive variable	values of sensitive variable
bank	age, balance, duration	marital	single, married
census	age, final-weight, education-num, capital-gain, hours-per-week	sex	male, female
diabetes	age, time-in-hospital	gender	male, female

Table 3
Description of real-world datasets.

Dataset	features	Sensitive variable	values of sensitive variable
recruitment	field of degree, education percentage, degree percentage, work experience, etc.	gender	male, female
heart	age, blood pressure, cholesterol in mg, fasting blood sugar, chest pain type, etc.	sex	male, female
stroke	age, hypertension work type, bmi smoking status, etc.	gender	male, female

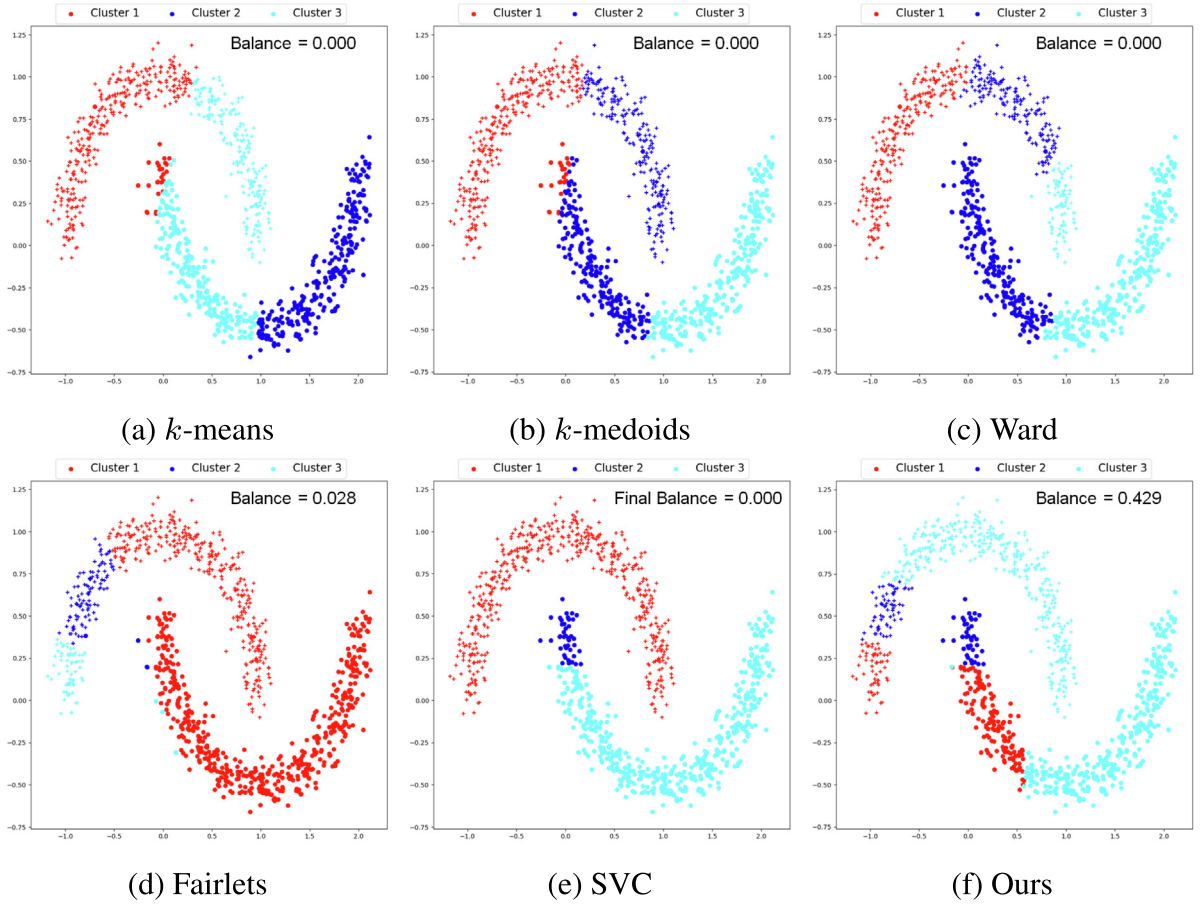
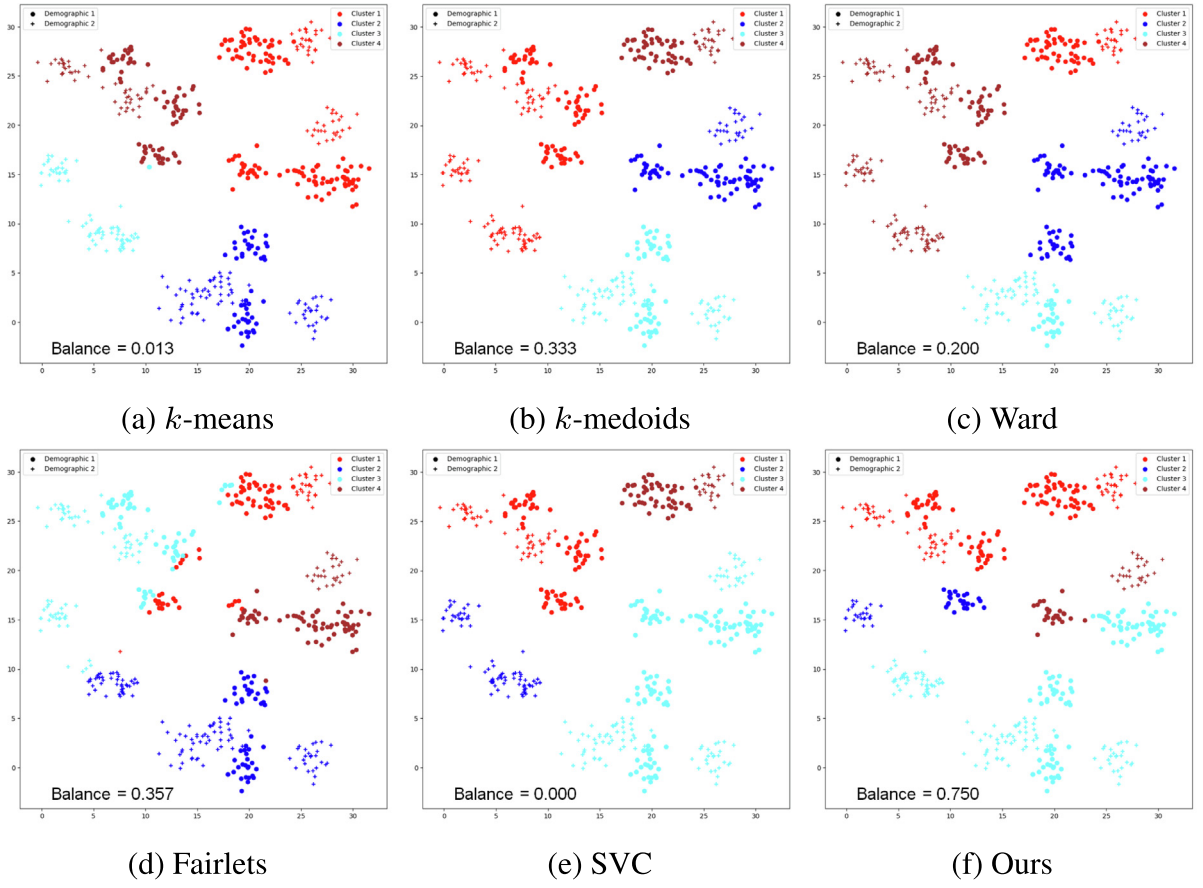


Fig. 4. Visualization of the clustering results of Two-moon dataset for each model in a two-dimensional data space, where the number of cluster is three. Sensitive variable is represented by two different markers, the result of clustering is represented by three different colors.

Table 4
Detailed configuration of each cluster in Two-moon dataset.

Cluster number	Sensitive attribute	Methods					
		k-means	k-medoids	Ward	Fairlets	SVC	Ours
Cluster 1	# of samples (s = 0)	300	280	246	337	500	60
	# of samples (s = 1)	26	23	0	494	0	140
	Total	326	303	246	831	500	200
	Balance of cluster 1	0.086	0.082	0.000	0.682	0.000	0.429
Cluster 2	# of samples (s = 0)	261	220	172	104	0	64
	# of samples (s = 1)	0	202	221	3	44	44
	total	261	422	393	107	44	108
	Balance of cluster 2	0.000	0.918	0.778	0.028	0.000	0.687
Cluster 3	# of samples (s = 0)	200	0	82	59	0	376
	# of samples (s = 1)	213	275	279	3	456	316
	total	413	275	361	62	456	692
	Balance of cluster 3	0.939	0.000	0.294	0.051	0.000	0.840
	Balance	0.000	0.000	0.000	0.028	0.000	**0.429**



(a) k-means (b) k-medoids (c) Ward

(d) Fairlets (e) SVC (f) Ours

Fig. 5. Visualization of the clustering results of 18-Gaussian dataset for each model in a two-dimensional data space, where the number of cluster is four. Sensitive variable is represented by two different markers, the result of clustering is represented by four different colors.

130 US hospitals. "gender" is used as a sensitive variable. For all three datasets, we use only numerical variables and sample 500 points for each value of the sensitive variable, which is the same experimental setting as [11]. Table 2 shows the detailed use of the three UCI datasets.

