Fairness-aware Multi-view Clustering

Lecheng Zheng* Yada Zhu† Jingrui He*

Abstract

In the era of big data, we are often facing the challenge of data heterogeneity and the lack of label information simultaneously. In the financial domain (e.g., fraud detection), the heterogeneous data may include not only the numerical data (e.g., total debt and yearly income), but also text and images (e.g., financial statement and invoice images). At the same time, the label information (e.g., fraud transactions) may be missing for building predictive models. To address these challenges, many state-of-the-art multi-view clustering methods have been proposed and achieved outstanding performance. However, these methods typically do not take into consideration the fairness aspect, and are likely to generate biased results using sensitive information such as race and gender. Therefore, in this paper, we propose a fairness-aware multi-view clustering method named Fair-MVC. It incorporates the group fairness constraint into the soft membership assignment for each cluster to ensure that the fraction of different groups in each cluster is approximately identical to the entire data set. Meanwhile, we adopt the idea of both contrastive learning and non-contrastive learning, and propose novel regularizers to handle heterogeneous data in complex scenarios with missing data or noisy features. Experimental results on real-world data sets demonstrate the effectiveness and efficiency of the proposed framework. We also derive insights regarding the relative performance of the proposed regularizers in various scenarios.

1 Introduction

In the era of big data, the volume of data grows at an unprecedented rate. Compared with homogeneous data in the past, nowadays, the data collected from many real-world applications usually exhibit the nature of heterogeneity (e.g., view heterogeneity). For instance, on social media, one or two decades ago, users shared their daily lives with others mainly via text data; but with the development of electronic devices, users tend to share their experiences by a mixture of multiple types of data, such as a recorded video or several photos along with the text description. Another example is in the financial domain. Take fraud detection as an example: the heterogeneous data may include not only the numerical data (e.g., total debt and yearly income) but also text and images (e.g., financial statements and invoice images). On the other hand, with the advent of big data across multiple high-impact domains, the label information is largely lacking. This phenomenon may be caused by the expensive labeling cost or the mismatch between the speed of generating data and labeling data [25]. Regardless of the reasons behind this phenomenon, exploring and analyzing these newly-created data is urgent in many domains [10, 17]. To address this problem, many state-of-the-art (SOTA) multi-view clustering algorithms have been proposed, including the earliest work (e.g., Co-EM algorithms [4, 26], Canonical Correlation Analysis-based clustering methods [3, 5]) and current deep learning based methods [14, 33, 44]. In addition, the collected data sometimes consist of missing entries or noisy data. However, many existing SOTA algorithms [3, 5, 14, 33, 39, 40] fail to effectively handle such complex scenarios. For instance, Figure 1 shows the performance of SOTA methods in terms of normalized mutual information score on the Credit Card data set [35] in the missing feature scenario. In particular, the x-axis is the percentage of the missing features and the y-axis is the normalized mutual information score. By observation, when the percentage of missing features increases, the performance of these state-of-the-art methods (e.g., DEC, LF-IMVC, CC) starts to decrease dramatically, suggesting that they couldn't effectively handle the missing feature scenario.

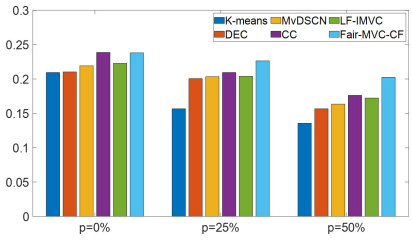


Figure 1: Performance of SOTA methods on the Credit Card data set in missing feature scenario, where p denotes the percentage of missing features.

*University of Illinois Urbana-Champaign, {lecheng4, jingrui}@illinois.edu

†MIT-IBM Watson AI Lab, IBM Researh, yzhu@us.ibm.com

On the other hand, the collected data may contain sensitive information (e.g., race, gender) in many domains. The straightforward application of existing machine learning algorithms may render severely biased results [11]. For instance, when analyzing whether a bank should increase the interest rate for a credit card holder, some sensitive information, such as race and gender, should be excluded from the algorithms. In other words, these algorithms are expected to achieve good performance while satisfying the fairness constraint. Despite the outstanding performance of these aforementioned methods for addressing their respective problems [14, 19, 27, 32, 33, 38], most (if not all) of these multi-view clustering methods only aim to improve the performance, and thus fail to consider the fairness constraint. Besides, though the existing fair single view clustering methods [7, 18, 37] achieve the excellent performance, we couldn't directly apply them to handle multi-view data sets as a study [34] shows that simply concatenating multiple views into one feature vector may lead to sub-optimal solution.

To fill in this gap, in this paper, we propose a fairness-aware multi-view clustering method named Fair-MVC. It seamlessly integrates the fairness constraint into the clustering process of multi-view data. More specifically, Fair-MVC maximizes the mutual agreement of the soft membership assignment from each view to generate the clusters. In the meanwhile, it incorporates the group fairness constraint into the soft membership assignment for each cluster to ensure that the fraction of different groups in each cluster is approximately identical to the fraction in the whole data set. In addition, to handle heterogeneous data in complex scenarios with missing data or noisy features, we adopt the idea of contrastive learning and non-contrastive learning and propose novel regularizers.

Our main contributions are summarized below.

- We formalize a new problem setting: fairness-aware multi-view clustering;
- We propose novel contrastive and non-contrastive regularizations to handle complex scenarios with missing data or noisy features;
- We provide insights regarding the relative performance of contrastive and non-contrastive regularizers in various scenarios;
- Experimental results on both synthetic and real-world data sets demonstrate the effectiveness and efficiency of the proposed framework.

The rest of this paper is organized as follows. After a brief review of the related work in Section 2, we introduce the problem definition and our proposed framework to address this problem in Section 3. The systematic evaluation of the proposed framework on both synthetic and real-world data sets is presented in Section 4 before we conclude the paper in Section 5.

2 Related Work

In this section, we briefly review the related works.

Multi-view Clustering: Multi-view clustering has been studied for decades. Starting from the earliest work, such as Co-EM algorithms [4, 26], Canonical Correlation Analysis-based clustering methods [3, 5], to current works [19, 27], more and more researchers pay attention to deep multi-view clustering [14, 33] due to the great performance to handle various types of data. [21] proposed a novel multi-view clustering method in the adversarial setting by learning the latent representation with an auto-encoder and capturing the data distribution with adversarial training. However, all of these neglect the importance of fairness and to bridge the gap, we propose the fairness-aware multi-view clustering method, which incorporates group fairness into our proposed multi-view clustering algorithm.

Fairness Machine Learning: Recent year has witnessed the surge of the fairness machine learning algorithms [1, 7, 13, 18, 36, 37]. [37] considered both group fairness and individual fairness by encoding the input data as well as fairness constraint into a latent space and meanwhile obfuscating the membership information. [13] proposed a fairness measure against sensitive attributes in the classification problem to ensure equal opportunity for both protected and unprotected groups. [12] introduced a fairness measure for classification problems and provided theoretical results to demonstrate the effectiveness of the test for disparate impact on real-world datasets. Different from these fairness algorithms, we propose a novel fairness-aware clustering algorithm in a more sophisticated setting by considering the data heterogeneity.

Contrastive Learning: Recently, contrastive learning has exhibited outstanding performance by modeling the data without supervision. One of the earliest works [29] proposes the contrastive predictive coding framework (Info-NCE) to extract a compact lower-dimensional representation to maximize the mutual information between the hidden representation of the input data and the targeted signal. Recent studies [6, 15, 22, 28, 41, 42] reveal a surge of research interest in contrastive learning. [28] extended contrastive coding to a multi-view setting by maximizing the mutual information between each pair of views. [8] addressed the drawbacks of contrastive learning-based methods by removing the negative pairs and only maximizing the similarity of positive

pairs. Nevertheless, directly combining the current contrastive learning with the multi-view clustering method may lead to sub-optimal performance in some specific scenarios To address this issue, we propose novel contrastive and non-contrastive regularizations, which enable our proposed method to handle the perturbed data in more sophisticated scenarios.

