JMLR: Workshop and Conference Proceedings 1:1–6, 2021 Algorithmic Fairness through the Lens of Causality and Robustness

Appendix

Anshuman Chhabra CHHABRA@UCDAVIS.EDU
University of California, Davis
Adish Singla ADISHS@MPI-SWS.ORG
MPI-SWS
Prasant Mohapatra PMOHAPATRA@UCDAVIS.EDU
University of California, Davis

Contents

A Proofs and Derivations for Results in the Main Paper **2**
A.1 Proof for Theorem 3 2
A.2 Derivations for Section 3.1 2

B Experiments for Algorithm 1 with $\mathcal{C}_{\text{SON}}$ and $\mathcal{F}_{\text{social}}$ **3**

C Additional Experiments for Algorithm 2 **4**
C.1 Experiments for Algorithm 2 for $k = 3$ and $k = 4$ 4
C.2 Experiments Comparing Clustering Performance on Davies-Bouldin index (2) and Calinski-Harabasz score (1) 4

A. Proofs and Derivations for Results in the Main Paper

A.1. Proof for Theorem 3

Theorem *(6). Let $V^* \in \mathbb{R}^{V_s^{(t)} \times d}$ be a minimizer for the function $f(V)$ in an iteration t of Algorithm 3 and for $\epsilon > 0$ define $X = \{V \in \mathbb{R}^{V_s^{(t)} \times d} \,|\, f(V) - f(V^*) \leq \epsilon\}$. Let $\mathbb{P}_X$ denote the average probability of successfully sampling from the uniform distribution over X by algorithm $\mathcal{A}$, and it takes n_X samples to realize $\mathbb{P}_X$. Then, the number of queries to f that $\mathcal{A}$ makes to compute $\tilde{V}$ s.t. $f(\tilde{V}) - f(V^*) \leq \epsilon$ with probability at least $1 - \delta$ is bounded as $\mathcal{O}(\max\{\frac{\ln(\delta^{-1})}{\mathbb{P}_X}, n_X\})$.*

Proof The proof follows from Theorem 1 in (6). Since we are considering a sampling-only framework, we set $\lambda = 1$ in Theorem 1 (6) to obtain the result. ■

A.2. Derivations for Section 3.1

In this subsection we discuss the derivation of the single-level reduction using KKT constraints with $\mathcal{C}_{\text{SON}}$ and $\mathcal{F}_{\text{social}}$. First, consider the original strongly convex SON clustering objective:

$$\min_{\mu' \in \mathbb{R}^{n \times d}} \frac{1}{2} \sum_{j=1}^{n} ||U_j - \mu'_j||^2 + \lambda \sum_{i<j} ||\mu'_i - \mu'_j|| \tag{1}$$

As described in the main text, we create the ordering O, the graph G and define its node-arc-incidence matrix I (5) and then reformulate the above objective:

$$\min_{\mu \in \mathbb{R}^{n \times d},\, \eta \in \mathbb{R}^{d \times |O|}} \frac{1}{2}||\mu - U||^2 + \lambda \sum_{i \in O} ||\eta^i|| \quad \text{s.t.} \quad \mu^T I - \eta = 0 \tag{2}$$

It can be verified that objectives (5) and (6) are equivalent. We can even define the dual formulation for the above primal problem (where $\langle , \rangle$ denotes the matrix Frobenius inner-product):

$$\begin{aligned} \max_{\theta \in \mathbb{R}^{n \times d},\, \zeta \in \mathbb{R}^{d \times |O|}} \quad & \langle U^T, \theta \rangle - \frac{1}{2}||\theta||^2 \\ \text{s.t.} \quad & I\zeta^T - \theta = 0 \\ & ||\zeta_i|| \leq \lambda, \forall i \in O \end{aligned} \tag{3}$$

Now, we discuss the KKT conditions. Since the SON objective is strongly convex, we can use the reformulated primal (6) and dual (7) problems to arrive at the KKT conditions:

$$\begin{aligned} &\theta + \mu - U = 0 \\ &\eta - \mathcal{P}(\eta + \zeta) = 0 \\ &\mu^T I - \eta = 0 \\ &I\zeta^T - \theta = 0 \end{aligned} \tag{4}$$

Here $\mathcal{P}(.)$ refers to the proximal operator of the Euclidean norm, therefore $\mathcal{P}(\eta + \zeta) = \max\{0, 1 - \frac{1}{||\eta+\zeta||}\}(\eta + \zeta)$. Since we now have the KKT conditions we can undertake the single-level reduction for problem P1.R.

