Cora Hyun Jung (hj1399), Emmy Phung (mtp363), Sujeong Cha (sjc433), Wendy Hou (wh916)

Sentencing Under Fairness Constraints:
Finding a Fair Policy with Offline Contextual Bandit

I. Introduction

While the importance of fairness in the criminal justice system is evident, it can be difficult to define what fairness means and achieve it in real life. As machine learning is more and more involved in the decision-making processes in our criminal justice system, it is crucial to think about how the algorithms ensure fairness among different groups of defendants when deciding the appropriate sentencing for them. The criminal sentencing practice has a significant impact on criminal recidivism, meaning the defendant commits the same crime in a short period after his or her release from sentencing [1].

RobinHood is an offline contextual bandit algorithm, designed to satisfy customizable fairness constraints while lowering the recidivism rate when choosing sentencing actions [2]. In this project, we replicate RobinHood's performance on a criminal recidivism experiment in [2] using data released by ProPublica, prepare our own sentencing and recidivism data from the New Orleans District Attorney (NODA) dataset provided by the World Bank Group, and apply the RobinHood algorithm to find a sentencing policy that minimizes recidivism rate under fairness constraints. For this project, we define fairness as minimal differences in false-positive rates between two groups of defendants. We then experiment with different subsets of covariates that are related to criminal sentencing and compare RobinHood's performances on our NODA dataset to two baseline models: Naive and Offset Tree. Our results show that, in some circumstances, RobinHood is indeed the fairest algorithm while still maintaining relatively high reward returns.

II. Models

This section defines three offline contextual bandit models adopted from [2] that we use in our experiments. The three models are Naive, Offset Tree [3], and RobinHood. The RobinHood algorithm, which can produce optimal policies under a given fairness constraint, is our primary subject of research and will be compared against Naive and Offset Tree based on both expected rewards and our fairness metrics. We will first explain the bandit set up in the original criminal recidivism experiment on the ProPublica dataset and our set up on the NODA dataset. Next, we will elaborate on how the three models work as well as how they can contribute to our learning objectives.

a. ProPublica Replication

This experiment investigates the potential racial bias of a classification model that assesses the probability of recidivism of defendants given their demographic information (ie. race, gender, age) and the details of their crimes (ie. charge degree, juvenile felony count). In this problem, the agent is tasked with predicting the likelihood of a defendant recidivating on a scale of 1 to 10. This is referred to as the decile score and serves as the action space in the original experiment. A decile score of 5 and above indicates that the individual is likely to reoffend and a decile score less than 5 indicates that the individual is not likely to do so. According to the code implementation, the agent receives a reward of -1 if recidivism is observed and 1 otherwise. We find this inconsistent with what is mentioned in [2]. In this setup, maximizing the expected reward is equivalent to improving the calibrated classification model that produces the decile score.

The fairness constraint denoted $g(\theta)$ is defined as the difference between the false-positive rates of the predictions given to the two races, Caucasian and African American [Eq 1]. This gap represents the difference in treatments that the agent assigns to defendants belonging to the protected and unprotected classes. A fair policy should minimize this gap. We have successfully replicated this experiment using three models, Naive, Offset Tree, and RobinHood. Results are shared in Appendix A.

$$g(\theta) := |P(\hat{A} = 1,\ A = 0 \mid Caucasian) - P(\hat{A} = 1,\ A = 0 \mid African\ American)| - \epsilon \quad \cdots\ Eq\ 1$$

(where ϵ is a threshold value, $\hat{A}$ is the predicted label, recidivate or not, and A is the true label).