3 Proposed Fair-MVC Framework

In this section, we present our proposed Fairness-Aware Multi-view Clustering (Fair-MVC) framework. We first introduce the major notation and the problem definition; then we discuss the proposed Fair-MVC framework along with the regularization terms. Finally, we provide the overall objective function.

3.1 Notation and Problem Definition In this paper, we denote $\mathcal{D} = \{\boldsymbol{X^1}, \boldsymbol{X^2}, ..., \boldsymbol{X^v}, \boldsymbol{R}\}$ as a data set with $\boldsymbol{V}$ views and n samples, where $\boldsymbol{X^i} \in \mathbb{R}^{n \times d_i}$ is the input feature matrix for the i^{th} view, $\boldsymbol{R} \in \mathbb{R}^{n \times d_r}$ is the sensitive features (e.g., race, gender, etc.), d_r is the dimensionality of sensitive features, and d_i is the dimensionality of the input features for the i^{th} view. We aim to assign the n samples into k clusters with the membership matrix $\boldsymbol{Q^v} \in \mathbb{R}^{n \times k}$, each represented by a centroid $\boldsymbol{\mu}_j^v \in \mathbb{R}^d, j = 1, ..., k$, where d is the dimensionality of the centroid. Instead of clustering these samples directly in the input space, we propose to first transform these samples with a non-linear mapping $\boldsymbol{f^v} : \boldsymbol{X^v} \to \boldsymbol{Z^v}$, i.e., $\boldsymbol{Z^v} = \boldsymbol{f^v}(\boldsymbol{X^v})$, where $\boldsymbol{Z^v} \in \mathbb{R}^d$ is the latent representation for the v^{th} view. We denote x_i as the i^{th} sample and z_i as the hidden representation of x_i. Throughout this paper, we use $\boldsymbol{x_i^j}$ to denote the j^{th} view of the i^{th} sample in $\boldsymbol{X^j}$, $\boldsymbol{z_i^j}$ to denote the representation of the sample $\boldsymbol{x_i^j}$ and $\boldsymbol{r_i}$ to denote the sensitive feature of the the i^{th} sample. For the ease of explanation, we only consider two views in the next few subsections, although our proposed method could be naturally extended to multiple views. With all the aforementioned notation, we are ready to formalize the fairness-aware multi-view clustering problem as follows.

Problem 1. ***Fairness-aware Multi-view Clustering***

Input: *a set of unlabeled data $\mathcal{D}$ along with the sensitive features $\boldsymbol{R}$ and the number of the clusters k.*

Output: *: the membership matrix $\boldsymbol{Q}$ for each sample in $\mathcal{D}$ with the fairness constraint.*

3.2 Fairness-Aware Multi-view Clustering Following the strategy in [32], we measure the similarity between the hidden representation $\boldsymbol{z_i^v}$ and centroid μ_j as follows.

$$\boldsymbol{q}_{ij}^v = \frac{e^{sim(\boldsymbol{z_i^v}, \boldsymbol{\mu_j^v})}}{\sum_{j'} e^{sim(\boldsymbol{z_i^v}, \boldsymbol{\mu_{j'}^v})}} \tag{3.1}$$

where $sim(\boldsymbol{z_i^v}, \boldsymbol{\mu_j^v}) = -|\boldsymbol{z_i^v} - \boldsymbol{\mu_j^v}|^2$. Here, we denote $\boldsymbol{q}_{ij}^v$ as the element in the the i^{th} row and the j^{th} column of $\boldsymbol{Q^v}$. After getting the probability of the soft assignment, we could update the centroid via the formulation below:

$$\boldsymbol{\mu}_j^v = \frac{\sum_{i=1}^n \boldsymbol{q}_{ij}^v \boldsymbol{z_i^v}}{\sum_{i=1}^n \boldsymbol{q}_{ij}^v} \tag{3.2}$$

In many real-world applications, we want the clustering results to be fair, and to not discriminate against any protected group. For instance, when a bank makes a decision to increase the interest rate for a credit card holder, some sensitive information, (e.g., race and gender) should not be included in the algorithm but fairness measurement should be taken into consideration to ensure the fair results for its customers. Based on the above equations, to minimize the potential bias, we follow the idea proposed in [16] that each group is approximately represented with the same fraction as in the whole data set. Given the sensitive features $\boldsymbol{R}$, the group fairness constraint could be formalized as follows.

$$\begin{aligned} \boldsymbol{s_j} &= \frac{\sum_{i=1}^n \sum_{v=1}^V \boldsymbol{q}_{ij}^v \boldsymbol{r_i}}{\sum_{i=1}^n \sum_{v=1}^V \boldsymbol{q}_{ij}^v}, \boldsymbol{s_D} = \frac{1}{n} \sum_{i=1}^n \boldsymbol{r_i} \\ L_F &= \sum_{j=1}^k \|\boldsymbol{s_j} - \boldsymbol{s_D}\|_2^2 \end{aligned} \tag{3.3}$$

where $\boldsymbol{r_i}$ is the sensitive feature of the the i^{th} sample in $\boldsymbol{R}$, $\boldsymbol{s_j}$ represents the weighted mean of each sensitive feature in the j^{th} cluster and $\boldsymbol{s_D}$ measures the average value of each sensitive feature in the whole data set. Intuitively, minimizing L_F imposes the constraint that the fraction of sensitive features in each cluster should be close to the fraction of sensitive features in the whole data set. Besides simply adding the fairness regularization term (i.e., L_F) as a regularizer, we incorporate the fairness constraint in the soft assignment to further mitigate the potential bias as follows.

$$\begin{aligned} \boldsymbol{q}_{ij}^v &= \frac{e^{sim(\boldsymbol{z_i^v}, \boldsymbol{\mu_j^v}) + \alpha G(\boldsymbol{s}_j, \boldsymbol{s}_D, \boldsymbol{r}_i)}}{\sum_{j'} e^{sim(\boldsymbol{z_i^v}, \boldsymbol{\mu_j^v}) + \alpha G(\boldsymbol{s}_{j'}, \boldsymbol{s}_D, \boldsymbol{r}_i)}} \\ G(\boldsymbol{s}_j, \boldsymbol{s}_D, \boldsymbol{r}_i) &= ||\boldsymbol{s}_j - \kappa - \boldsymbol{s}_D||_2^2 - ||\boldsymbol{s}_j - \boldsymbol{s}_D||_2^2 \end{aligned} \tag{3.4}$$

where $\kappa = \frac{\sum_{v=1}^V \boldsymbol{q}_{ij}^v \boldsymbol{r_i}}{\sum_{i=1}^n \sum_{v=1}^V \boldsymbol{q}_{ij}^v - \sum_{v=1}^V \boldsymbol{q}_{ij}^v}$ is the re-weighted sensitive feature of the i-th sample and α is a constant parameter balancing two terms. The intuition of the

fairness constraint $G(\boldsymbol{s}_j, \boldsymbol{s}_D, \boldsymbol{r}_i)$ is straightforward. If $G(\boldsymbol{s}_j, \boldsymbol{s}_D, \boldsymbol{r}_i) > 0$, it means that removing the i-th sample from j-th cluster (i.e., $||\boldsymbol{s}_j - \kappa - \boldsymbol{s}_D||_2^2$) increases the difference between $\boldsymbol{s}_j$ and $\boldsymbol{s}_D$, and it will cause the clustering results to be unfair. Thus, we should keep the i-th sample in j-th cluster. Otherwise, we should remove the i-th sample from j-th cluster to decrease the difference.

