Supplementary materials of "Fairness-Aware Streaming Feature Selection with Causal Graphs"

Leizhen Zhang[1], Lusi Li[1], Di Wu[2], Sheng Chen[3] and Yi He[1] ✉

I. Preliminaries

A. Definitions

Definition 1 (D-Separated): : Nodes X and Y are D-Separated if all paths are blocked path between them, otherwise, they are d-connected. To formally define d-separation, a path p is considered blocked by a set of nodes Z if either

(1) p contains a chain A $\rightarrow$ B $\rightarrow$ C or fork A $\leftarrow$ B $\rightarrow$ C with a middle node B in Z, or

(2) p contains a collider A $\rightarrow$ B $\leftarrow$ C with collision node B not in Z and none of its descendants in Z.

If Z blocks all paths between X and Y, then X and Y are D-Separated and thus independent given Z.

B. Discrimination

For simplicity, we assume the label has two values: positive(granted) and negative(rejected), and the sensitive attribute has two values: male(favored) and female(rejected). We adopt statistical parity, as referenced in [1], to assess discrimination. The first step must be calculated independently.

DR (deprived-rejected): females rejected a benefit.

DG (deprived-granted): females granted a benefit.

FR (favored-rejected): males rejected a benefit.

FG (favored-granted): males granted a benefit.

Step 2: Calculate discrimination using this formula.

$$Disc(D) = \frac{FG}{FG + FR} - \frac{DG}{DG + DR} \tag{1}$$

This formula shows that if the share of approvals in the favored group is more than that in the deprived group, then the deprived group can say they are being discriminated against.

C. Online streaming feature selection

This paper's research is based on the online streaming feature selection scenario, where features continuously emerge. This results in an uncertain or even potentially infinite feature set size, unlike having all features predetermined. Here, features are dynamically generated and sequentially presented [2]. The key online vs offline difference is that offline obtains all features in advance, while online cannot. This limitation prevents online from obtaining complete information when analyzing causal relationships, which may adversely affect the final result. Another interesting online situation is promptly selecting an important arriving feature can conserve time and expedite identifying crucial features, unlike waiting for the complete feature set before selecting. Therefore, if our method achieves a result similar to offline, it would prove our method's effectiveness.

II. Algorithm

A. Optimizing the Accuracy-Fairness Tradeoff

1) Why is it not suitable to delete only sensitive feature or delete all features in the graph G_S directly?

Simply deleting only the sensitive feature S without other processing might not significantly decrease discrimination, as other features causally related to S can reconstruct its information [3]. Deleting all features in G_S can decrease the discrimination, but since G_S and G_Y might intersect, removing the intersection will cause G_Y to lose necessary information and significantly decrease accuracy.

2) Why can replacing the intersection features with $AD1$ and $AD2$ achieve our two goals?

When sensitive causal graph G_S and label causal graph G_Y intersect, we can utilize the $AD1$ to replace the intersection. The key is that $AD1$ is irrelevant from sensitive feature S and $AD1$ contain some information of intersection. Our approach analyzes the causal relationship between intersection features and a$AD1$. Based on this, we choose features that reflect the intersection information and are D-separated with sensitive S to replace the intersection.

Achieve fairness: Intersection features belong to both G_S and G_Y, not D-separated from sensitive S, increasing discrimination. Replacing with $AD1$ makes the final selected features D-separated from sensitive S, decreasing discrimination.

Maintain accuracy: We identify the causal relationships between $AD1$ and intersection features, selecting those with relationships to replace intersections. The selected $AD1$ have causal relationships with intersections and can reconstruct their information. This maintains accuracy without significant reduction.

B. Conditional independence tests

We employ two methodologies to conduct the conditional independence tests between attributes: Fisher's z Test and the G^2 test [4], [5]. The former is tailored for continuous data analysis, while the latter is apt for discrete data. Both

*This work was supported in part by the National Science Foundation (NSF) under Grants CNS-2245918, IIS-2245946, and IIS-2236578, and by the Commonwealth Cyber Initiative (CCI).