For this we first have to substitute U for $U \cup V$. As in the main text, $V_s^{(t)}$ denotes $|V|$ in iteration t of Algorithm 1. The number of centers we have will thus be $\mu \in \mathbb{R}^{m \times d}$ where $m = n + V_s^{(t)}$ for $U \cup V$. So now we can use the KKT conditions by replacing U with $U \cup V$ and n with m. The other variables will also then be: $\mu \in \mathbb{R}^{m \times d}, \eta \in \mathbb{R}^{d \times |O|}, \theta \in \mathbb{R}^{m \times d}, \zeta \in \mathbb{R}^{d \times |O|}$. The original problem P1.R for $\mathcal{C}_{\text{SON}}$ and $\mathcal{F}_{\text{social}}$ is:

$$\begin{aligned} \min_{V,\mu} \quad & \mathcal{F}_{\text{social}}(\mu, U) \\ \text{s.t.} \quad & \mu = \mathcal{C}_{\text{SON}}(U \cup V) \end{aligned} \tag{5}$$

Replacing (8) as constraints for the upper-level objective in (9) and removing the lower-level objective gives us the single-level optimization problem we present in the main paper:

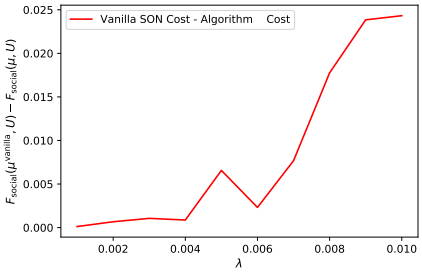
$$\begin{aligned} \min_{V,\mu,\eta,\theta,\zeta} \quad & \mathcal{F}_{\text{social}}(\mu, U) \\ \text{s.t.} \quad & \theta + \mu - (U \cup V) = 0 \\ & \eta - \max\{0, 1 - \frac{1}{||\eta + \zeta||}\}(\eta + \zeta) = 0 \\ & \mu^T I - \eta = 0 \\ & I\zeta^T - \theta = 0 \end{aligned}$$

B. Experiments for Algorithm 1 with $\mathcal{C}_{\text{SON}}$ and $\mathcal{F}_{\text{social}}$

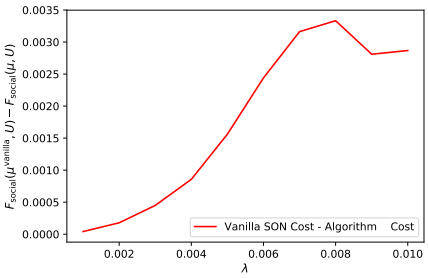
We provide results for Algorithm 1 when using CVX as the solver (3) for the KKT reformulated single-level objective with fairness cost as $\mathcal{F}_{\text{social}}$. We show how the antidote data computed by our optimization consistently reduces the fairness cost compared to vanilla SON clustering while varying the regularization parameter λ from 0.001 to 0.01. That is, we are showcasing the trend in fairness (cost) as a function of the *number of clusters* (ie, number of unique centers) which are determined by the value of λ.

For a given λ we run Algorithm 1 and obtain V, and then re-run regular SON clustering on $U \cup V$. We then compare this obtained fairness cost to the one obtained by regular vanilla SON clustering on U. Note that since CVX does not scale well with large inputs, we subsample all datasets to 100. We also let $\gamma = 0.99$ and for all experiments we obtain $|V| \leq 10$, ie $|V|/|U| \leq 0.1$. We now present the results in Figure 1.

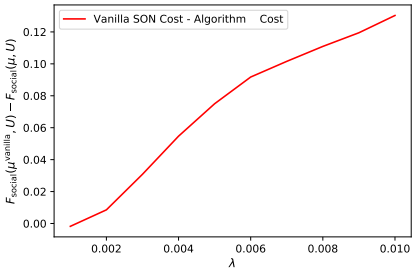
Each of the 4 figures corresponds to the 4 real-world datasets we consider. To further simplify the model we only consider 2 protected groups. As the vanilla SON fairness cost values and fairness cost values for the V obtained by Algorithm 1 could vary widely due to λ, it would be hard to decipher their curves individually. Thus, instead, we present the results as a difference between the vanilla SON fairness cost $\mathcal{F}_{\text{social}}(\mu^{\text{vanilla}}, U)$ and the fairness cost values we obtain as a result of Algorithm 1 denoted by $\mathcal{F}_{\text{social}}(\mu, U)$. That is, the y-axis of the figures represents $\mathcal{F}_{\text{social}}(\mu^{\text{vanilla}}, U) - \mathcal{F}_{\text{social}}(\mu, U)$ and the x-axis represents λ. It is clear to see then that in each figure, if the difference curves are positive on the y-axis, Algorithm 1 outperforms vanilla SON. As can be seen, the centers obtained as a result of clustering on $U \cup V$ where V is obtained from Algorithm 1, lead to more fair clusters than traditional SON clustering (for the $\mathcal{F}_{\text{social}}$ metric).