After mitigating the bias in the soft assignment, we propose to iteratively refine the clusters by minimizing the distance between $\boldsymbol{z}_i^v$ and $\boldsymbol{\mu}_i^v$ as follows.

$$L_d = \sum_{i,j,v} \boldsymbol{c}_{ij}^v \|\boldsymbol{z}_i^v - \boldsymbol{\mu}_j^v\|_2^2 \tag{3.5}$$

where $\boldsymbol{c}_{ij}^v \in \{0, 1\}$ denotes whether the i^{th} sample belongs to the j^{th} cluster based on the v^{th} view. L_d aims to ensure that the samples belonging to the same cluster will get closer. In addition, based on the assumption [34] in multi-view learning that the information contained in each view is consistent, we aim to match the soft assignment made by the first view to the soft assignment made by the second view by minimizing the KL divergence between two distributions:

$$\begin{aligned} L_{KL} &= KL(\boldsymbol{Q^1}||\boldsymbol{Q^2}) + KL(\boldsymbol{Q^2}||\boldsymbol{Q^1}) \\ &= \sum_i \sum_j (\boldsymbol{q}_{ij}^1 \log \frac{\boldsymbol{q}_{ij}^1}{\boldsymbol{q}_{ij}^2} + \boldsymbol{q}_{ij}^2 \log \frac{\boldsymbol{q}_{ij}^2}{\boldsymbol{q}_{ij}^1}) \end{aligned} \tag{3.6}$$

where $\boldsymbol{Q^1}$ and $\boldsymbol{Q^2}$ are two soft assignment matrices.

3.3 Regularization The main idea of the unsupervised contrastive loss is to utilize the rich unlabeled data to enhance the quality of the hidden representation. Rather than directly imposing the contrastive constraint on the latent space $\boldsymbol{Z}$, we first transform $\boldsymbol{Z}$ into another space $\boldsymbol{H}$ with the second encoder g^v (e.g., $\boldsymbol{h_i^v} = g^v(\boldsymbol{z_i^v})$) by following the idea proposed in [6] to avoid distorting the hidden representation $\boldsymbol{Z}$ and then we regularize the hidden space $\boldsymbol{H}$ as follows.

$$L_1 = -\mathbb{E}_{x_i \in \mathcal{D}}[\log \frac{f(\boldsymbol{h_i^1}, \boldsymbol{h_i^2})}{f(\boldsymbol{h_i^1}, \boldsymbol{h_i^2}) + \sum_{\boldsymbol{x_j} \in \mathcal{N}_i^{\mathcal{D}}} \sum_v f(\boldsymbol{h_i^v}, \boldsymbol{h_j^v})}] \tag{3.7}$$

where $\boldsymbol{x_j^v}$ is the v^{th} view of $\boldsymbol{x_j}$, $\boldsymbol{h_j^v}$ is the hidden representation of $\boldsymbol{x_j^v}$ after non-linear mappings, $f(\boldsymbol{h_i^1}, \boldsymbol{h_i^2})$ is a similarity measurement function, e.g., $f(a, b) = \exp(\frac{a \cdot b}{\tau})$, τ is the temperature, and $\mathcal{N}_i^{\mathcal{D}} = \mathcal{D} \backslash \{i\}$. However, L_1 suffers from the class collision problem [43], where minimizing L_1 pushes two samples from the same cluster away from each other and thus leads to sub-optimal performance. To alleviate these potential concerns, we propose a novel weighting strategy as follows.

$$L_{ctr} = -\mathbb{E}_{x_i \in \mathcal{D}}[\log \frac{f(\boldsymbol{h_i^1}, \boldsymbol{h_i^2})}{f(\boldsymbol{h_i^1}, \boldsymbol{h_i^2}) + \sum_{\boldsymbol{x_j} \in \mathcal{N}_i^{\mathcal{D}}} \sum_v sim(\boldsymbol{q_i}, \boldsymbol{q_j}) f(\boldsymbol{h_i^v}, \boldsymbol{h_j^v})}] \tag{3.8}$$

where $\boldsymbol{q_i} = [q_i^1; ...; q_i^v]$ is the concatenation of the v views soft membership for the i^{th} samples and $sim(\boldsymbol{q_i}, \boldsymbol{q_j}) = \exp(1 - \frac{\boldsymbol{q_i} \cdot \boldsymbol{q_j}}{|\boldsymbol{q_i}||\boldsymbol{q_j}|})$. The intuition of the weighting function $sim(\boldsymbol{q_i}, \boldsymbol{q_j})$ is that if two samples have the similar probabilities of being assigned to the same cluster, then this pair of samples should be considered as a positive pair, and we need to reduce the weight of this pair of samples in the denominator in Equation 3.8 in order to address the class collision issue. Notice that if $\boldsymbol{q_i}$ and $\boldsymbol{q_j}$ are equal in the extreme case, then the value of the weighting function is 1. The more dissimilar $\boldsymbol{q_i}$ and $\boldsymbol{q_j}$ are, the large the value of the weighting function $sim(\boldsymbol{q_i}, \boldsymbol{q_j})$ is. Eq. 3.7 assigns the equal weight to all negative samples, which inevitably pushes two samples from the same cluster away from each other, while in our proposed weighted contrastive loss L_{ctr}, we utilize the pseudo-label to alleviate such an issue.

One drawback of the contrastive learning based regularization is the high computational cost as well as the high memory requirement to compute and store the similarity matrix for any pairs of two samples [8]. To address this issue, a non-contrastive learning based method [8] has been proposed:

$$L_2 = -\mathbb{E}_{x_i \in \mathcal{D}}(\frac{\boldsymbol{h_i^1}}{|\boldsymbol{h_i^1}|_2} \cdot SG(\frac{\boldsymbol{z_i^2}}{|\boldsymbol{z_i^2}|_2}) + \frac{\boldsymbol{h_i^2}}{|\boldsymbol{h_i^2}|_2} \cdot SG(\frac{\boldsymbol{z_i^1}}{|\boldsymbol{z_i^1}|_2})) \tag{3.9}$$

where SG denotes stop gradient operation, and $\boldsymbol{H_i^v} = g(\boldsymbol{Z_i^v}) \in \mathbb{R}^{n \times d}$. Notice that different from contrastive regularization L_{ctr}, $g(\cdot)$ is shared by two views in L_2. Intuitively, L_2 aims to maximize the similarity of the hidden representations of two views. However, in practice, if parts of the original features are missing or noisy, L_2 might also result in sub-optimal solution, which is examined in the case study in Section 4.4. Inspired by [24], we propose a cross attention module to borrow the information from the other view to alleviate this issue:

$$\begin{aligned} \boldsymbol{C}_{1,2} &= \boldsymbol{H}^1 \boldsymbol{W}_{1,2} (\boldsymbol{H}^2)^T \\ \boldsymbol{O}^1 &= \tanh(\boldsymbol{Z}^1 \boldsymbol{W}_1 + \boldsymbol{Z}^2 \boldsymbol{W}_2 \boldsymbol{C}_{1,2}) \\ \boldsymbol{O}^2 &= \tanh(\boldsymbol{Z}^2 \boldsymbol{W}_2 + \boldsymbol{Z}^1 \boldsymbol{W}_1 \boldsymbol{C}_{1,2}^T) \\ \boldsymbol{A}^v &= \text{softmax}(\boldsymbol{O}^v) \\ \boldsymbol{T}^v &= \boldsymbol{H}^v \odot \boldsymbol{A}^v \end{aligned} \tag{3.10}$$