[1] Department of Computer Science, Old Dominion University, USA. E-mails: {lzhan011, l3li}@odu.edu, yihe@cs.odu.edu

[2] College of Computer and Information Science, Southwest University, China. E-mails: wudi.cigit@gmail.com

[3] Center for Advance Computer Studies, University of Louisiana, Lafayette, USA. E-mails: sheng.chen@louisiana.edu

✉ Corresponding Author: Dr. Yi He (yihe@cs.odu.edu)

methods yield a value denoted by ρ. Setting a significance threshold, α, usually at levels like 0.01 or 0.05, helps ascertain the acceptance or rejection of the null hypothesis of conditional independence. Specifically: If $\rho <= \alpha$, we do not reject the null hypothesis, implying the two attributes maintain conditional independence given a specific set of features. Conversely, if $\rho > \alpha$, the null hypothesis is rejected, suggesting the attributes exhibit conditional dependence when considering the same feature set.

For the continuous features, We use the Fisher's z Test to calculate the degree of relevance. The formula of Fisher's z Test can be described as:

$$z = \frac{1}{2}\sqrt{n - |X_c| - 3}(\ln(\frac{1+\zeta}{1-\zeta})) \tag{2}$$

where n represents the sample size, $|X_c|$ denotes the the number of distinct values of conditional feature X_c, and ζ represents the conditional correlation coefficient between features X_a and X_b under the condition of X_c. ζ is calculated by the formula:

$$\zeta(X_aX_b|X_c) = \frac{\zeta(X_aX_b) - \zeta(X_bX_c)\zeta(X_aX_c)}{\sqrt{(1-\zeta^2_{X_aX_c})(1-\zeta^2_{X_bX_c})}} \tag{3}$$

where $\zeta(X_aX_b)$ represent the correlation coefficient between features X_a and X_b, $\zeta^2_{X_aX_c}$ represents the square of $\zeta(X_aX_b)$, and $\zeta(X_aX_b|X_c)$ represents the the conditional correlation coefficient of the features X_a and X_b in the condition of X_c.

In this work, $\zeta(X_aX_b)$ can be calculated by:

$$\zeta(X_aX_b) = \frac{\sum_{i=1}^{n}(x_a^i - \overline{X_a})(x_b^i - \overline{X_b})}{\sqrt{\sum_{i=1}^{n}(x_a^i - \overline{X_a})^2}\sqrt{\sum_{i=1}^{n}(x_b^i - \overline{X_b})^2}} \tag{4}$$

where X_a = $[x_a^1, x_a^2, x_a^3, ..., x_a^n]^\top$, X_b = $[x_b^1, x_b^2, x_b^3, ..., x_b^n]^\top$, $\overline{X_a}$ represents the average value of the feature X_a, and X_a^i represents the i-th value of the feature X_a. $\zeta(X_aX_i)$, $\zeta(X_bX_i)$ can be calculated via the formula 3.

For the discrete features, we use the G^2 test to calculate the relevance degree. If X_a and X_b are conditionally independent given X_c, the formula of G^2 test can be described:

$$G^2 = 2\sum_{i,j,k} N_{abc}^{ijk} \ln \frac{N_{abc}^{ijk} N_c^k}{N_{ac}^{ik} N_{bc}^{jk}} \tag{5}$$

where N_{abc}^{ijk} represents the number of features X_a, X_b, X_c, when X_a = i, X_b = j, X_c = k. Additionally, N_{ac}^{ik}, N_{bc}^{jk} and N_c^k are defined in a similar manner.

C. Time complexity Analysis

In the online streaming feature selection scenario, the main time consumption of our algorithms consists of two steps, the first step is aligning the different features, the second step is the feature selection operation. Herein, we will analyze the time complexity in these two steps.