b. NODA Adaptation

Our experiment uses the criminal data given by the Orleans Parish District Attorney's Office to study potential racial and gender discrimination in prosecutorial decisions. In our setup, the agent's task is to assign appropriate sentencing treatments to the defendants conditioning on their demographic information (ie. race, sex, age, criminal flag), crime-related details (ie. lead category, charge severity level, bond set amount), and the demographic information of other selected parties (ie. judges, trial assistant district attorney, screening assistant district attorney). Since sentencing treatments are uncountable, including confinement, probation, and conditions of different periods, we need to design an action space that is 1) discrete, 2) reasonably small, and 3) has a clear ranking in terms of severity so that we can compare whether a defendant of a certain race or gender receives a more severe or lenient treatment. We decide to limit our actions to confinement only and construct our action space on a scale of 0 to 4, indicating the length of a confinement period (ie. 0: 0-year incarceration, 1: < 0.5-year incarceration, 2: 0.5-2 year incarceration). We will explain how we choose this ranking or label encoding system in the Dataset section. The agent will receive a reward of 1 if the defendant does not get sentenced for the same lead charge category within 2 years after release and -1 otherwise. Since our objective for this project is to minimize the rate of recidivism under a fairness constraint, it is equivalent to train the agent to choose an appropriate sentencing treatment that helps prevent recidivism.

We consider an action of scale 2 and above (meaning greater than 2-year incarceration) as severe and the others as lenient. We define false-positive as assigning a severe treatment to someone who does not recidivate. Therefore, our fairness constraint, $g(\theta)$, captures the disparity between the rate of assigning false-positive treatments to defendants of protected versus an unprotected race or gender class [Eq 2 & 3]. We expect a fair policy should minimize this gap and obtain a value close to zero for the fairness metrics. We use three models, Naive, Offset Tree, and RobinHood to find the optimal policy for assigning sentencing treatments, which will be explained in the next subsection.

$$g_{race}(\theta) := |P(A = Severe,\ Recid = No \mid White) - P(A = Severe,\ Recid = No \mid Black)| - \epsilon \quad \cdots Eq\ 2$$

$$g_{sex}(\theta) := |P(A = Severe,\ Recid = No \mid Female) - P(A = Severe,\ Recid = No \mid Male)| - \epsilon \quad \cdots Eq\ 3$$

(where ϵ is a threshold value, A is the assigned action and *Recid* indicates whether the recidivism is observed).

c. Naive

The Naive algorithm aims at maximizing the expected rewards without considering the fairness constraints. It estimates the expected reward using the importance weighting approach, where the importance weight (IW) is computed as follows.

$$IW = \pi_{new}(A) \,/\, \pi_{logging}(A)$$

The model uses an optimizer called CMA-ES [2] provided in the Python package *cma* to find and return an optimal policy. The Naive algorithm uses the same method of estimating expected returns and also the same optimizer as RobinHood. However, unlike the RobinHood algorithm, the Naive approach does not take into consideration the fairness constraints.

d. Offset Tree

The Offset Tree model is implemented using a pre-built package named *contextualbandits* [3]. This algorithm aims at solving the partial feedback problem where the rewards for unselected actions are not observed in the data. This problem is directly relevant to our case, where we only observe the reward for choosing an action that is the same as the actual sentencing treatment that a defendant received. To cope with this problem, the Offset Tree algorithm reduces the original k-arm contextual bandit to at most k - 1 binary arm bandit problems and compares rewards between pairs of actions to obtain estimates for the unobserved rewards using regret analysis [3]. Therefore, the difference between Naive and Offset Tree algorithms is that Naive uses importance weighting to estimate the expected reward while Offset Tree uses a form of imputed reward estimates. Similar to Naive, Offset Tree does not take into account our fairness constraints. Therefore, they both serve as our baselines to compare with RobinHood in terms of both fairness metrics and expected rewards. One advantage of the Offset Tree algorithm is that the reduction is performed in a tree-based fashion and thus proven to be computationally more efficient at both training and testing run time.

e. RobinHood

The RobinHood algorithm is our primary focus in this project as it is the only model that can produce an optimal policy under user-defined fairness constraints. RobinHood aims at producing an optimal policy that maximizes the expected reward and, more importantly, it only returns a policy if the policy satisfies fairness constraints with a user-specified probability threshold. If fair solutions do not exist due to conflicting fairness constraints or limited data, the model returns *No Solution Found* (*NSF*). Compared to the two algorithms mentioned above, the novelties of RobinHood include 1) the ability to add any fairness constraints and specify the probability threshold at which the returned policy's violation of one of the constraints is acceptable; and 2) the fact that it can notify if no fair solutions are found.