where $\boldsymbol{H^v} = g(\boldsymbol{Z^v}) \in \mathbb{R}^{n \times d}$ denotes the hidden representation after the mapping function $g(\cdot)$, $\boldsymbol{W}_{1,2} \in \mathbb{R}^{n \times n}$, $\boldsymbol{W}_1 \in \mathbb{R}^{d \times d}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{d \times d}$ are the weight matrices and $\boldsymbol{C}_{1,2} \in \mathbb{R}^{d \times d}$ aims to capture the relatedness of

features across two views. By leveraging the consensus information to measure the importance of each feature, $\boldsymbol{O}^1 \in \mathbb{R}^{n\times d}$ and $\boldsymbol{O}^2 \in \mathbb{R}^{n\times d}$ encode the information from both views in order to alleviate the issue of missing or noisy features. $\boldsymbol{A}^v \in \mathbb{R}^{n\times d}$ is the attention matrix for the v^{th} view, $\boldsymbol{T}^v \in \mathbb{R}^{n\times d}$ is the output of the cross attention module for the v^{th} view and $\odot$ denotes the element-wise multiplication operation. The main difference between $\boldsymbol{H}^v$ and $\boldsymbol{T}^v$ is that $\boldsymbol{A}^v$ first encodes the information from both views and then adjust the importance of the features in $\boldsymbol{H}^v$ based on the consensus information from both views to mitigate the issue of the missing or noisy features. Similar to L_2, the non-contrastive learning loss could be updated as follows.

$$L_{nctr} = -\mathbb{E}_{X_i\in\mathcal{D}}\left(\frac{\boldsymbol{t}_i^1}{|\boldsymbol{t}_i^1|_2}\cdot SG\left(\frac{\boldsymbol{z}_i^2}{|\boldsymbol{z}_i^2|_2}\right) + \frac{\boldsymbol{t}_i^2}{|\boldsymbol{t}_i^2|_2}\cdot SG\left(\frac{\boldsymbol{z}_i^1}{|\boldsymbol{z}_i^1|_2}\right)\right) \tag{3.11}$$

3.4 Objective Function and Proposed Algorithm Now, we are ready to introduce the overall objective function:

$$\min J = L_{KL} + \gamma L_d + \alpha L_F + \beta L_{reg} \tag{3.12}$$

where L_{KL} is KL-divergence maximizing the mutual agreement of soft assignment of two views, L_d ensures that the samples belonging to the same cluster will get closer, L_F is the group fairness constraint, L_{reg} is either L_{ctr} or L_{nctr} regularizing the latent representations, and α, β, and γ are positive hyper-parameters balancing these terms. Notice that α is the same parameter as in Equation 3.4. The proposed method could be solved in Expectation-Maximization (EM) steps. Our algorithm is presented in Appendix A.1. Specifically, we take the results of K-means as the initial centroids of k clusters in the first step. Next, we first compute the soft assignment based on Equation 3.4, and maximize the mutual agreement of soft membership of multiple views based on Equation 3.12 in Step 2 and Step 3; then we update the centroid of each cluster based on Equation 3.2 in step 4. These steps are repeated T times, where T is the number of the iterations. Finally, we compute the soft assignment based on Equation 3.1 by excluding the sensitive features at the test phase.

4 Experiments

In this section, we demonstrate the performance of our proposed framework in terms of effectiveness by comparing it with state-of-the-art methods.

4.1 Experimental Setup We mainly evaluate our proposed algorithm on three data sets with fairness constraints, including Credit Card data set [36], Bank Marketing data set [36] and Zafar data set [36], and two data sets without fairness constraints, including Noisy MNIST [2] and X-ray Microbeam (XRMB) [31]. Specifically, the sensitive feature on the Credit Card data set is gender, The sensitive feature on Bank Marketing data set is marital status. Zafar data set [36] is a widely-used synthetic data set, where one binary value is generated as the sensitive feature. More details of these data sets could be found in Appendix A.2.

Baselines: In the experiment, L_{ctr} is the regularization term (i.e., L_{reg}) in FAIR-MVC-C and L_{nctr} is the regularization term (i.e., L_{reg}) in FAIR-MVC-N. We compare the performance of our methods with the following baselines: (1). K-means: a method aiming to partition samples into several clusters where each sample is assigned to the nearest cluster; (2). DEC [32]: a deep embedded clustering method by learning feature representations and cluster assignments with deep neural networks; (3). Contrastive-Clustering (CC) [20]: a contrastive learning based clustering method, which optimizes the instance-level and cluster-level contrastive loss simultaneously; (4). MvDSCN [45]: a multi-view deep subspace clustering network aiming to learn a multi-view self-representation matrix; (5). LF-IMVC [23]: an incomplete multi-view clustering method (as this method is designed for missing feature scenario, we only report its performance in table 2). To investigate the contributions of different parts of FAIR-MVC-N and FAIR-MVC-C, we conduct ablation study by introducing four variations of FAIR-MVC, including FAIR-MVC-NF that removes fairness constraint from FAIR-MVC-N, FAIR-MVC-CF that removes fairness constraint from FAIR-MVC-C, FAIR-MVC-1 where L_{reg} is replaced by L_1, and FAIR-MVC-2 where L_{reg} is replaced by L_2.

Evaluation: We present the results regarding the following metrics: (1) NMI [9]: normalized mutual information, which measures the mutual dependency of two variables. (2) Balance: a group fairness measurement, which is formulated as follows:

$$Balance = \min_i \frac{\min_a |C_i \cup r_j|}{|C_i|} \tag{4.13}$$

where $C_i \in \{0, 1\}$ denotes the i-th cluster and r_j denotes the j-th protected subgroup. Typically, the upper bound of balanced is determined by the distribution of the sensitive feature, and a higher value of balance indicates a fairer result.

4.2 Experimental results In this subsection, we demonstrate the effectiveness of the proposed method. Table 1 shows the performance of state-of-the-art methods and our proposed methods. By observations, we find that (1) most baselines fail to provide fair results, though many of them achieve outstanding performance;

Table 1: Results on three data sets with sensitive features. (Higher balance score indicates better fairness.)

-	Credit Card		Zafar		Bank Marketing	
Model	**NMI**	**Balance**	**NMI**	**Balance**	**NMI**	**Balance**
K-means	0.2094 ± 0.0114	0.3553 ± 0.0037	0.7032 ± 0.0078	0.1706 ± 0.0076	0.2867 ± 0.0144	0.3765 ± 0.0066
DEC	0.2103 ± 0.0209	0.3596 ± 0.0060	0.7255 ± 0.0192	0.1685 ± 0.0073	0.3093 ± 0.0115	0.3760 ± 0.0096
MvDSCN	0.2192 ± 0.0153	0.3582 ± 0.0041	0.7691 ± 0.0042	0.1713 ± 0.0065	0.3624 ± 0.0055	0.3759 ± 0.0067
CC	0.2387 ± 0.0128	0.3574 ± 0.0047	0.7895 ± 0.0068	0.1701 ± 0.0071	0.3623 ± 0.0101	0.3746 ± 0.0097
Fair-MVC-1	0.2423 ± 0.0070	0.3743 ± 0.0028	0.7878 ± 0.0101	0.2735 ± 0.0077	0.3587 ± 0.0050	0.4116 ± 0.0073
Fair-MVC-2	0.2434 ± 0.0039	0.3710 ± 0.0036	0.7996 ± 0.0110	0.2767 ± 0.0055	0.3632 ± 0.0069	0.4106 ± 0.0092
Fair-MVC-CF	0.2386 ± 0.0100	0.3599 ± 0.0028	0.7994 ± 0.0107	0.1770 ± 0.0044	0.3861 ± 0.0103	0.3735 ± 0.0070
Fair-MVC-NF	**0.2484 ± 0.0095**	0.3618 ± 0.0034	**0.8161 ± 0.0157**	0.1768 ± 0.0053	**0.3899 ± 0.0091**	0.3776 ± 0.0091
Fair-MVC-C	0.2459 ± 0.0078	**0.3783 ± 0.0032**	0.7974 ± 0.0051	**0.2896 ± 0.0059**	0.3816 ± 0.0116	0.4208 ± 0.0059
Fair-MVC-N	0.2471 ± 0.0041	0.3743 ± 0.0029	0.8119 ± 0.0150	0.2827 ± 0.0074	0.3839 ± 0.0109	**0.4240 ± 0.0075**