The time complexity of alignment is considered in a scenario where features are processed sequentially. For instance, we have two features, X_a and X_b, representing vectors: $X_a = [x_a^1, x_a^2, x_a^3, ..., x_a^n]^\top$ and $X_b = [x_b^1, x_b^2, x_b^3, ..., x_b^m]^\top$, respectively. When we receive the two features, the elements order of X_a and X_b might be different and not initially aligned. However, the calculation of their relevance requires the alignment of X_a and X_b before performing the computation. The time complexity of aligning two features ranges from $\mathcal{O}(n \times \log(m))$ to $\mathcal{O}(n \times m)$, where the lengths of the two features are denoted as n and m, respectively. In traditional offline feature selection methods, the alignment operation is full-scale operation, meaning when a new(t-th) feature comes in, it requires aligning the new feature with all previously arrived features every time, resulting in an alignment time complexity between $\mathcal{O}(|t| \times n \times m)$

Algorithm 1: $SFCF$ algorithm

input : Class label $Y := \{0,1\}^N$, *protected* feature $S \notin \mathcal{X}$, streaming features $\mathcal{X} = \{X_i \mid i = 1, \ldots, D\} \in \mathbb{R}^{D \times N}$

output: $\mathcal{F}_i^*$(Selected fairness features)

$Irrelevant_0(Y) = \varnothing$; $Irrelevant_0(S) = \varnothing$; $\mathcal{CFS}(Y) = \varnothing$; $\mathcal{CFS}(S) = \varnothing$

Target $T \in \{Y, S\}$

// Causal Graph Construction with Markov Blanket

for $X_i(i \leftarrow 1$ **to** $D)$ **do**

  // Null-Conditional Independence analysis

  Using Fisher's z Test or G^2 to calculate P(X_iT), P(X_i)P(T)

  if *P(X_iT) == P(X_i)P(T)* **then**

    $\mathcal{I}rrelevant_i(T) = \mathcal{I}rrelevant_{i-1}(T) \cup X_i$

  else

    $\mathcal{CFS}(T) = \mathcal{CFS}(T) \cup X_i$

    // Redundancy analysis

    for $X_m(m \leftarrow 0$ **to** $|\mathcal{CFS}(T)|)$ **do**

      powerset = powerset($\mathcal{CFS}(T) \setminus X_m$)

      for $X_j(j \leftarrow 0$ **to** $|powerset|)$ **do**

        Given X_j, Using Fisher's z Test or G^2 to calculate P($T|X_m$, X_j), P($T|X_j$)

        if *P($T|X_m$, X_j) == P($T|X_j$)* **then**

          $\mathcal{R}edundant_i(T) = \mathcal{R}edundant_{i-1}(T) \cup X_m$

    $\mathcal{MB}_i(T) = \mathcal{CFS}(T) \setminus Redundant_{i-1}(T)$

  // Optimizing the Accuracy-Fairness Tradeoff

  $\mathcal{IA}_i = \mathcal{MB}_i(S) \cup \mathcal{R}edundant_i(S)$

  $\mathcal{A}_i = \mathcal{MB}_i(Y) \cup \mathcal{R}edundant_i(Y) \setminus \mathcal{IA}_i$

  $\mathcal{MI}_i = \mathcal{MB}_i(Y) \cup IA_i$

  $\mathcal{RI}_i = \mathcal{MB}_i(Y) \setminus \mathcal{MI}_i$

  $\mathcal{AD}1_i = \mathcal{ICRF}_i(Y) \cap A_i$

  $\mathcal{F}_i^* = \mathcal{RI}_i \cup \mathcal{AD}2_i = \mathcal{MB}_i(Y) \setminus \mathcal{MI}_i \cup \mathcal{AD}1_i$

  $\mathcal{AD}2_i = \mathcal{ICRF}_i(Y) \cap \mathcal{R}edundant_i(S)$

  $\mathcal{F}_i^* = \mathcal{RI}_i \cup \mathcal{AD}1_i = \mathcal{MB}_i(Y) \setminus \mathcal{MI}_i \cup \mathcal{AD}1_i$

TABLE I: CHARACTERISTICS OF THE STUDIED DATASETS

Dataset	#Inst.	#Feat.	Sensitive S
D1	48,842	14	Sex
D2	6,173	12	Race
D3	1,000	25	Age
D4	30,000	24	Sex
D5	1,994	123	Race

and $\mathcal{O}(|t| \times n \times \log(m))$. In our online feature selection method, due to our alignment operation being incremental computation, when the t-th feature arrives, we align it with the previously aligned results, resulting in a reduced time complexity ranging from $\mathcal{O}(n \times \log(m))$ to $\mathcal{O}(n \times m)$. Hence, in the alignment step, our algorithms have a lower time complexity than offline methods.