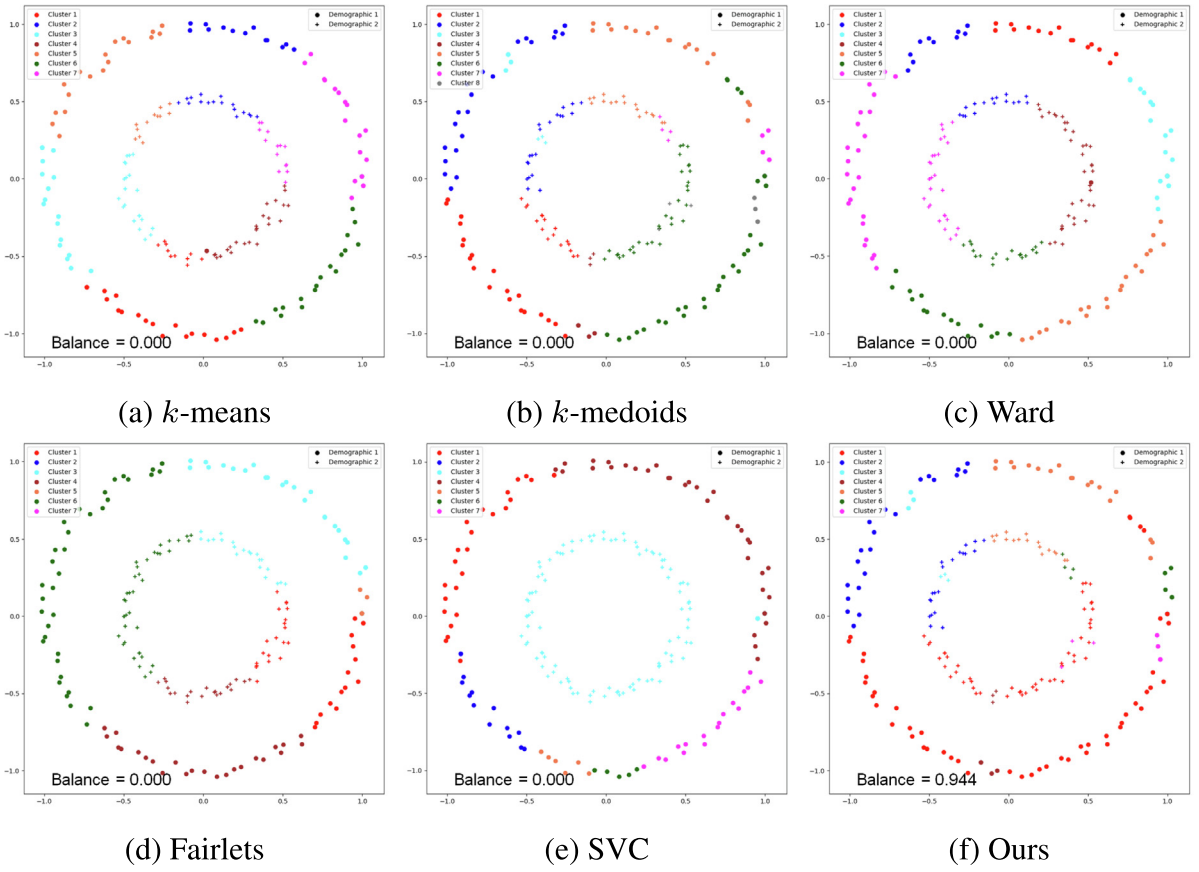


Fig. 6. Visualization of the clustering results of Two-circles dataset for each model in a two-dimensional data space, where the number of cluster is seven. Sensitive variable is represented by two different markers, the result of clustering is represented by seven different colors.

Moreover, we additionally experimented on three different real-world datasets: **recruitment**, **heart**, and **stroke**. The **recruitment**[4] dataset has 215 data points 15 attributes that represents academic and employ ability factors that influences placement. 'gender' is used as a sensitive attribute, which has two elements. **heart**[5] dataset has 304 points with 15 attributes that represents healthcare information. We used 'sex' as a sensitive attribute. The **stroke**[6] dataset contains 5511 data points that has 11 clinical features for predicting stroke events. In this dataset we used "gender" as a sensitive variable. Table 3 shows the description of three real-world datasets.

A detailed description of the synthetic and real datasets is provided in Table 1. Fig. 3(d) and (e) show the data samples from the bank and census datasets, respectively. Since these datasets are high-dimensional, a dimension reduction via t-SNE has been performed for the sake of clear visualization.

4.2. Experimental settings

***k*-means**, ***k*-medoids**,**Ward**, **Fairlets**, **SVC**, and ***k*-means** are used as benchmark models. SVC denotes the support vector-based clustering method utilizing a dynamical system [21], and fairlets denote a scalable fair k-median clustering algorithm [3].

In the experiments, balance, which is a method for measuring fairness in clustering algorithms, is measured for each model for the number of clusters $K = 1, \ldots, 9$. *Balance* $= 1$ implies that all the clusters have perfect fairness for the sensitive attribute. In addition, the computation time is also measured to verify the time efficiency of our proposed model in comparison to other benchmark models, as in the main study. We implemented fair SVC, normal SVC using MATLAB, and used open source library on other clustering methods in MATLAB. Exceptionally, a scalable fairlets algorithm was implemented in Python 3.6.8 with a *scikit-learn* library.

[4] https://www.kaggle.com/benroshan/factors-affecting-campus-placement.
[5] https://www.kaggle.com/nareshbhat/health-care-data-set-on-heart-attack-possibility?.
[6] https://www.kaggle.com/fedesoriano/stroke-prediction-dataset.

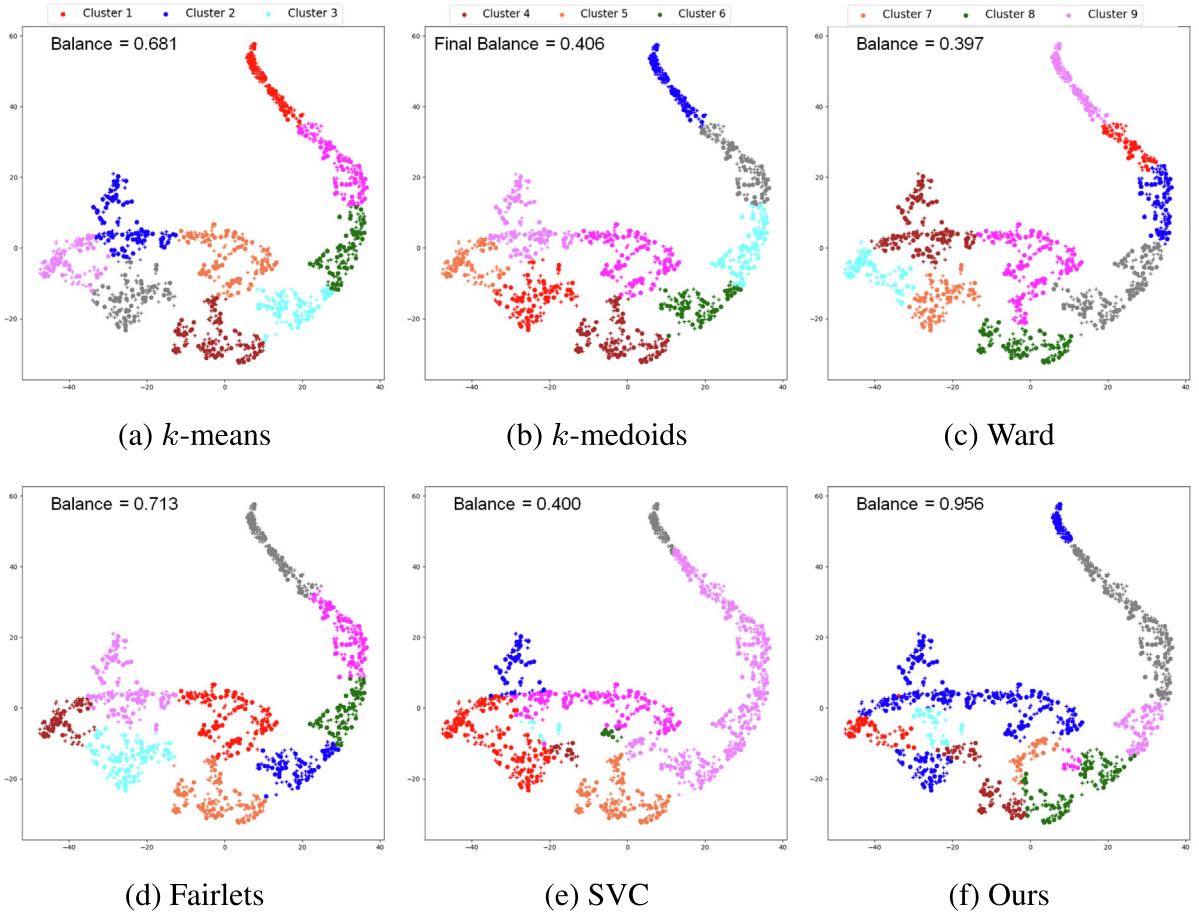


Fig. 7. Visualization of the clustering results of Bank dataset for each model in a two-dimensional data space. To visualize high-dimensional data, the dimension reduction of the data set was performed using T-SNE. Sensitive variable(single or married) is represented by two different markers. The result of clustering is represented by nine different colors.

4.3. Results

We show the clustering results in three synthetic datasets (Two-moons, 18-Gaussian, and Two-circles) and two real datasets (bank and census). The results of the clustering are represented by different colors and we indicate the value of the balance in each figure. The closer the value is to 1, the fairer the result will be.