In our experiment, a fairness constraint (or a behavioral constraint) is represented by $g_i(\theta)$ and a confidence level δ, which has the form :

$$P(g_i(a(D)) \leq 0) \geq 1 - \delta_i$$

where $a(D)$ means the algorithm applied on the logging data D. If $g_i(\theta) \leq 0$, then the policy with parameters θ is fair. Consider Eq 2 as an example:

$$g_{race}(\theta) := |P(A = Severe,\ Recid = No \mid White) - P(A = Severe,\ Recid = No \mid Black)| - \epsilon \quad \cdots Eq\ (2)$$

This constraint ensures that the difference in sentencing assignments given to the protected and unprotected racial classes is less than ϵ. Having a set of fairness constraints defined, RobinHood partitions the training data into a candidate selection set D_c and a safety set D_s. The algorithm then tests whether each candidate policy satisfies the fairness constraints. If a policy fails to satisfy any of the constraints, it will no longer be considered. The model will return an optimal policy that maximizes rewards under fairness constraints. Detailed pseudocode can be found in Appendix B.

III. Dataset

The data we use for this project comes from NODA, a comprehensive dataset of criminal cases processed by the Orleans Parish District Attorney's Office in New Orleans, Louisiana from January 1988 to November 1999, established under the guidance of an attorney named Harry Connick. The codebook for NODA provides descriptions of the tables and variables in this dataset. In total, NODA contains detailed records for approximately 590,000 charges, 280,000 cases, and 145,000 unique defendants. For each defendant, the dataset contains demographic information such as age, gender, race, ethnicity, and home address. For each case, information about the list of charges, prosecutors, and the judge assigned to the case are provided. There are 30 tables and more than 300 covariates that give us information about defendants, prosecutors, judges, nature of the charges, etc.

One of our crucial early steps for this project is to narrow down the covariate space for our offline contextual bandit model. For example, the model should not consider the demographic information (e.g. race or gender) of the defendants that we use as "treatments" in our experiments, while carefully examining possible covariates that may inherently contain race or gender bias when deciding the fair sentencing policy. In contrast, some of the covariates we utilize are the types of crimes committed, numbers of defendants involved, whether the defendants are first-time offenders, whether the defendants have criminal records, etc.

a. Data Preprocessing

The data is constructed in a way that the final dataset contains every possible combination of the defendant-case pairs. The original dataset consists of 30 tables (e.g. the 'Sentence' table provides sentencing information on a charge level, the 'Defendant' table contains demographic information about defendants, the 'Charge' table contains information about charges filed). We consider the data found in the 'Sentence' table as our base. Only tables that contain our selected covariates are merged with the 'Sentence' table using unique defendant IDs as the keys. We then narrow down the covariates from 300 to 35 by going over the entire covariate list [Appendix C]. All the cases with juvenile defendants are dropped (those who are younger than 18 years old). Also, we drop any instances that have a released date after 1997, since we have data only up to 1999. If the released date is beyond this time frame, then we won't be able to observe the recidivism.

b. Data Imputation

Out of the 31 selected covariates, we find that 28 covariates have missing values. For numerical covariates, the missing data are imputed using mean or median values, and a dummy variable is created to indicate whether the instance was originally missing. As an example, for the AGE_TRIAL_ADA covariate, ages that are either negative or less than 20 are considered invalid, and then all the invalid and missing data are replaced with the median age calculated from valid data. Then the dummy variable AGE_TRIAL_ADA_ISNA, which contains a binary value (0 or 1), is created to indicate whether each instance is imputed or not. For categorical covariates, all the missing and invalid values are categorized into one 'NA' category.