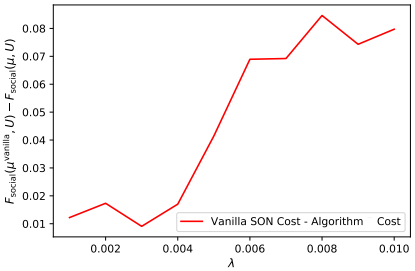
(a) Results for `adult`



(b) Results for `bank`



(c) Results for `creditcard`



(c) Results for `LFW`

Figure 1: Results for Algorithm 1

C. Additional Experiments for Algorithm 2

C.1. Experiments for Algorithm 2 for $k = 3$ and $k = 4$

As mentioned in the paper, we provide experiments for Algorithm 2 for k greater than 2. We obtain results for $k = 3$ and $k = 4$ and present them in Table 1 and Table 2, respectively. The results are tabulated similar to the main paper– we are comparing Algorithm 2 with vanilla clustering for each of the three combinations considered in the paper. As can be observed in both Table 1 and Table 2, Algorithm 2 can achieve improved fairness than traditional clustering. Another interesting trend that can be noted is that more antidote data needs to be added for larger k, in general. This can be seen by observing the $|V|/|U|$ values for both tables individually. This trend is intuitive however, as having more number of clusters can introduce more complexities in reducing fairness cost thus requiring more antidote data addition.

C.2. Experiments Comparing Clustering Performance on Davies-Bouldin index (2) and Calinski-Harabasz score (1)

In this subsection we present additional results for clustering performance of Algorithm 2 compared to other fair clustering approaches, thus extending the results for the Silhouette scores presented in the main paper for $k = 2$. In particular, we consider the Davies-Bouldin index (2) and Calinski-Harabasz score (1). However, these are less indicative and harder to decipher, compared to the more straightforward Silhouette score. This is because these scores are unbounded, and unlike the Silhouette score, do not lie between -1 and 1. It is thus harder

Table 1: Results for Algorithm 2 when $k = 3$

Clustering-Fairness Combination	Dataset	α	$\lvert V\rvert/\lvert U\rvert$	$\mathcal{F}(\mu^{\text{vanilla}}, U)$	$\mathcal{F}(\mu, U)$
Combination #1: $\mathcal{C}_{\text{k-means}}, \mathcal{F}_{\text{balance}}$	`adult`	0.0	0.002	0.0	**-0.6249**
	`bank`	-0.2919	0.00044	-0.2919	**-0.2945**
	`creditcard`	-0.7957	0.00067	-0.7957	**-0.7977**
	`LFW`	-0.7834	0.0015	-0.7834	**-0.7871**
Combination #2: $\mathcal{C}_{\text{k-means}}, \mathcal{F}_{\text{social}}$	`adult`	3.4898	0.002	3.4898	**3.4848**
	`bank`	1.7985	0.00044	1.7985	**1.7983**
	`creditcard`	17.177	0.00067	17.177	**17.165**
	`LFW`	1308.099	0.0015	1308.099	**1307.841**
Combination #3: $\mathcal{C}_{\text{spectral}}, \mathcal{F}_{\text{balance}}$	`adult`	-0.543	0.02	-0.543	**-0.576**
	`bank`	-0.4216	0.02	-0.4216	**-0.4298**
	`creditcard`	-0.8056	0.02	-0.8056	**-0.8261**
	`LFW`	-0.7736	0.02	-0.7736	**-0.7938**