4.3.1. Results of synthetic datasets

Fig. 4 visualizes the clustering results for the two-moons dataset with $k = 3$. Since the data are almost symmetric with respect to the x-axis, k-means, k-medoids, and Ward simply partition the data space parallelly to the y-axis. SVC captured each moon well but resulted in an overall balance zero. The Fairlets algorithm was designed to balance the sensitive attribute values, but did not show good balance values, and the sizes of the clusters varied significantly. In comparison, our model yields balanced clusters with relatively consistent sizes.

Table 4 shows the detailed configuration of the results in Fig. 4. First, we find that the sample of Cluster 2 of k-means are from from one demographic group (s = 1). Consequently, the balance of this cluster becomes zero. Likewise, the balance of k-medoids, Ward, and SVC is also zero, since they have at least one cluster that consists of samples from a single group. Fairlets, which also seeks a fair clustering, failed in this experiment, resulting in a balance of 0.028. In contrast, all three clusters based on of our methods consist of samples from both demographic groups (s = 0 and s = 1). As a result, the balance of our method is 0.429, the fairest result among the compared methods.

Fig. 5 illustrates the results of the 18-Gaussian in $k = 4$. The samples of this dataset are randomly chosen from 18 different Gaussian distributions, and each distribution has one sensitive attribute. From a traditional clustering perspective, all six

methods successfully partitioned the samples into four clusters. However, in terms of fairness, our method outperformed the other methods.

The clustering results of two-circles is shown in Fig. 6 for $k = 7$. In this dataset, the inner circle and outer circle have different types of sensitive attributes. Therefore, to obtain fair results, each cluster must contain samples from both circles. All other five comparison methods resulted in a zero balance, which indicates that at least one of the clusters consists of only samples of inner or outer circles. On the other hand, our method has a balance value of 0.944, which means all clusters have almost the same number of samples in different groups.

4.3.2. Results from real datasets

Fig. 7 shows the clustering results for the bank dataset. Because the bank data is high-dimensional data, dimension reduction is performed by using t-SNE for a clear visualization. The results of *k*-means, *k*-medoids, and Ward partition the data space into simple convex clusters. Fairlets attempt to cluster while controlling the balance over sensitive attributes, but the result does not differ significantly from that of *k*-means. SVC attempts to find a clustering based on sample density, but yields the lowest balance (0.400). On the other hand, our model captures the shape of the data soundly by partitioning the data space into non-convex clusters, while demonstrating a much higher balance value (0.956) than the fairlets.

The results in Fig. 8 visualize the clustering results of the census dataset when k is five. Among these, our method showed different results than the other methods. Consequently, our method was the only one with a balance value close to 1.

Fig. 9 summarizes the balance of the proposed model and the comparative models for each dataset used in our experiments, while the balance is measured according to the definition in Section 2.1. Our model (red) outperformed the other models by achieving the highest balance value in almost every experiment. We can find that our proposed method shows better results in terms of fairness in standard datasets and real-world datasets. The running times of the methods are summarized in Fig. 10. Compared to fairlets, which also focuses on fair clustering, our method significantly reduced computational costs by using a distribution based approach.

The results of our method on datasets with a larger number of samples are summarized in Appendix B. To handle large datasets, we have used uniform sampling to construct support density function p, atomic cells, and the edges. Then we pre-

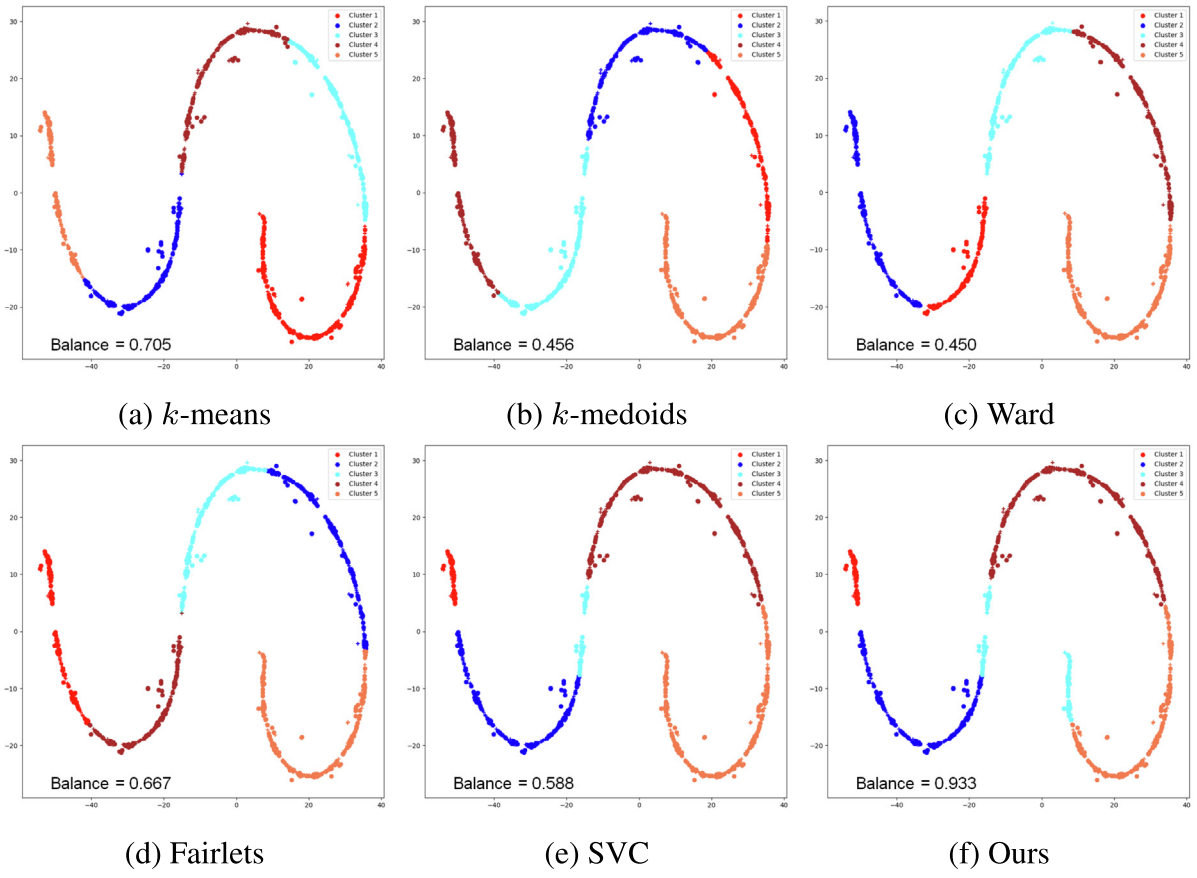
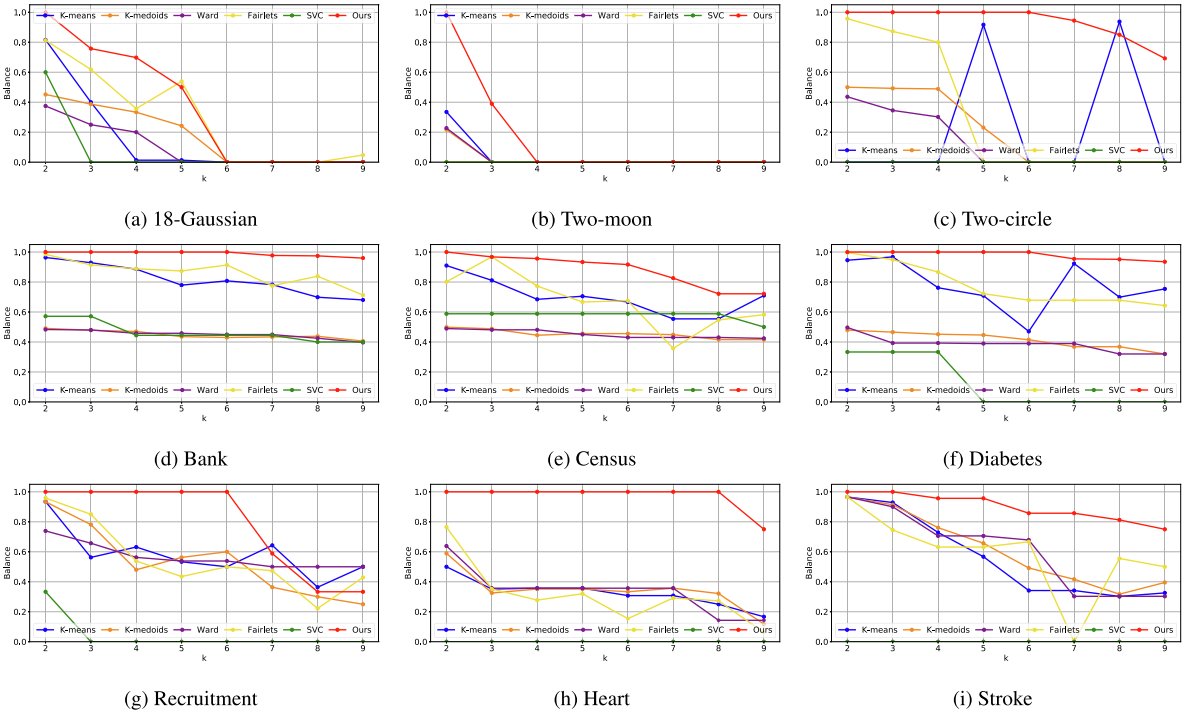


Fig. 8. Visualization of the clustering results of Census dataset for each model in a two-dimensional data space. To visualize high-dimensional data, the dimension reduction of the data set was performed using T-SNE. Sensitive variable(male or female) is represented by two different markers. The result of clustering is represented by five different colors.