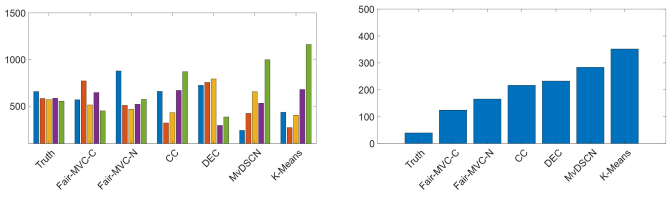
for instance, though CC achieves the competitive performance on the Credit Card data set, Zafar data set, and Bank Marketing data set, its balance score is much lower than that of Fair-MVC-C and Fair-MVC-N; (2) Fair-MVC-NF outperforms all baselines on the Credit Card data set, Zafar data set and Bank Marketing data set in Table 1 without considering the fairness; (3) comparing with state-of-the-art methods, Fair-MVC-C and Fair-MVC-N achieve much better balance score by taking fairness into considerations. The experimental results on non-fairness data sets (i.e., Noisy MNIST and XRMB) could be found in Appendix A.3.

Ablation study To demonstrate the effectiveness of each component in our proposed framework, we conduct an ablation study. In Table 1, comparing the performance of Fair-MVC-C with Fair-MVC-CF, the balance score of Fair-MVC-C increases by more than 63% on Zafar data set while the NMI of Fair-MVC-C only decreases by less than 0.25% on Zafar data set, which demonstrates the effectiveness of Fair-MVC-C. By comparing the performance of Fair-MVC-2 and Fair-MVC-N, we demonstrate that our proposed novel non-contrastive regularizer can improve the performance by leveraging information from the complementary view to some extent in the presence of the missing feature scenario. What's more, Fair-MVC-C outperforms CC and Fair-MVC-1 on Credit Card data set, Zafar data set and Bank Marketing data set. As we mentioned early, the main drawback of vanilla contrastive regularizer (i.e., L_1 in Equation 3.7) is that minimizing the loss function pushes two samples from the same cluster away from each other, resulting in the class collision problem. Fair-MVC-C and Fair-MVC-CF consider the soft membership by reducing the weights for any pairs of samples possibly from the same cluster in the denominator of Equation 3.8.

4.3 Fairness Analysis

Why do we care about the fairness of the clustering results? To answer this question, let us first look at the clustering results on the Credit Card data set. On the Credit Card data set, the attributes consist of the historical payments; the sensitive feature is gender; it consists of five clusters. The first cluster means that the customers pay their debt duly and the rest four clusters mean that the customers fail to pay the debt in one, two, three, or more than three consecutive months. If the banks aim to determine whether to lower the interest rate of the customers based on their payment records, they want the decisions made upon the clustering results to be fair, and to not discriminate against any protected group. Thus, reducing the potential bias is crucial for the clustering methods. In Figure 2, we visualize the fairness measurement in terms of the count of males in each cluster on the Credit Card data set, as the sensitive feature in this data set is gender. In Figure 2, five bars (in the top figure) in each group (algorithm) denote the number of males (sensitive feature) for five clusters and the bar in the bottom figure means the standard deviation of the number of males for each method. Intuitively, the distribution of males for a fair clustering result should be identical to the distribution of males using the ground truth. The more dissimilar to the ground truth, the more unfair the clustering results. By observation, we find that Fair-MVC-C is mostly identical to the ground truth, compared with other baselines in terms of the count distribution and

Figure 2: Fairness analysis on the Credit Card data set. **Top**: Five bars in each group (algorithm) denote the number of males (sensitive feature) in five clusters. The more discrepancy to ground truth, the worse fairness. **Bottom**: each bar means the standard deviation of the number of males for each method. The more similar to the ground truth, the fairer the clustering results.



the standard deviation of the count of males. These baselines fail to consider the fairness constraint, thus leading to lower balance score.

4.4 Case studies: Missing Features. In this subsection, we first demonstrate the effectiveness of the two regularizers in our proposed methods (i.e., Fair-MVC-N and Fair-MVC-C) and state-of-the-art methods in the presence of missing features. To control the percentage of missing features p, we randomly mask the features with Bernoulli distribution (where p is the possibility of being masked) on the Credit Card data set. Table 2 shows the performance of these methods and we gradually increase the ratio of missing features from 0 to 25% and then to 50%. Notice that the upper bound of the balance score is determined by the distribution of sensitive features, which is 0.4092 for the Credit Card data set. Based on the results from Table 2, we have the following observations: (1) when there are no missing features (i.e., $p = 0\%$), Fair-MVC-N outperforms Fair-MVC-C on Credit card data set; (2) when we gradually increase the percentage of missing features, Fair-MVC-C gradually outperform Fair-MVC-N; (3) the performance of the most baseline methods decreases dramatically as the percentage of missing feature increases. As for the second observation, we conjecture that the contrastive regularizer maximizes the similarity between two views from the same instance, and meanwhile, it contrasts the difference between two views from two different instances. This contrasting operation leverages the information from other instances to infer the missing features, thus enhancing the quality of hidden representation. Different from the contrastive regularizer, the non-contrastive regularizer only aims to maximize the similarity between two views from the same instance, and thus it fails to infer extra information from other instances. Therefore, the performance of Fair-MVC-N is a little bit worse than Fair-MVC-C.

Noisy Features. Next, we further investigate the effectiveness of the two regularizers (i.e., non-contrastive regularizer and contrastive regularizer) in the presence of noisy features. Table 3 shows the performance of state-of-the-art methods. Here is the procedure to perturb the raw data. We first use Bernoulli distribution to select p percent of data and then inject the white noise (e.g., $\mathcal{N}(0, 1)$) into the raw data on the Credit Card data set. By observations, we find that when p percent of white noise is added to raw data, the performance of most methods begins to decrease. Different from most baseline methods, the performance of Fair-MVC-C decreases slightly, when 50% percent of noise is added. We conjecture that our proposed contrastive regularizer is more robust to the noisy feature as it can contrast one instance's representation with others' representations. However, vanilla contrastive-based method (i.e., CC) suffer from the class collision issue and two non-contrastive regularizers fail to leverage the information from other instances.

4.5 Discussion Combining the experimental results from Table 1, Table 2 and Table 3, we observe that for the clean input data, the non-contrastive regularizer tends to have better performance than the contrastive regularizer for clustering, as the contrastive regularizer usually suffers from the class collision issue. Nevertheless, in the presence of missing or noisy features, we observe that the performance of the non-contrastive regularizer decreases by a large margin, while the performance of the contrastive regularizer decreases slowly by contrasting with other instances and inferring extra information form other instances. On the other hand, based on the experimental results in the efficiency analysis in Appendix A.5, we observe that although the contrastive regularizer outperforms the non-contrastive regularizer, the time complexity of the contrastive regularizer is quadratic with respect to the number of instances, whereas the time complexity of non-contrastive regularizer is linear.

5 Conclusion

In this paper, we propose Fair-MVC, a deep fairness-aware multi-view clustering method. Fair-MVC maximizes the mutual agreement of the soft membership assignment based on each view; in the meanwhile, it enforces the fairness constraint by requiring that the fraction of different groups in each cluster be approximately the same as the fraction in the whole data set. In addition, we adopt the idea of contrastive learning and non-contrastive learning, and propose novel regularizers to handle heterogeneous data in complex scenarios with missing or noisy features. The experimental results on both synthetic and real-world data sets demonstrate the effectiveness of the proposed framework. We also provide insights regarding the relative performance of contrastive and non-contrastive regularizers in different scenarios.