After alignment, feature selection is performed. The worst-case time complexity for traditional offline feature selection methods is $\mathcal{O}(|t| \times k^{|t|})$, where $|t|$ represents the number of samples and k represents the maximum size of the conditional set. For our online feature selection, the worst-case time complexity is $\mathcal{O}(|MB(T)| \times k^{|MB(T)|})$, where $|MB(T)|$ denotes the number of features in MB(T) and MB(T) is the Markov Blanket building result for the target T. If all features are selected into MB(T), offline and online methods exhibit the same time complexity. However, it's rare that all features are included in MB(T). Therefore, our algorithm offers a lower time complexity than offline methods.

After the analysis of time complexity in two steps, we can conclude that our algorithm has a lower time complexity than the offline methods.

III. Experiments

A. Datasets

We use five benchmark datasets, frequently employed in fairness literature, to robustly assess $SFCF$'s effectiveness across diverse applications including income, credit card, and crime domains. The characteristics and statistics of these datasets are presented in Table I.

- **Adult dataset (D1)** [6]. This dataset, from the 1994 Census database and in the UCI repository, has features like demographic info, education level, occupation, and income above or below $50k. The prediction is if income is above or below $50k. Sex, with male and female, is the sensitive feature.
- **COMPAS dataset (D2).** This dataset contains info on crime recidivism in Broward County, Florida during 2013-2014. It has features like prior convictions, age, and charge degree severity. The label is two-year recidivism, indicating if this instance is a recidivism in two years. Race is a sensitive feature, and we only looked at African-American and Caucasian people.
- **German Credit dataset (D3)** [7]. This dataset contains financial info on credits given to individuals in Germany who borrowed money from a bank. The label indicates if a person will repay the credit, and age is the sensitive feature. Each entry represents a person classified as having good or bad credit based on specific attributes.
- **Credit Card dataset (D4)** [8]. This dataset, available in the UCI repository, contains info on default payments, demographic factors, payment history, and bill statements of credit card clients in Taiwan. The label indicates if the user will default in the next month. Sex(male and female) is treated as a sensitive feature.
- **Communities Crime dataset (D5)** [9]. This dataset, available in the UCI repository, combines socio-economic, law enforcement, and crime data from the USA. The label indicates if a community has a high crime rate (above the 70th percentile). The 'racepct-black' column, representing the percentage of the African American population, is treated as a sensitive feature and discretized at the 70-th percentile threshold.

B. Competitors

We have selected six distinct methods as competitors to compare with our algorithm from various perspectives. Here, we introduce competitors' main ideas and explain why select them as our comparators.

- **Baseline.** The Baseline used all the features without feature selection as input for the classifier. This serves as a reference to determine the effectiveness of our algorithm.
- **Remove Sensitive.** This method removes the sensitive feature from the datasets and keeps the remaining features. This is to demonstrate that only removing the sensitive feature cannot significantly decrease the EO score.
- **Kamiran-massaging/Kamiran-reweighting** [10]. Dataset massaging adjusts the labels or weights of training dataset features to decrease the EO score from the input data by improving deprived-granted and favored-rejected values while decreasing favored-granted and deprived-rejected values.
- **Capuchin** [11]. After users identify admissible and inadmissible features, Capuchin adjusts their weights to achieve D-separation between the sensitive feature and label.
- **FairExp** [12]. This approach employs feature engineering and multi-objective feature selection to make a trade-off between high accuracy and low fairness.

The competitors mentioned above, including Baseline, Remove Sensitive, Kamiran-massaging , Kamiran-reweighting, Capuchin, and FairExp are offline algorithms to deal with algorithmic fairness problems.

- **OSFS** [2]. It is a feature selection method for streaming features using Markov Blanket to identify MB, redundant, irrelevant feature sets. OSFS works in streaming feature selection scenarios like ours, but does not aim to reduce classifier discrimination.
- $SFCF$**-RI/AD1/AD2.** $SFCF$-RI/AD1/AD2 are our methods. After build MB(Y) and MB(Y), $SFCF$-RI refers to removing the intersection features between

MB(Y) and MB(S), $SFCF$-AD1/AD2 refer to using $\mathcal{A}D1$ or $\mathcal{A}D2$ to replace the intersection.