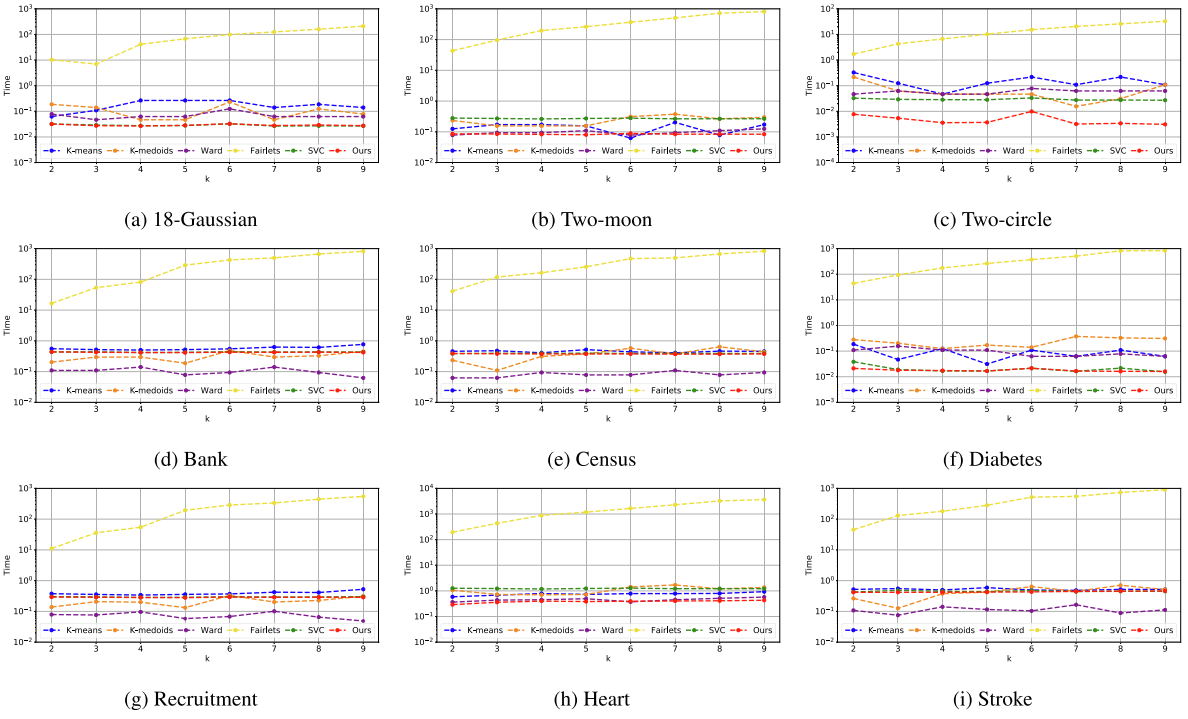


(a) 18-Gaussian (b) Two-moon (c) Two-circle

(d) Bank (e) Census (f) Diabetes

(g) Recruitment (h) Heart (i) Stroke

Fig. 9. Balance of ours and comparative models on the synthetic, standard, and real-world data sets with varying number of clusters from k = 2 to 9.



(a) 18-Gaussian (b) Two-moon (c) Two-circle

(d) Bank (e) Census (f) Diabetes

(g) Recruitment (h) Heart (i) Stroke

Fig. 10. Running time (s) of ours and comparative models on the synthetic and real data sets with varying number of clusters from k = 2 to 9.

dict the cluster labels of the whole data points. In both datasets, our method showed high balance value regardless of the number of k. In this way, we obtain fair clustering results of big datasets without the computational burden.

4.3.3. Discussion

In most experiments, our algorithm gave more balanced clustering results, than the fairlets-based model. In addition, the execution time of our algorithm was always significantly shorter than that of the fairlets algorithm, which sometimes required hours for an optimization. While the time complexity of our model was consistent with the number of clusters, the fairlets algorithm displayed an exponential growth in time consumption with an increase in k.

Our method is based on SVC, which accurately detects non-convex clusters by sequentially merging the most adjacent cells. However, our method sometimes combines atomic cells that are sub optimal in terms of adjacency, to find balanced clusters. Hence, the clustering results might deviate from the intuitive image of an ideal clustering. Nevertheless, it is a natural consequence of the trade-off between fairness and performance in machine learning models.

We have applied our method on three real-world dataset (Recruitment, Heart, and Stroke) from Kaggle, and our method has successfully obtained fair clustering results. Fair clustering results can be widely used when having a decision making on policies or strategies. For example, fair clustering results on Recruitment dataset can be used when the government plans to provide social aids to job applicants based on clustering results. Or it can be used for companies are trying to hire based on clustering results while considering fairness issues.In addition, the results for the Heart and Stroke dataset can be utilized when governments are trying to provide social assistance based on individual health care datasets with the majority and minority ratios maintained.

5. Conclusion

In this study, we presented a new perspective to address the fair clustering problem based on data distribution. We hypothesized that a fair clustering can be achieved by tracing a fair correspondence distribution from the sample distribution, which is considered biased because of real-world problems. Based on the theoretical results, we proposed an efficient cluster labeling framework at an atomic-cell level with an upper bound on the generalization error of the resulting clustering function.

Our study can be extended in several ways. Here we propose a greedy amalgamation of sub-clusters, grounded on the precision guarantee for an optimal labeling method over the mode space. However, it may be possible to improve this methodology through a direct approximation of the fair density. In addition, the time complexity may be reduced further if we choose different methods for density estimation. We anticipate that distribution-based approach can be naturally applied to other important tasks such as fair classification, in addition to the fair clustering problems treated here.

At last, the term 'balance' used in fair clustering is focused on binary variable. As a result, various previous works and our paper only considered the sensitive attributes on binary cases. However, recently using binary attributes is not modern enough. For example, 'sex' should be considered as a complex point of view [27]. For future work, we will work on suggesting an algorithm that can consider various types of sensitive attributes.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (NRF- 2019R1A2C2002358).

Appendix A. Definitions, Proof of Theorems, and Additional Explanations

First we will provide basic definitions related with generalization error and PAC learning as in [26]. Given a hypothesis $h \in \mathcal{H}$, a target concept $c : \mathcal{X} \to \mathcal{Y}$, a loss function $L : \mathcal{Y} \times \mathcal{Y} \to \Re^+$, and an underlying distribution D,

(i) the *generalization error* or *risk* of h is defined by

$$\begin{aligned} R_D(h) &= \mathbb{E}_{x\sim D}[L(h(x), c(x))] \quad \text{(regression)} \\ &= \mathbb{P}_{x\sim D}[h(x) \neq c(x)] = \mathbb{E}_{x\sim D}\left[1_{h(x)\neq c(x)}\right] \quad \text{(binary classification)} \end{aligned}$$

(ii) the *empirical error* or *empirical risk* of h for a sample $\hat{\mathcal{D}}_N = \{x_1, .., x_N\}$ is defined by

$$\begin{aligned} \widehat{R}_{\hat{D}_N}(h) &= \tfrac{1}{N}\sum_{i=1}^{N}[L(h(x_i), c(x_i))] \quad \text{(regression)} \\ &= \tfrac{1}{N}\sum_{i=1}^{N}\left[1_{h(x_i)\neq c(x_i)}\right] \quad \text{(binary classification)} \end{aligned}$$

Note that from the Law of the Large Numbers (LLN), the empirical error based on an i.i.d. sample $\hat{\mathcal{D}}_N$ converges to the generalization error as $N \to \infty$:

$$\lim_{N\to\infty} \widehat{R}_{\hat{\mathcal{D}}_N}(h) = R_{\mathcal{D}}(h))$$

Definition 3 [26]. A concept class $\mathcal{C}$ is said to be PAC-learnable if there exists an algorithm $\mathcal{A}$ and a polynomial function $poly(\cdot,\cdot,\cdot,\cdot)$such that for any $\epsilon > 0$ and $\delta > 0$, for all distributions $\mathcal{D}$ on $\mathcal{X}$ and for any target concept $c \in \mathcal{C}$, the following holds for any sample size $m \geqslant poly(1/\epsilon, 1/\delta, n, size(c))$:

$$\mathbb{P}_{S\sim\mathcal{D}^m}[R(h_S) \leqslant \epsilon] \geqslant 1-\delta..$$

If $\mathcal{A}$ further runs in $poly(1/\epsilon, 1/\delta, n, size(c))$, then $\mathcal{C}$ is said to be efficiently PAC-learnable. When such an algorithm $\mathcal{A}$ exists, it is called PAC-learnable algorithm for $\mathcal{C}$.

In PAC learning framework, the learner has samples and generalization function from a certain class of possible functions. Then, with high probability ('probability'), the goal is to select a function that have a low generalization error ('approximately correct'). The learner need to learn the concept given any arbitrary approximation ratio, probability of success, or distribution of the samples.