Acknowledgment

This work is supported by National Science Foundation under Award No. IIS-1947203, IIS-2117902, IIS-2137468, IIS-19-56151, IIS-17-41317, and IIS 17-04532, the C3.ai Digital Transformation Institute, MIT-IBM Watson AI Lab, and IBM-Illinois Discovery Accelerator Institute - a new model of an academic-industry partnership designed to increase access to technology education and skill development to spur breakthroughs in

Table 2: **Case Study: Missing Feature Scenario.** Results on Credit Card data set, where p denotes the percentage of missing features. (Higher balance score indicates better fairness.)

-	$p = 0\%$		$p = 25\%$		$p = 50\%$	
Model	**NMI**	**Balance**	**NMI**	**Balance**	**NMI**	**Balance**
K-means	0.2094 ± 0.0114	0.3553 ± 0.0037	0.1567 ± 0.0148	0.3602 ± 0.0060	0.1356 ± 0.0063	0.3632 ± 0.0038
DEC	0.2103 ± 0.0209	0.3596 ± 0.0060	0.2005 ± 0.0079	0.3640 ± 0.0078	0.1567 ± 0.0121	0.3626 ± 0.0040
MvDSCN	0.2192 ± 0.0153	0.3582 ± 0.0041	0.2034 ± 0.0159	0.3594 ± 0.0048	0.1634 ± 0.0183	0.3663 ± 0.0069
CC	0.2387 ± 0.0128	0.3574 ± 0.0047	0.2095 ± 0.0094	0.3616 ± 0.0047	0.1762 ± 0.0175	0.3680 ± 0.0049
LF-IMVC	0.2228 ± 0.0093	0.3625 ± 0.0043	0.2039 ± 0.0083	0.3581 ± 0.0040	0.1723 ± 0.0144	0.3621 ± 0.0031
FAIR-MVC-C	0.2459 ± 0.0078	**0.3783 ± 0.0032**	0.2165 ± 0.0054	**0.3871 ± 0.0046**	**0.1871 ± 0.0079**	**0.3907 ± 0.0076**
FAIR-MVC-N	**0.2471 ± 0.0041**	0.3743 ± 0.0029	**0.2209 ± 0.0075**	0.3820 ± 0.0121	0.1803 ± 0.0060	0.3854 ± 0.0114

Table 3: **Case Study: Noisy Feature Scenario.** Results on Credit Card data set, where p denotes the percentage of perturbed features (Higher balance score indicates better fairness.)

-	$p = 0\%$		$p = 25\%$		$p = 50\%$	
Model	**NMI**	**Balance**	**NMI**	**Balance**	**NMI**	**Balance**
K-means	0.2094 ± 0.0114	0.3553 ± 0.0037	0.1793 ± 0.0078	0.3589 ± 0.0080	0.1663 ± 0.0098	0.3561 ± 0.0072
DEC	0.2103 ± 0.0209	0.3596 ± 0.0060	0.1923 ± 0.0087	0.3561 ± 0.0072	0.1779 ± 0.0100	0.3617 ± 0.0030
MvDSCN	0.2192 ± 0.0153	0.3582 ± 0.0041	0.2011 ± 0.0070	0.3612 ± 0.0051	0.1885 ± 0.0051	0.3626 ± 0.0039
CC	0.2387 ± 0.0128	0.3574 ± 0.0047	0.2078 ± 0.0078	0.3609 ± 0.0087	0.1960 ± 0.0083	0.3583 ± 0.0040
FAIR-MVC-C	0.2459 ± 0.0078	**0.3783 ± 0.0032**	**0.2490 ± 0.0147**	**0.3785 ± 0.0041**	**0.2397 ± 0.0052**	0.3819 ± 0.0067
FAIR-MVC-N	**0.2471 ± 0.0041**	0.3743 ± 0.0029	0.2394 ± 0.0051	0.3741 ± 0.0047	0.2256 ± 0.0068	**0.3832 ± 0.0054**

emerging areas of technology. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

References

[1] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. M. Wallach. A reductions approach to fair classification. In *Proceedings of the 35th ICML 2018*, volume 80, pages 60–69. PMLR, 2018.

[2] S. Basu, M. Karki, S. Ganguly, R. DiBiano, S. Mukhopadhyay, S. Gayaka, R. Kannan, and R. R. Nemani. Learning sparse feature representations using probabilistic quadtrees and deep belief nets. *Neural Process. Lett.*, 45(3):855–867, 2017.

[3] M. B. Blaschko and C. H. Lampert. Correlational spectral clustering. In *2008 IEEE (CVPR 2008)*. IEEE Computer Society, 2008.

[4] U. Brefeld and T. Scheffer. Co-em support vector learning. In *Machine Learning, Proceedings of (ICML)*, volume 69. ACM, 2004.

[5] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th Annual ICML 2009*, volume 382, pages 129–136. ACM, 2009.

[6] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th ICML*, volume 119, pages 1597–1607. PMLR, 2020.

[7] X. Chen, B. Fain, L. Lyu, and K. Munagala. Proportionally fair clustering. In *Proceedings of the 36th ICML 2019*, volume 97, pages 1032–1041. PMLR, 2019.

[8] X. Chen and K. He. Exploring simple siamese representation learning. In *IEEE CVPR 2021*, pages 15750–15758. Computer Vision Foundation, 2021.

[9] T. M. Cover and J. A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[10] S. Das, R. K. Behera, S. K. Rath, et al. Real-time sentiment analysis of twitter streaming data for stock prediction. *Procedia computer science*, 132:956–964, 2018.

[11] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science 2012*, pages 214–226. ACM, 2012.

[12] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD*, pages 259–268. ACM, 2015.

[13] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Advances in NeurIPS*, pages 3315–3323, 2016.

[14] S. Huang, Z. Kang, and Z. Xu. Auto-weighted multi-view clustering via deep matrix decomposition. *Pattern Recognit.*, 97, 2020.

[15] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. In *Advances in NeurIPS*, 2020.

[16] M. Kleindessner, S. Samadi, P. Awasthi, and J. Morgenstern. Guarantees for spectral clustering with fairness constraints. In *Proceedings of the 36th ICML 2019*, volume 97, pages 3458–3467. PMLR, 2019.

[17] C. Lee and I. Paik. Stock market analysis from twitter and news based on streaming big data infrastructure. In *2017 IEEE 8th (iCAST)*, pages 312–317. IEEE, 2017.

[18] P. Li, H. Zhao, and H. Liu. Deep fair clustering for visual learning. In *2020 IEEE/CVF CVPR 2020*, pages 9067–9076. Computer Vision Foundation / IEEE, 2020.

[19] S. Li, Y. Jiang, and Z. Zhou. Partial multi-view clustering. In *Proceedings of the Twenty-Eighth AAAI*, pages 1968–1974. AAAI Press, 2014.

[20] Y. Li, P. Hu, J. Z. Liu, D. Peng, J. T. Zhou, and X. Peng. Contrastive clustering. In *Thirty-Fifth AAAI 2021*, pages 8547–8555. AAAI Press, 2021.

[21] Z. Li, Q. Wang, Z. Tao, Q. Gao, and Z. Yang. Deep adversarial multi-view clustering network. In *Proceedings of IJCAI*, pages 2952–2958. ijcai.org, 2019.

[22] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng. COMPLETER: incomplete multi-view clustering via contrastive prediction. In *IEEE CVPR 2021*, pages 11174–11183. Computer Vision Foundation, 2021.