Table 2: Results for Algorithm 2 when $k = 4$

Clustering-Fairness Combination	Dataset	α	$\lvert V\rvert/\lvert U\rvert$	$\mathcal{F}(\mu^{\text{vanilla}}, U)$	$\mathcal{F}(\mu, U)$
Combination #1: $\mathcal{C}_{\text{k-means}}, \mathcal{F}_{\text{balance}}$	`adult`	0.0	0.15	0.0	**-0.5384**
	`bank`	-0.2821	0.0011	-0.2821	**-0.2837**
	`creditcard`	-0.7694	0.0017	-0.7694	**-0.7705**
	`LFW`	-0.7383	0.0038	-0.7383	**-0.7691**
Combination #2: $\mathcal{C}_{\text{k-means}}, \mathcal{F}_{\text{social}}$	`adult`	3.0299	0.005	3.0299	**3.01**
	`bank`	1.315	0.0011	1.315	**1.301**
	`creditcard`	15.989	0.0017	15.989	**15.976**
	`LFW`	1242.913	0.0038	1242.913	**1242.570**
Combination #3: $\mathcal{C}_{\text{spectral}}, \mathcal{F}_{\text{balance}}$	`adult`	-0.3584	0.05	-0.3584	**-0.5354**
	`bank`	-0.273	0.05	-0.273	**-0.476**
	`creditcard`	-0.8425	0.05	-0.8425	**-0.9307**
	`LFW`	-0.7574	0.065	-0.7574	**-0.7658**

to discern and compare different clusterings in an objective manner. The Davies-Bouldin index is an inverse score, in that a lower value signifies better clustering performance, with the lowest possible being 0. This is different from the Calinski-Harabasz score, which gives well-formed clusters higher values, and ill-formed clusters lower ones. Also as in the main paper, we let $k = 2$.

We then present the results for Algorithm 2 with the Davies-Bouldin index being used as a clustering performance metric in Table 3. Similar results are shown for the Calinski-Harabasz score in Table 4. It can be observed from the values obtained in the table that we generally have very similar performance to the state-of-the-art fair clustering algorithms, while providing improved fairness (as the results show in the main text). In some cases, we can see that we have better performance, such as in the case of comparisons with the Fair-Lloyd algorithm of (4) for Combination #2 and the `creditcard` and `LFW` datasets. We can also observe that while the Sihouette score and Davies-Bouldin index give more reasonable differences in values, values for the Calinski-Harabasz score vary widely and are not easy to justify. However, overall, we can see that Algorithm 2 obtains competitive clustering performance while providing improved fairness on these two metrics as well.

Table 3: Results for Davies-Bouldin index (2)

Clustering-Fairness Combination	Dataset	SOTA Fair Algorithm	Algorithm 3
Combination #1: $\mathcal{C}_{\text{k-means}}, \mathcal{F}_{\text{balance}}$	adult	1.977075899	1.987745353
	bank	1.32144642	1.323176876
	creditcard	1.535894783	1.545304948
	LFW	1.955060538	1.957352428
Combination #2: $\mathcal{C}_{\text{k-means}}, \mathcal{F}_{\text{social}}$	adult	0.253210096	0.253863016
	bank	1.320149817	1.352818868
	creditcard	1.549144574	1.346206321
	LFW	1.964263126	1.6180073
Combination #3: $\mathcal{C}_{\text{spectral}}, \mathcal{F}_{\text{balance}}$	adult	1.855412037	1.939762524
	bank	1.411452861	9.265055739
	creditcard	1.839992765	3.568314803
	LFW	1.918287593	2.04272442

Table 4: Results for Calinski-Harabasz score (1)

Clustering-Fairness Combination	Dataset	SOTA Fair Algorithm	Algorithm 3
Combination #1: $\mathcal{C}_{\text{k-means}}, \mathcal{F}_{\text{balance}}$	adult	1745.35127	1670.931497
	bank	14025.77526	13935.82229
	creditcard	6411.585792	6365.048578
	LFW	3208.171468	3200.597262
Combination #2: $\mathcal{C}_{\text{k-means}}, \mathcal{F}_{\text{social}}$	adult	2170.257982	2054.422064
	bank	13923.43025	14677.25017
	creditcard	6411.04393	10905.30975
	LFW	3193.024907	6293.817629
Combination #3: $\mathcal{C}_{\text{spectral}}, \mathcal{F}_{\text{balance}}$	adult	171.7174597	128.9134408
	bank	284.5544472	6.931179858
	creditcard	197.9853811	51.74285991
	LFW	249.6535979	181.497717

References

[1] T. Caliński and J Harabasz. A Dendrite Method for Cluster Analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[2] David L. Davies and Donald W. Bouldin. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1(2):224–227, 1979.

[3] Steven Diamond and Stephen P. Boyd. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *JMLR*, 17:83:1–83:5, 2016.

[4] Mehrdad Ghadiri, Samira Samadi, and Santosh S. Vempala. Socially Fair k-means Clustering. In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 438–448, 2021.

[5] André A. Keller. Chapter 3 - elements of technical background. In *Mathematical Optimization Terminology*, pages 239–298. 2018.

[6] Yang Yu and Hong Qian. The Sampling-and-learning Framework: A Statistical View of Evolutionary Algorithms. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC*, pages 149–158, 2014.