Theorem 4. *Let $\mathcal{D}_S$ and $\mathcal{D}_F$ be the sample and the fair correspondence distributions with support densities $p_S(\cdot)$ and $p_F(\cdot)$, respectively. Let the ground truth cluster labeling function be $c_F = f \circ \omega_F$ induced from $\mathcal{D}_F$ and the optimal cluster labeling function in the sample distribution be $c_S = f \circ \omega_S$ induced from $\mathcal{D}_S$. Then for any hypothesis $h \in \mathcal{H} = \{f \circ \omega : \omega \in \mathcal{W}\}$, the following inequality holds:*

$$\mathcal{R}_{\mathcal{D}_F}(f\circ\omega) \leqslant \mathcal{R}_{\mathcal{D}_S}(f\circ\omega) + d_{\mathcal{W}}(\mathcal{D}_S,\mathcal{D}_F) + \min\{\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_S}[\|f\circ\omega_S(\mathbf{x}) - f\circ\omega_F(\mathbf{x})\|_K], \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_F}[\|f\circ\omega_S(\mathbf{x}) - f\circ\omega_F(\mathbf{x})\|_K]\} \tag{A.1}$$

Proof 1.

$$\begin{aligned}
\mathcal{R}_{\mathcal{D}_F}(f\circ\omega) &= \mathcal{R}_{\mathcal{D}_F}(f\circ\omega, f\circ\omega_F)\\
&= \mathcal{R}_{\mathcal{D}_S}(f\circ\omega, f\circ\omega_F) + \mathcal{R}_{\mathcal{D}_F}(f\circ\omega, f\circ\omega_F) - \mathcal{R}_{\mathcal{D}_S}(f\circ\omega, f\circ\omega_F)\\
&\leqslant \mathcal{R}_{\mathcal{D}_S}(f\circ\omega, f\circ\omega_F) + d_{\mathcal{W}}(\mathcal{D}_S,\mathcal{D}_F)\\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_S}[\|f\circ\omega(\mathbf{x}) - f\circ\omega_F(\mathbf{x})\|_K] + d_{\mathcal{W}}(\mathcal{D}_S,\mathcal{D}_F)\\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_S}[\|f\circ\omega(\mathbf{x}) - f\circ\omega_S(\mathbf{x}) + f\circ\omega_S(\mathbf{x}) - f\circ\omega_F(\mathbf{x})\|_K] + d_{\mathcal{W}}(\mathcal{D}_S,\mathcal{D}_F)\\
&\leqslant \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_S}[\|f\circ\omega(\mathbf{x}) - f\circ\omega_S(\mathbf{x})\|_K]\\
&\quad + \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_S}[\|f\circ\omega_S(\mathbf{x}) - f\circ\omega_F(\mathbf{x})\|_K] + d_{\mathcal{W}}(\mathcal{D}_S,\mathcal{D}_F)\\
&\leqslant \mathcal{R}_{\mathcal{D}_S}(f\circ\omega, f\circ\omega_S) + \mathcal{R}_{\mathcal{D}_S}(f\circ\omega_S, f\circ\omega_F) + d_{\mathcal{W}}(\mathcal{D}_S,\mathcal{D}_F)\\
&= \mathcal{R}_{\mathcal{D}_S}(f\circ\omega) + d_{\mathcal{W}}(\mathcal{D}_S,\mathcal{D}_F) + \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_S}[\|f\circ\omega_S(\mathbf{x}) - f\circ\omega_F(\mathbf{x})\|_K]
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
\mathcal{R}_{\mathcal{D}_F}(f\circ\omega) &= \mathcal{R}_{\mathcal{D}_F}(f\circ\omega, f\circ\omega_F) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_F}[\|f\circ\omega(\mathbf{x}) - f\circ\omega_F(\mathbf{x})\|_K]\\
&= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_F}[\|f\circ\omega(\mathbf{x}) - f\circ\omega_S(\mathbf{x}) + f\circ\omega_S(\mathbf{x}) - f\circ\omega_F(\mathbf{x})\|_K]\\
&\leqslant \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_F}[|f\circ\omega(\mathbf{x}) - f\circ\omega_S(\mathbf{x})|] + \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_F}[|f\circ\omega_S(\mathbf{x}) - f\circ\omega_F(\mathbf{x})|]\\
&= \mathcal{R}_{\mathcal{D}_F}(f\circ\omega, f\circ\omega_S) + \mathcal{R}_{\mathcal{D}_F}(f\circ\omega_S, f\circ\omega_F)\\
&= \mathcal{R}_{\mathcal{D}_S}(f\circ\omega, f\circ\omega_S) + \mathcal{R}_{\mathcal{D}_F}(f\circ\omega, f\circ\omega_S)\\
&\quad - \mathcal{R}_{\mathcal{D}_S}(f\circ\omega, f\circ\omega_S) + \mathcal{R}_{\mathcal{D}_F}(f\circ\omega_S, f\circ\omega_F)\\
&\leqslant \mathcal{R}_{\mathcal{D}_S}(f\circ\omega, f\circ\omega_S) + d_{\mathcal{W}}(\mathcal{D}_S,\mathcal{D}_F) + \mathcal{R}_{\mathcal{D}_F}(f\circ\omega_S, f\circ\omega_F)\\
&= \mathcal{R}_{\mathcal{D}_S}(f\circ\omega) + d_{\mathcal{W}}(\mathcal{D}_S,\mathcal{D}_F) + \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_F}[\|f\circ\omega_S(\mathbf{x}) - f\circ\omega_F(\mathbf{x})\|_K]
\end{aligned}$$

Therefore, the result follows. ¶

Fig. 11 provides a visualization of the margin loss function Ψ_ρ, which shows that the empirical margin loss for multi-class classification $\hat{\mathcal{R}}_{S,\rho}(f\circ\omega)$ is uuper bounded by a fraction of the sample data points that have been misclassified or correctly classified with a confidence less than or equal to ρ as in:

$$\hat{\mathcal{R}}_{S,\rho}(f\circ\omega) := \frac{1}{N}\sum_{i=1}^{N}\Psi_\rho\left(\xi^{\omega}_{\mathcal{D}_S}(\mathbf{x}_i,\ y_i)\right) \leqslant \frac{1}{m}\sum_{i=1}^{N}\mathbf{1}_{\xi^{\omega}_{\mathcal{D}_S}(\mathbf{x}_i,\ y_i)\leqslant\rho} = \hat{\mathcal{R}}_{\mathcal{D}_S}(f\circ\omega)$$

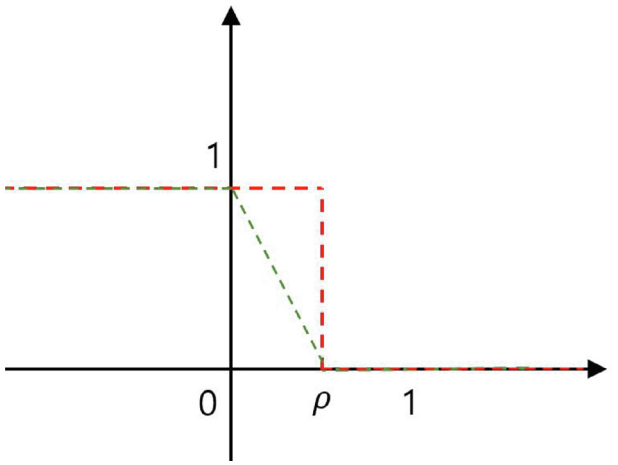


Fig. 11. The margin loss, defined with respect to margin parameter ρ.

Theorem 5. *Let $\langle \mathcal{D}_S, c_S = f \circ \omega_S \rangle$ and $\langle \mathcal{D}_F, c_F = f \circ \omega_F \rangle$ with $\omega_S = \omega_F$ be the sample and the fair correspondence distributions, respectively. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following fair clustering generalization bound holds for all hypothesis $f \circ \omega$ where $\omega \in \mathcal{W}$:*

$$\mathcal{R}_{\mathcal{D}_F}(f \circ \omega) \leqslant \hat{\mathcal{R}}_{S,\rho}(f \circ \omega) + \frac{4K}{\rho}\mathfrak{R}_N\left(\Lambda_{\mathcal{D}_S}(\mathcal{W})\right) + 2\sqrt{\frac{\log\frac{2}{\delta}}{2m}} + d_{\mathcal{H}\Delta\mathcal{H}}\left(\hat{\mathcal{D}}_S^N, \hat{\mathcal{D}}_F^N\right) + 4\sqrt{\frac{d\log\frac{em}{2d}}{m}}$$

where $d = \mathrm{VCdim}(\mathcal{H})$.

Proof 2. We first derive a margin upper bound of $\mathcal{R}_{\mathcal{D}_S}(f \circ \omega)$. Observe that

$$\xi^{\omega,+}_{\mathcal{D}_S}(\mathbf{x},\ y) := 1_{y=f\circ\omega(\mathbf{x})} - \max_{y\prime\in\mathcal{Y}_K}\left(1_{y\prime=f\circ\omega(\mathbf{x})} - 2\rho 1_{y\prime=y}\right) = \begin{cases} \min\{2\rho, 1\} & \text{if } y = f\circ\omega(\mathbf{x}) \\ -1 & \text{otherwise} \end{cases}$$

We have $\xi^{\omega,+}_{\mathcal{D}_S}(\mathbf{x},\ y) \leqslant \xi^{\omega}_{\mathcal{D}_S}(\mathbf{x},\ y)$ and so $\mathbb{E}\left[1_{\xi^{\omega}_{\mathcal{D}_S}(\mathbf{x},\ y)}\right] \leqslant \mathbb{E}\left[1_{\xi^{\omega,+}_{\mathcal{D}_S}(\mathbf{x},\ y)}\right]$.

Also we have $\Psi_\rho\left(\xi^{\omega,+}_{\mathcal{D}_S}(\mathbf{x}_i,\ y_i)\right) = \Psi_\rho\left(\xi^{\omega}_{\mathcal{D}_S}(\mathbf{x}_i,\ y_i)\right)$ by the definition of Ψ_ρ.