[23] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao. Late fusion incomplete multi-view clustering. *IEEE TPAMI*, 41(10):2410–2423, 2018.

[24] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances in NeurIPS*, pages 289–297, 2016.

[25] N. E. Morden, C. H. Colla, T. D. Sequist, and M. B. Rosenthal. Choosing wisely—the politics and economics of labeling low-value services. *The New England journal of medicine*, 370(7):589, 2014.

[26] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the 2000 ACM CIKM*, pages 86–93. ACM, 2000.

[27] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou. COMIC: multi-view clustering without parameter selection. In *Proceedings of ICML*, volume 97, pages 5092–5101. PMLR, 2019.

[28] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. In *Computer Vision - ECCV*, volume 12356, pages 776–794. Springer, 2020.

[29] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[30] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes. On deep multi-view representation learning. In *Proceedings of ICML*, volume 37, pages 1083–1092. JMLR.org, 2015.

[31] J. Westbury. X-ray microbeam speech production database user's handbook: Madison. *WI: Waisman Center, University of Wisconsin*, 1994.

[32] J. Xie, R. B. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33nd ICML 2016*, volume 48, pages 478–487. JMLR.org, 2016.

[33] Y. Xie, B. Lin, Y. Qu, C. Li, W. Zhang, L. Ma, Y. Wen, and D. Tao. Joint deep multi-view learning for image clustering. *IEEE TKDE*, 33(11):3594–3606, 2021.

[34] C. Xu, D. Tao, and C. Xu. A survey on multi-view learning. *CoRR*, abs/1304.5634, 2013.

[35] I.-C. Yeh and C.-h. Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications*, 36(2):2473–2480, 2009.

[36] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi. Fairness constraints: Mechanisms for fair classification. In *Proceedings of the 20th AISTATS 2017*, volume 54, pages 962–970. PMLR, 2017.

[37] R. S. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *Proceedings of the 30th ICML*, volume 28, pages 325–333, 2013.

[38] H. Zhao, Z. Ding, and Y. Fu. Multi-view clustering via deep matrix factorization. In *Proceedings of the Thirty-First AAAI*, pages 2921–2927. AAAI Press, 2017.

[39] L. Zheng, Y. Cheng, and J. He. Deep multimodality model for multi-task multi-view learning. In T. Y. Berger-Wolf and N. V. Chawla, editors, *Proceedings of the SDM 2019*, pages 10–18. SIAM, 2019.

[40] L. Zheng, Y. Cheng, H. Yang, N. Cao, and J. He. Deep co-attention network for multi-view subspace learning. In *WWW '21: The Web Conference 2021*, pages 1528–1539. ACM / IW3C2, 2021.

[41] L. Zheng, D. Fu, R. Maciejewski, and J. He. Deepergxx: deepening arbitrary gnns. *arXiv preprint arXiv:2110.13798*, 10, 2022.

[42] L. Zheng, J. Xiong, Y. Zhu, and J. He. Contrastive learning with complex heterogeneity. In *KDD '22: The 28th ACM SIGKDD 2022*, pages 2594–2604. ACM, 2022.

[43] M. Zheng, F. Wang, S. You, C. Qian, C. Zhang, X. Wang, and C. Xu. Weakly supervised contrastive learning. *CoRR*, abs/2110.04770, 2021.

[44] D. Zhou, L. Zheng, Y. Zhu, J. Li, and J. He. Domain adaptive multi-modality neural attention network for financial forecasting. In *WWW '20: The Web Conference 2020*, pages 2230–2240. ACM / IW3C2, 2020.

[45] P. Zhu, B. Hui, C. Zhang, D. Du, L. Wen, and Q. Hu. Multi-view deep subspace clustering networks. *CoRR*, abs/1908.01978, 2019.

A Appendix

A.1 Algorithm The proposed method could be solved in Expectation-Maximization (EM) steps. Our algorithm is presented as follows.

Algorithm 1 FAIR-MVC Algorithm

Input: The total number of iterations T, the input data $\boldsymbol{X}^1, \boldsymbol{X}^2, ..., \boldsymbol{X}^v$, the sensitive features $\boldsymbol{R}$ and the number of cluster k.

Output: The membership matrix $\boldsymbol{Q}$.

Step 1: Take the output of K-means as the initial centroids of $\boldsymbol{k}$ clusters or randomly initialize the centroids.

for $t = 1$ to T **do**

Step 2: Compute the soft assignment based on Equation 3.4.

Step 3: Minimize the overall objective function based on Equation 3.12.

Step 4 : Update the centroids based on Equation 3.2.

end for

Step 5: Compute the membership by averaging the soft assignment of different views based on Eq. 3.1.

A.2 Experimental Setup We mainly evaluate our proposed algorithm on three data sets with fairness constraints, including Credit card clients data set [*], Bank Marketing Data set [†], Zafar data set [‡], and two data sets without fairness constraints, which are Noisy MNIST data set [§], and X-ray Microbeam (XRMB) [¶]. Specifically, the Credit card clients data set describes the customers' default payments in Taiwan and this data set consists of 30,000 samples with 24 attributes. The sensitive feature in this data set is gender. Bank Marketing Data set is associated with direct marketing campaigns of a Portuguese banking institution and it aims to see if the product (bank term deposit) would be ('yes') or not ('no') subscribed. The sensitive feature in this data set is marital status. This data set consists of 1,000 instances and 20 attributes. Zafar data set [36] is a widely-used synthetic data set, where one binary value is generated as the sensitive feature. For Credit Card, and Bank Marketing data set, we use two non-linear functions (e.g., *Sigmoid* and *Relu*) to generate two views. Noisy MNIST data set originally consists of 70,000 images of handwritten digits and we follow [30] by adding white Gaussian noise to each pixel to generate the first view, and randomly rotating a figure with an angle from $[-\frac{\pi}{4}, \frac{\pi}{4}]$ to generate the second view. XRMB [31] is a multi-view multi-class data set, which consists of 40 binary labels and two views. The first view is acoustic data with 273 features and the second view is articulatory data with 112 features. As some state-of-the-art methods are very slow, we reduce the number of instances for some large data sets to ensure that we can include the results of most baselines. We randomly sample 5,000 instances from the Noisy MNIST data set and 3,000 instances from the XRMB data set from 6 classes.

Table 4: Results on real-world data sets without sensitive features

-	**NMI**	**NMI**
Model	XRMB	Noisy MNIST
K-means	0.1692 ± 0.0049	0.3882 ± 0.0117
DEC	0.2012 ± 0.0086	0.4899 ± 0.0227
CC	0.2107 ± 0.0100	0.4902 ± 0.0101
MvDSCN	0.2056 ± 0.0078	0.4770 ± 0.0128
FAIR-MVC-C	0.2163 ± 0.0101	**0.4997 ± 0.164**
FAIR-MVC-N	**0.2214 ± 0.0071**	0.4686 ± 0.182

Configuration: In all experiments, we set the learning rate to be 0.001 and the weight decay rate to be 0.0005. The optimizer is momentum SGD. The neural network structure for $f^v(\cdot)$ of the proposed methods is an two-layer Multi-layer Perceptron (MLP) and The neural network structure for $g^v(\cdot)$ of the proposed methods is an one-layer MLP. The experiments are repeated 5 times if not specified. The code of our algorithms could be found in the link [‖]. The experiments are performed on a Windows machine with a 16GB RTX 5000 GPU and 64GB memory.

A.3 Experimental results on real-world data sets without sensitive features In this subsection, we evaluate the performance of our proposed method on two datasets without sensitive features, including Noisy MNIST and XRMB. Specifically, we randomly sample 5,000 instances from the Noisy MNIST data set and 3,000 instances from the XRMB data set from 6 classes. Table 4 shows the performance of state-of-the-art methods and our proposed methods. By observations, we find that our proposed method FAIR-MVC-N and FAIR-MVC-C outperform all state-of-the-art methods on the XRMB data set and Noisy MNIST data set in Table 4.