C. Evaluation Protocol

Considering our goals, we need to use two main metrics, equalized odds(EO) and accuracy, to make trade-offs between accuracy and fairness goals.

1) Equalized odds(EO): For sensitive feature S containing group female(f) and male(m), the class label Y contains the values 1 and 0, in which 1 represents positive and 0 represents negative. $\hat{Y}$ is the predicted result. we can use the following formula to calculate TPR_f, TPR_f, FPR_m and FPR_m and then calculate equalized odds(EO).

$$TPR_f = \frac{TP_f}{TP_f + FN_f} \tag{6}$$

where TP_f = P($\hat{Y}$=1|Y=1,S=f), FN_f = P($\hat{Y}$=0|Y=1,S=f)

$$FPR_f = \frac{FP_f}{FP_f + TN_f} \tag{7}$$

where FP_f = P($\hat{Y}$=1|Y=0,S=f), TN_f = P($\hat{Y}$=0|Y=0,S=f)

The equalized odds (EO) fairness metric ensures the machine learning model performs equally well across diverse groups. Unlike demographic parity, EO requires not only independence of predictions from sensitive group membership but also equivalent distribution of false positive and true positive rates among groups. This achieves a higher level of fairness, mitigating discrimination and promoting a more equitable and accurate model. Therefore, we use it to measure discrimination.

We can calculate TPR_f by $TPR_f = \frac{TP_f}{TP_f+FN_f}$, Similarly, we can calculate FPR_f, TPR_m and FPR_m. Then we can use them to calculate EO:

$$EO = Max(|TPR_f - TPR_m|, |FPR_f - FPR_m|) \tag{8}$$

According to the EO formula, EO is between 0 and 1. A larger EO means more unfairness, while a smaller EO means more fairness. For a classifier to be fair for groups female(f) and male(m), it means equalized odds of 0.

2) Accuracy(ACC): Accuracy is commonly used in classification problems, measures the percentage of predictions that match the actual or ground label. The formula for accuracy is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

where TP, TN, FP, and FN represent True Positive, True Negative, False Positive and False Negative, respectively. In this study, accuracy is used to achieve the accuracy goal.

REFERENCES

[1] S. Verma and J. Rubin, "Fairness definitions explained," in *Proceedings of the international workshop on software fairness*, 2018, pp. 1–7.

[2] X. Wu, K. Yu, H. Wang, and W. Ding, "Online streaming feature selection," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 1159–1166.

[3] F. Kamiran, T. Calders, and M. Pechenizkiy, "Discrimination aware decision tree learning," in *2010 IEEE international conference on data mining*. IEEE, 2010, pp. 869–874.

[4] R. E. Neapolitan *et al.*, *Learning bayesian networks*. Pearson Prentice Hall Upper Saddle River, 2004, vol. 38.

[5] P. Spirtes, C. N. Glymour, and R. Scheines, *Causation, prediction, and search*. MIT press, 2000.

[6] B. Becker and R. Kohavi, "Adult," UCI Machine Learning Repository, 1996, DOI: https://doi.org/10.24432/C5XW20.

[7] H. Hofmann, "Statlog (German Credit Data)," UCI Machine Learning Repository, 1994, DOI: https://doi.org/10.24432/C5NC77.

[8] I.-C. Yeh, "default of credit card clients," UCI Machine Learning Repository, 2016, DOI: https://doi.org/10.24432/C55S3H.

[9] M. Redmond, "Communities and Crime," UCI Machine Learning Repository, 2009, DOI: https://doi.org/10.24432/C53W3X.

[10] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowledge and information systems*, vol. 33, no. 1, pp. 1–33, 2012.

[11] B. Salimi, L. Rodriguez, B. Howe, and D. Suciu, "Interventional fairness: Causal database repair for algorithmic fairness," in *Proceedings of the 2019 International Conference on Management of Data*, 2019, pp. 793–810.

[12] R. Salazar, F. Neutatz, and Z. Abedjan, "Automated feature engineering for algorithmic fairness," *Proceedings of the VLDB Endowment*, vol. 14, no. 9, pp. 1694–1702, 2021.