Now let $\tilde{\mathcal{F}}^{\xi}_{\mathcal{D}_S} = \left\{(\mathbf{x},\ y) \mapsto \xi^{\omega,+}_{\mathcal{D}_S}(\mathbf{x}, y) :\ \omega \in \mathcal{W}\right\}$. By Theorem 3.3 in [26], for any $\delta > 0$, with probability at least $1 - \delta$, for all $\omega \in \mathcal{W}$:

$$\mathbb{E}\left[\Psi_\rho\left(\xi^{\omega,+}_{\mathcal{D}_S}(\mathbf{x}, y)\right)\right] \leqslant \frac{1}{N}\sum_{i=1}^{N}\Psi_\rho\left(\xi^{\omega,+}_{\mathcal{D}_S}(\mathbf{x}_i,\ y_i)\right) + 2\mathfrak{R}_N\left(\Psi_\rho \circ \tilde{\mathcal{F}}^{\xi}_{\mathcal{D}_S}\right) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

The slope of the function Ψ_ρ defining the margin loss is at most $1/\rho$, thus Ψ_ρ is $1/\rho$-Lipschitz and so $\mathfrak{R}_N\left(\Psi_\rho \circ \tilde{\mathcal{F}}^{\xi}_{\mathcal{D}_S}\right) \leqslant \frac{1}{\rho}\mathfrak{R}_N\left(\tilde{\mathcal{F}}^{\xi}_{\mathcal{D}_S}\right)$ by Talagrand's lemma. Since $1_{u\leqslant 0} \leqslant \Psi_\rho(u)$ for all $u \in \mathbb{R}, \mathcal{R}_{\mathcal{D}_S}(f \circ \omega) = \mathbb{E}\left[1_{\xi^{\omega}_{\mathcal{D}_S}(\mathbf{x},\ y)}\right] \leqslant \mathbb{E}\left[1_{\xi^{\omega,+}_{\mathcal{D}_S}(\mathbf{x},\ y)}\right] \leqslant \mathbb{E}\left[\Psi_\rho\left(\xi^{\omega,+}_{\mathcal{D}_S}(\mathbf{x}, y)\right)\right]$, and $\Psi_\rho\left(\xi^{\omega,+}_{\mathcal{D}_S}(\mathbf{x}_i,\ y_i)\right) = \Psi_\rho\left(\xi^{\omega}_{\mathcal{D}_S}(\mathbf{x}_i,\ y_i)\right)$, we have for any $\delta > 0$, with probability at least $1 - \delta$, for all $\omega \in \mathcal{W}$:

$$\mathcal{R}_{\mathcal{D}_S}(f \circ \omega) \leqslant \frac{1}{N}\sum_{i=1}^{N}\Psi_\rho\left(\xi^{\omega}_{\mathcal{D}_S}(\mathbf{x}_i,\ y_i)\right) + \frac{2}{\rho}\mathfrak{R}_N\left(\tilde{\mathcal{F}}^{\xi}_{\mathcal{D}_S}\right) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

Next to derive an upper bound of $\mathfrak{R}_N\left(\tilde{\mathcal{F}}^{\xi}_{\mathcal{D}_S}\right)$, we observe the following, similarly to the proof of Theorem 9.2 in [26]:

$$\begin{aligned}\mathfrak{R}_N\left(\tilde{\mathcal{F}}^{\xi}_{\mathcal{D}_S}\right) &= \tfrac{1}{N}\mathbb{E}_{S,\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{N}\sigma_i\xi^{\omega,+}_{\mathcal{D}_S}(\mathbf{x}_i,y_i)\right]\\ &= \tfrac{1}{N}\mathbb{E}_{S,\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{N}\sigma_i\left(1_{y_i=f\circ\omega(\mathbf{x}_i)}-\max_{y\in\mathcal{Y}_K}\left(1_{y=f\circ\omega(\mathbf{x}_i)}-2\rho 1_{y=y_i}\right)\right)\right]\\ &\leqslant \tfrac{1}{N}\mathbb{E}_{S,\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{N}\sigma_i 1_{y_i=f\circ\omega(\mathbf{x}_i)}\right]+\tfrac{1}{N}\mathbb{E}_{S,\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{N}\sigma_i\max_{y\in\mathcal{Y}_K}\left(1_{y=f\circ\omega(\mathbf{x}_i)}-2\rho 1_{y=y_i}\right)\right]\end{aligned}$$

The first term in the right hand side of the above inequality is upper-bounded as follows:

$$\begin{aligned}\tfrac{1}{N}\mathbb{E}_{\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{N}\sigma_i 1_{y_i=f\circ\omega(\mathbf{x}_i)}\right] &= \tfrac{1}{N}\mathbb{E}_{\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{N}\sum_{y\in\mathcal{Y}_K}\sigma_i 1_{y=f\circ\omega(\mathbf{x}_i)}1_{y=y_i}\right]\\ &\leqslant \tfrac{1}{N}\sum_{y\in\mathcal{Y}_K}\mathbb{E}_{\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{N}\sigma_i 1_{y=f\circ\omega(\mathbf{x}_i)}1_{y=y_i}\right]\\ &= \tfrac{1}{N}\sum_{y\in\mathcal{Y}_K}\mathbb{E}_{\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{N}\sigma_i 1_{y=f\circ\omega(\mathbf{x}_i)}\left(\tfrac{2\cdot 1_{y=y_i}-1}{2}+\tfrac{1}{2}\right)\right]\\ &\leqslant \tfrac{1}{2m}\sum_{y\in\mathcal{Y}_K}\mathbb{E}_{\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{N}\sigma_i\mathcal{R}_i 1_{y=f\circ\omega(\mathbf{x}_i)}\right]\\ &\quad+\tfrac{1}{2m}\sum_{y\in\mathcal{Y}_K}\mathbb{E}_{\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{N}\sigma_i 1_{y=f\circ\omega(\mathbf{x}_i)}\right]\\ &= \sum_{y\in\mathcal{Y}_K}\tfrac{1}{N}\mathbb{E}_{\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{N}\sigma_i 1_{y=f\circ\omega(\mathbf{x}_i)}\right]\leqslant\sum_{y\in\mathcal{Y}_K}\hat{\mathfrak{R}}_{\mathcal{D}_S}\left(\Lambda_{\mathcal{D}_S}(\mathcal{W})\right)\end{aligned}$$

where $\mathcal{R}_i := 2\cdot 1_{y=y_i}-1\in\{-1,\ +1\}$ and σ_i and $\sigma_i\mathcal{R}_i$ admit the same distribution. Therefore we have

$$\frac{1}{N}\mathbb{E}_{S,\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{N}\sigma_i 1_{y_i=f\circ\omega(\mathbf{x}_i)}\right]\leqslant K\mathfrak{R}_N\left(\Lambda_{\mathcal{D}_S}(\mathcal{W})\right)$$

The second term in the right hand side of the above inequality is upper-bounded as follows. Observing that $-\sigma_i$ and σ_i are distributed in the same way and using the sub-additivity of sup and Lemma 9.1 in [26] leads to

$$\begin{aligned}&\tfrac{1}{m}\mathbb{E}_{S,\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{m}\sigma_i\max_{y\in\mathcal{Y}_K}\left(1_{y=f\circ\omega(\mathbf{x}_i)}-2\rho 1_{y=y_i}\right)\right]\\ &\leqslant\sum_{y\in\mathcal{Y}_K}\tfrac{1}{m}\mathbb{E}_{S,\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{m}\sigma_i\left(1_{y=f\circ\omega(\mathbf{x}_i)}-2\rho 1_{y=y_i}\right)\right]\\ &=\sum_{y\in\mathcal{Y}_K}\tfrac{1}{m}\mathbb{E}_{S,\sigma}\left[\sup_{\omega\in\mathcal{W}}\left(\sum_{i=1}^{m}\sigma_i\left(1_{y=f\circ\omega(\mathbf{x}_i)}\right)-2\rho\sum_{i=1}^{m}\sigma_i 1_{y=y_i}\right)\right]=\sum_{y\in\mathcal{Y}_K}\tfrac{1}{m}\mathbb{E}_{S,\sigma}\left[\sup_{\omega\in\mathcal{W}}\sum_{i=1}^{m}\sigma_i 1_{y=f\circ\omega(\mathbf{x}_i)}\right]\leqslant K\mathfrak{R}_m\left(\Lambda_{\mathcal{D}_S}(\mathcal{W})\right)\end{aligned}$$

since σ_i have zero mean. Therefore we have the following general margin bound of $\mathcal{R}_{\mathcal{D}_S}(f\circ\omega)$

$$\mathcal{R}_{\mathcal{D}_S}(f\circ\omega)\leqslant\hat{\mathcal{R}}_{S,\rho}(f\circ\omega)+\frac{4K}{\rho}\mathfrak{R}_N\left(\Lambda_{\mathcal{D}_S}(\mathcal{W})\right)+2\sqrt{\frac{\log\frac{2}{\delta}}{2m}}+\sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

Finally we derive a generalization upper bound for $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S,\mathcal{D}_F)$ as follows: By Theorem 3.3 in [26], for any $\delta>0$, with probability at least $1-\delta/2$ for all $\omega\in\mathcal{W}$ (for all $g=(f\circ\omega)\oplus(f\circ\omega\prime)\in\mathcal{H}\Delta\mathcal{H}$):

$$\begin{aligned}d_{\mathcal{W}}(\mathcal{D}_S,\hat{\mathcal{D}}^N_S) &= \sup_{\omega,\omega'\in\mathcal{H}}|\mathcal{R}_{\mathcal{D}_S}(f\circ\omega,f\circ\omega')-\hat{\mathcal{R}}_{\hat{\mathcal{D}}^N_S}(f\circ\omega,f\circ\omega')|=\sup_{g\in\mathcal{W}}|\mathbb{E}_{\mathcal{D}_S}[1_g]-\hat{\mathbb{E}}_{\hat{\mathcal{D}}^N_S}[1_g]|\\ &\leqslant 2\mathfrak{R}_{\mathcal{D}_S}(\mathcal{W})+\sqrt{\tfrac{\log\frac{2}{\delta}}{2m}}.\end{aligned}$$