A.4 Parameter Analysis In the subsection, we conduct the parameter analysis regarding α, β, and γ for FAIR-MVC-C. Specifically, we change the value of one hyper-parameter, fix the other hyper-parameters, and report the results. Figure 3 shows the results regarding these three hyper-parameters. Figure 3 (a) and Figure 3 (b) show the NMI and balance score when we change the value of α. We observe that when $\alpha = 10$, FAIR-MVC-C achieves the high-

[*] https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[†] https://ashryaagr.github.io/Fairness.jl/dev/datasets/#Bank-Marketing-Dataset

[‡] https://ashryaagr.github.io/Fairness.jl/dev/datasets/#Fairness.genZafarData

[§] http://yann.lecun.com/exdb/mnist/

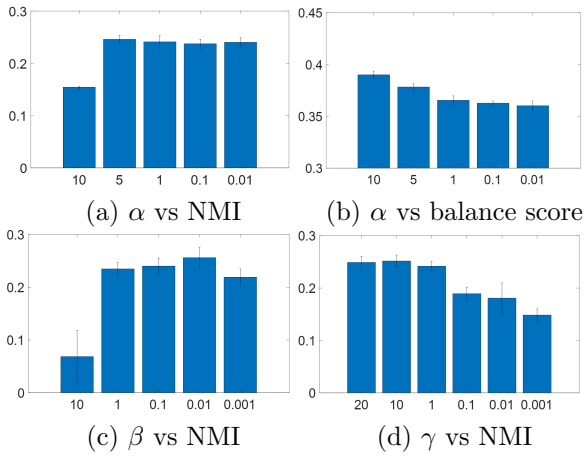
[¶] https://ttic.uchicago.edu/~klivescu/XRMB_data/full/README

[‖] https://drive.google.com/file/d/1TOsxjOfnNe7JUBYoKyjPydoKIqHFWcd9/view

est balance score but its performance is the worst as the algorithm mainly focuses on minimizing the fairness loss. When we reduce the value of α, then the performance increases to 24.5% at $\alpha = 5$ and starts to change slightly from $\alpha = 5$ to $\alpha = 0.01$. However, Figure 3 (b) shows that the results gradually become unfair (with a lower balance score) if we decrease the value of α from 10 to 0.01. Based on these observations, we may conclude that there is a trade-off between balance score and NMI, and Fair-MVC-C tends to have a higher NMI and a lower balance score with a lower α and vice versa. Figure 3 (c) shows the performance of Fair-MVC-C by changing the value of β. We observe that the algorithm achieves the best performance when $\beta = 0.01$ and it tends to have a large standard deviation when β is large (e.g., $\beta = 10$). In the overall objective function (e.g., Equation 3.12), β is the weight of contrastive regularizer. A large number of β greatly reduces the importance of other components (e.g., the centrality of the clustering) and thus it leads to the unstable performance of clustering results. Figure 3 (d) shows the performance of Fair-MVC-C with different value of γ. We observe that the algorithm achieves the best performance when γ is around 10. In the overall objective function (e.g., Equation 3.12), γ controls the importance of centrality and a large value of γ implies that the instances assigned to the same cluster will be closer in the hidden space. Thus, in Figure 3 (d), Fair-MVC-C with a large value of γ usually tends to have a better performance.

Next, we conduct the parameter analysis regarding α, β, and γ for Fair-MVC-N. Specifically, we change the value of one hyper-parameter, fix the rest hyper-parameters, and report the performance. Figure 4 shows the results regarding these three hyper-parameters. Figure 4 (a) and Figure 4 (b) show the NMI and balance score when we change the value of α. We observe that when $\alpha = 10$, Fair-MVC-N achieves the best balance score; when we decrease α, then the performance increases but the results become unfair (with a lower balance score). Based on the results from Figure 4 (a) and Figure 4 (b), we can also conclude that there is a trade-off between balance score and NMI, and Fair-MVC-N has a higher NMI with a lower balance score with smaller α. Figure 4 (c) shows the performance of Fair-MVC-N by changing the value of β. We observe that the algorithm achieves the best performance when $\beta = 0.01$ and it tends to have a large standard deviation when β is large. In the overall objective function (e.g., Equation 3.12), β is the weight of contrastive regularizer. A large number of β greatly reduce the importance of other components (e.g., the centrality of the clustering), and thus it leads to the unstable performance of clustering results. Figure 4 (d) shows the performance of Fair-MVC-N with different value of γ. We observe that the algorithm achieves the best performance when γ is 20. In the overall objective function (e.g., Equation 3.12), γ controls the importance of centrality and a large value of γ implies that the instances assigned to the same cluster will be closer in the hidden space. Thus, in Figure 4 (d), Fair-MVC-N with a large value of γ usually tends to have a better performance.

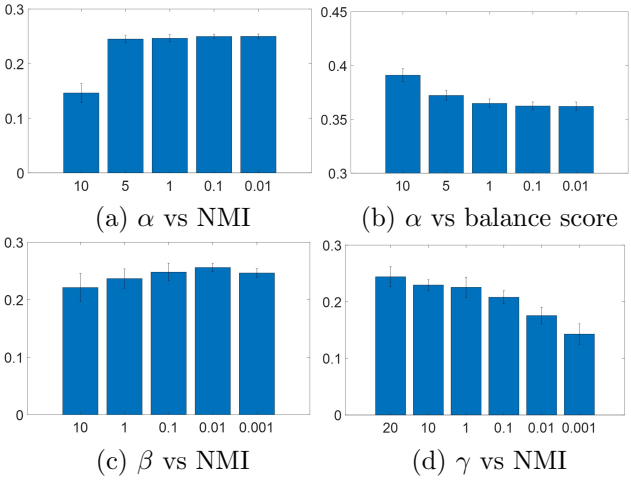
Figure 3: Parameter analysis on Credit Card data set for Fair-MVC-C



(a) α vs NMI (b) α vs balance score

(c) β vs NMI (d) γ vs NMI

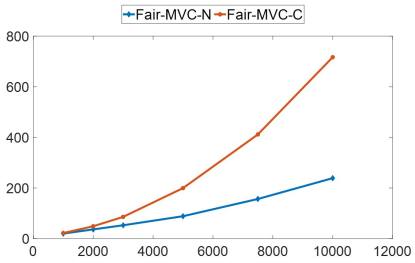
Figure 4: Parameter analysis on Credit Card data set for Fair-MVC-N



(a) α vs NMI (b) α vs balance score

(c) β vs NMI (d) γ vs NMI

Figure 5: Efficiency analysis: number of instances vs running time (Best viewed in color)



A.5 Efficiency Analysis In this subsection, we analyze the efficiency of our proposed algorithm with two different regularization terms (i.e., contrastive regularizer and non-contrastive regularizer) on the Zafar data set. Specifically, we increase the number of samples from 1,000 to 10,000 and record the running time (in seconds) for these two regularizations in Figure 5. The total number of iterations is 1000. The x-axis of this figure is the number of samples and the y-axis is the running time. By observations, we find that the running time is almost linear to the number of samples for non-contrastive regularizer (i.e., Fair-MVC-N) and the running time is quadratic to the number of samples for contrastive regularizer (i.e., Fair-MVC-C). The reason is that for Fair-MVC-C, we need to compute the similarity of any given two samples in the denominator of contrastive regularizer in Equation 3.8, while Fair-MVC-N only computes the similarity of two views for the same sample. Thus, the time complexity of contrastive-based regularization is $O(n^2)$, whereas the time complexity of non-contrastive-based regularization is $O(n)$, where n is the number of samples.