Also similarly we have, for any $\delta>0$, with probability at least $1-\delta/2$ for all $\omega\in\mathcal{W}$:

$$d_{\mathcal{W}}(\mathcal{D}_F,\hat{\mathcal{D}}^m_F)\leqslant 2\mathfrak{R}_{\mathcal{D}_F}(\mathcal{W})+\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

Utilizing the triangle inequality and symmetry property of $d_{\mathcal{W}}(\cdot,\cdot)$, we have, for any $\delta>0$, with probability at least $1-\delta$ for all $\omega\in\mathcal{W}$:

$$\begin{aligned} d_{\mathcal{W}}(\mathcal{D}_S, \mathcal{D}_F) &\leqslant d_{\mathcal{W}}(\mathcal{D}_S, \hat{\mathcal{D}}_S^N) + d_{\mathcal{W}}(\hat{\mathcal{D}}_S^N, \hat{\mathcal{D}}_F^N) + d_{\mathcal{W}}(\hat{\mathcal{D}}_F^N, \mathcal{D}_F) \\ &\leqslant d_{\mathcal{W}}(\hat{\mathcal{D}}_S^N, \hat{\mathcal{D}}_F^N) + 2\mathfrak{R}_{\mathcal{D}_S}(\mathcal{W}) + 2\mathfrak{R}_{\mathcal{D}_F}(\mathcal{W}) + 2\sqrt{\frac{\log\frac{2}{\delta}}{2m}} \end{aligned}$$

Observe that $\mathrm{VCdim}(\mathcal{W}) \leqslant 2\mathrm{VCdim}(\mathcal{H})$ as in [2] since any $\omega \in \mathcal{W}$ can be represented as a linear threshold network of depth 2 with 2 hidden units. Note also that $2\omega(\mathbf{x}) - 1 \in \{-1, 1\}$ since $\omega(\mathbf{x}) \in \{0, 1\}$. Therefore this result combined with Corollary 3.9 of [26] leads to

$$\mathfrak{R}_{\mathcal{D}_S}(\mathcal{W}) \leqslant \frac{1}{2}\sqrt{\frac{4d\log\frac{em}{2d}}{m}}, \quad \text{and} \quad \mathfrak{R}_{\mathcal{D}_F}(\mathcal{W}) \leqslant \frac{1}{2}\sqrt{\frac{4d\log\frac{em}{2d}}{m}}$$

where $d = \mathrm{VCdim}(\mathcal{H})$.

Remark: (i) The first term is the sample error caused by a discrepancy between the ground-truth sample density and the approximate support density. This is expected to be small with a large sample size, as the estimated density converges asymptotically to the true support density. The fourth term vanishes when $\omega_S = \omega_F$. Finally, the rest of the terms asymptotically converges to zero, when the number of samples N is sufficiently large and the ρ converges to 1. In contrast, the bound becomes loose when the empirical sample size is small and when the support density does not reflect the ground-truth sample density.

(ii) Under the restriction $\omega_S = \omega_F$, it is well-known in many literatures (cf. [26]) that the generalization upper bound in the multi-class kernel-based learning machines becomes: for any $\delta > 0$, with probability at least $1 - \delta$, for all $h \in \mathcal{H}_{K,2}$ $\left\{(x, y) \to \mathbf{w}_y \cdot \Phi(x) : W = (w_1, \ldots, w_K)^T, \sum_{i=1}^K \|\mathbf{w}_i\|^2 \leqslant \Delta^2\right\}$,

$$\mathcal{R}(h) \leqslant \frac{1}{N}\sum_{i=1}^N \xi_i + 4K\sqrt{\frac{r^2\Delta^2}{N}} + \sqrt{\frac{\log\frac{1}{\delta}}{2N}}, \tag{A.2}$$

where $\xi_i = \max\left(1 - \left[\mathbf{w}_{y_i} \cdot \Phi(x_i) - \max_{y' \neq y_i} \mathbf{w}_{y'} \cdot \Phi(x_i)\right], 0\right)$ for $i = 1, \ldots, N$. We can find that under the restriction $\omega_S = \omega_F$, the second and third term asymptotically converges to zero as $N \to \infty$, which shows that the upper bound is asymptotically tight. However, when the source and the target distribution is different, the upper bound cannot be tight as expected!.

Theorem 6. *Let a sample $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be generated by a distribution $\mathcal{D}_S$ with a sample support density $p_S(\mathbf{x})$. Assume the following conditions are satisfied,*

$$\lim_{N\to\infty} q_N = \infty \quad \text{and} \quad \lim_{N\to\infty} NC_N^2 q_N^{d/2} = 0.$$

Then the estimate $p_N(\mathbf{x})$ of the form

$$p_N(\mathbf{x}) = (q_N/\pi)^{n/2} \sum_{i=1}^N \bar{\beta}_i e^{-q_N\|\mathbf{x}-\mathbf{x}_i\|^2},$$

converges asymptotically to $p_S(\mathbf{x})$.

Proof 3. To show the asymptotic convergence to $p_S(\mathbf{x})$, we need to show the followings.

$$\lim_{N\to\infty} \mathbb{E}[p_N(\mathbf{x})] = p_S(\mathbf{x})$$
$$\lim_{N\to\infty} \mathrm{Var}[p_N(\mathbf{x})] = 0$$

First notice that for any fixed $\mathbf{x}$ the value of $p_N(\mathbf{x})$ depends on the i.i.d. random samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$. If we let $\delta_N(\mathbf{x} - \mathbf{x}_i) = (q_N/\pi)^{d/2} e^{-q_N\|\mathbf{x}-\mathbf{x}_i\|^2}$, then $\delta_N(\mathbf{x} - \mathbf{x}_i) \to 0$ if $\mathbf{x} \neq \mathbf{x}_i$, with normalization $\int \delta_N(\mathbf{x} - \mathbf{x}_i) d\mathbf{x} = 1$ since $q_N \to \infty$ as $N \to \infty$. Thus we have $\delta_N(\mathbf{x} - \mathbf{x}_i)$ approaches a Dirac delta function $\delta(\mathbf{x} - \mathbf{x}_i)$ as $N \to \infty$. Therefore we get

$$\begin{aligned} \lim_{N\to\infty} \mathbb{E}[p_N(\mathbf{x})] &= \lim_{N\to\infty} \sum_{i=1}^N \bar{\beta}_i \mathbb{E}[\delta_N(\mathbf{x} - \mathbf{x}_i)] = \sum_{i=1}^N \bar{\beta}_i \lim_{N\to\infty} \int \delta_N(\mathbf{x} - \mathbf{v}) p_S(\mathbf{v}) d\mathbf{v} \\ &= \left(\sum_{i=1}^N \bar{\beta}_i\right) \int \delta(\mathbf{x} - \mathbf{v}) p_S(\mathbf{v}) d\mathbf{v} = p_S(\mathbf{x}) \end{aligned}$$

Furthermore, since for any fixed $\mathbf{x}$, $p_N(\mathbf{x})$ is the weighted sum of functions of statistically independent random variables, i.e. $p_N(\mathbf{x}) = \sum_{i=1}^N \bar{\beta}_i (q_N/\pi)^{d/2} e^{-q_N\|\mathbf{x}-\mathbf{x}_i\|^2}$, the variance is

$$\begin{aligned}\lim_{N\to\infty}\mathrm{Var}[p_N(\mathbf{x})] &= \lim_{N\to\infty}\left(\sum_{i=1}^{N}\mathbb{E}\left[\bar{\beta}_i^2(q_N/\pi)^d e^{-2q_N\|\mathbf{x}-\mathbf{x}_i\|^2}\right]-\frac{1}{N}\mathbb{E}[p_N(\mathbf{x})]^2\right)\\ &\leqslant \lim_{N\to\infty}\left(\max_i\bar{\beta}_i\right)^2(q_N/2\pi)^{d/2}N\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}(2q_N/\pi)^{d/2}e^{-2q_N\|\mathbf{x}-\mathbf{x}_i\|^2}\right]\\ &\leqslant \lim_{N\to\infty}NC_N^2(q_N/2\pi)^{d/2}\mathbb{E}[p_S(\mathbf{x})]=0\end{aligned}$$

since $\max_i\bar{\beta}_i\leqslant C_N$ and $NC_N^2q_N^{d/2}\to 0$ as $N\to\infty$.

Since the dual optimal solutions $^{=}\beta_j$ where $(C,q)=(C_N,q_N)$ are in the set $S_{\bar{\beta}}^N=\left\{\bar{\beta}=\left(\bar{\beta}_1,\ldots\bar{\beta}_N\right)^T:\sum_{i=1}^N\bar{\beta}_i=1,\ 0\leqslant\bar{\beta}_i\leqslant C_N\right\}$ and the volume of the set $S_{\bar{\beta}}^N$ shrinks to zero as $C_N\to\infty$, controlling the parameters (C_N,q_N) as above leads to the decision function such as

$$\begin{aligned}f(\mathbf{x}) &= 1-2\sum_{i=1}^{N}\bar{\beta}_i e^{-q_N\|\mathbf{x}-\mathbf{x}_i\|^2}\\ &\quad+\sum_{i,j=1}^{N}\bar{\beta}_i\bar{\beta}_j e^{-q_N\|\mathbf{x}_i-\mathbf{x}_j\|^2}.\end{aligned}$$

converges to an unknown density function up to a constant multiple.

Appendix B. Additional results

See Figs. 12 and 13.

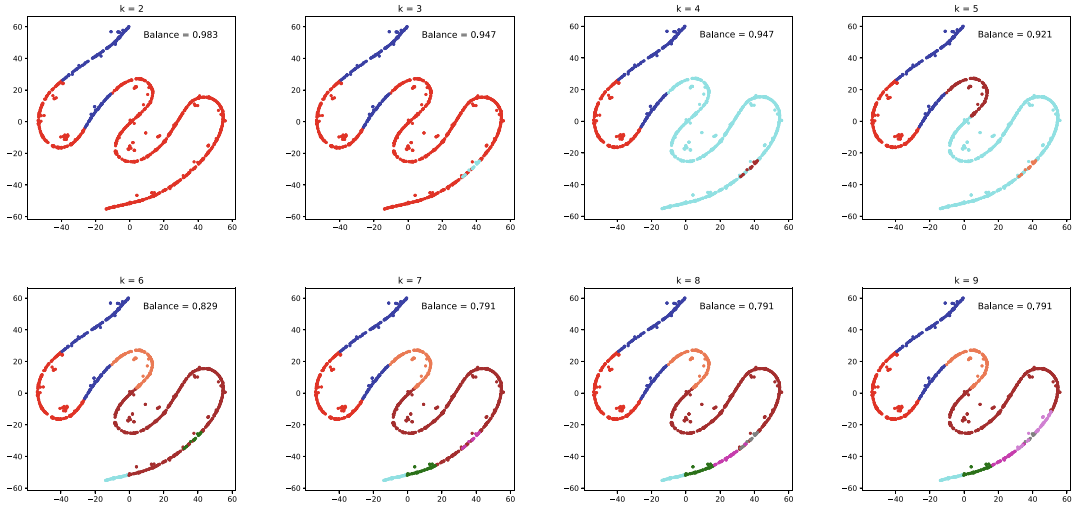


Fig. 12. Visualization of the clustering results of census dataset of our method with varying number of clusters from k = 2 to 9. We used maximum number of samples for census dataset.

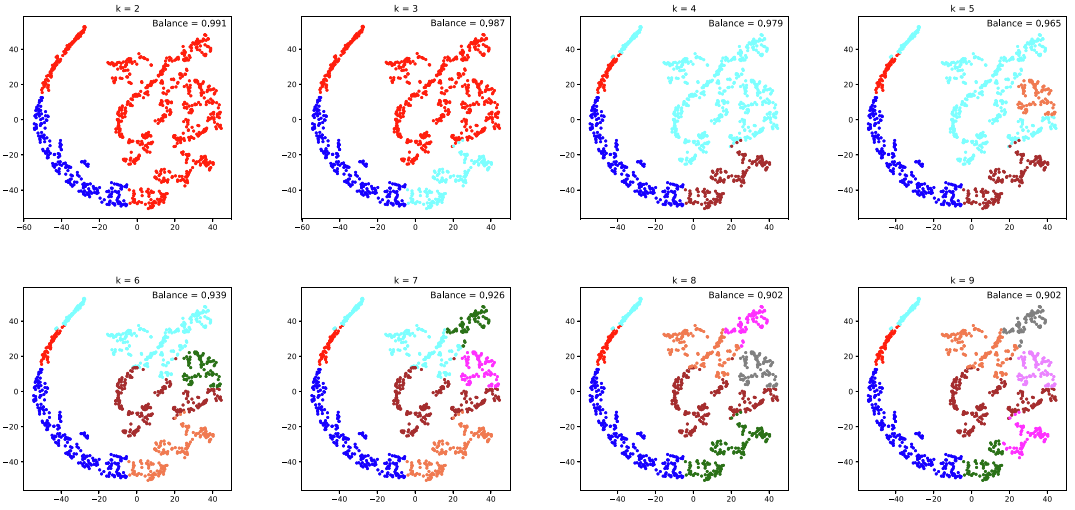


Fig. 13. Visualization of the clustering results of bank dataset of our method with varying number of clusters from $k = 2$ to 9. We used maximum number of samples for bank dataset.

References

[1] J. Angwin, J. Larson, S. Mattu, L. Kirchner, Machine bias. ProPublica, May 23, 2016..
[2] M. Anthony, P.L. Bartlett, Neural Network Learning: Theoretical Foundations, Cambridge University Press, 2009.
[3] A. Backurs, P. Indyk, K. Onak, B. Schieber, A. Vakilian, T. Wagner, Scalable fair clustering, in: International Conference on Machine Learning, 2019, pp. 405–413.
[4] A. Banerjee, P. Maji, Stomped-t: a novel probability distribution for rough-probabilistic clustering, Information Sciences 421 (2017) 104–125.
[5] S. Barocas, A.D. Selbst, Big data's disparate impact, Calif. L. Rev. 104 (2016) 671.
[6] A. Ben-Hur, D. Horn, H.T. Siegelmann, V. Vapnik, Support vector clustering, Journal of Machine Learning Research 2 (2001) 125–137.
[7] S.K. Bera, D. Chakrabarty, M. Negahbani, Fair algorithms for clustering, 2019, arXiv preprint arXiv:1901.02393..
[8] T. Bolukbasi, K.W. Chang, J.Y. Zou, V. Saligrama, A.T. Kalai, Man is to computer programmer as woman is to homemaker? debiasing word embeddings, in: Advances in Neural Information Processing Systems, 2016, pp. 4349–4357.
[9] J. Buolamwini, T. Gebru, Gender shades: Intersectional accuracy disparities in commercial gender classification, in: Conference on Fairness, Accountability and Transparency, 2018, pp. 77–91..
[10] A. Cena, M. Gagolewski, Genie+ owa: Robustifying hierarchical clustering with owa-based linkages, Information Sciences (2001).
[11] F. Chierichetti, R. Kumar, S. Lattanzi, S. Vassilvitskii, Fair clustering through fairlets, in: Advances in Neural Information Processing Systems, 2017, pp. 5029–5037.
[12] S.C. Chu, J.F. Roddick, C.J. Su, J.S. Pan, Constrained ant colony optimization for data clustering, in: Pacific Rim International Conference on Artificial Intelligence, Springer, 2004, pp. 534–543.
[13] M. Feldman, S.A. Friedler, J. Moeller, C. Scheidegger, S. Venkatasubramanian, Certifying and removing disparate impact, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 259–268.
[14] S.F. Galán, Comparative evaluation of region query strategies for dbscan clustering, Information Sciences 502 (2019) 76–90.
[15] M. Hardt, E. Price, N. Srebro, et al, Equality of opportunity in supervised learning, in: Advances in Neural Information Processing Systems, 2016, pp. 3315–3323.
[16] L. Huang, S. Jiang, N. Vishnoi, Coresets for clustering with fairness constraints, in: Advances in Neural Information Processing Systems, 2019, pp. 7587–7598.
[17] K.H. Jung, N. Kim, J. Lee, Dynamic pattern denoising method using multi-basin system with kernels, Pattern Recognition 44 (2011) 1698–1707.
[18] K. Kim, J. Lee, Nonlinear dynamic projection for noise reduction of dispersed manifolds, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (2014) 2303–2309.
[19] K. Kim, Y. Son, J. Lee, Voronoi cell-based clustering using a kernel support, IEEE Transactions on Knowledge and Data Engineering 27 (2014) 1146–1156.
[20] D. Lee, J. Lee, Domain described support vector classifier for multi-classification problems, Pattern Recognition 40 (2007) 41–51.
[21] D. Lee, J. Lee, Dynamic dissimilarity measure for support-based clustering, IEEE Transactions on Knowledge and Data Engineering 22 (2009) 900–905.
[22] J. Lee, D. Lee, An improved cluster labeling method for support vector clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 461–464.
[23] J. Lee, D. Lee, Dynamic characterization of cluster structures for robust and inductive support vector clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 1869–1874.
[24] B. Liu, M. Xu, P. Fu, Graph-based clustering with spatiotemporal contour energy for video salient object detection, J. Inf. Hiding Multim. Signal Process. 10 (2019) 359–367.
[25] D. Madras, E. Creager, T. Pitassi, R. Zemel, Learning adversarially fair and transferable representations, 2018, arXiv preprint arXiv:1802.06309..
[26] M. Mohri, A. Rostamizadeh, A. Talwalkar, Foundations of Machine Learning, MIT press, 2018.
[27] J. Ristori, C. Cocchetti, A. Romani, F. Mazzoli, L. Vignozzi, M. Maggi, A.D. Fisher, Brain sex differences related to gender identity development: Genes or hormones?, International Journal of Molecular Sciences 21 (2020) 2123
[28] B. Schölkopf, A.J. Smola, F. Bach, et al, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, 2002.
[29] D.M. Tax, R.P. Duin, Support vector domain description, Pattern Recognition Letters 20 (1999) 1191–1199.
[30] T. Yoon, J. Lee, W. Lee, Joint transfer of model knowledge and fairness over domains using wasserstein distance, IEEE Access 8 (2020) 123783–123798.
[31] M.B. Zafar, I. Valera, M. Rodriguez, K. Gummadi, A. Weller, From parity to preference-based notions of fairness in classification, in: Advances in Neural Information Processing Systems, 2017, pp. 229–239.