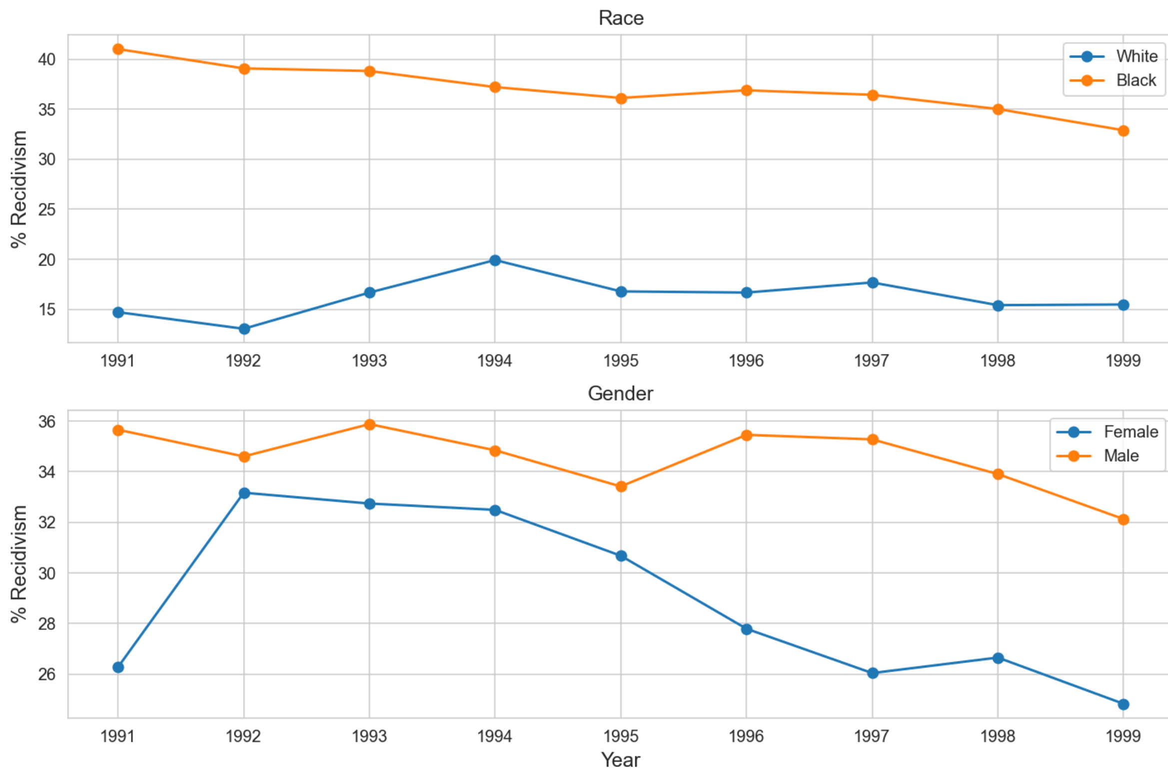
c. Recidivism Flag

The Recidivism flag is created as a dummy variable that has a value of 1 if the defendant commits the same crime (the same lead charge category) within two years of his or her release from confinement. In other words, if a defendant appears again with the same charge category but a different case number within two years after he or she is released, we mark this individual with a recidivism flag. There are a total of 14 different lead charge categories assigned to each unique defendant-case instance. Figure 1 shows the yearly trend of the percent of recidivism depending on the covariates race and gender. We can observe that the traditionally considered as more privileged groups in the criminal justice system, such as white and female, have shown lower recidivism rates. The total recidivism cases take 32.9% of the final dataset.

d. Action Space

Our action space is created based on the distribution of the length of the confinement sentencing, where the 25th percentile is 0.5 years, the 50th percentile is 2 years, and the 75th percentile is 3 years. Thus, the action space is divided into 5 different bins, where instances of 0-year confinement sentencing are assigned to 0, < 0.5 years of sentencing assigned to 1, 0.5-2 year of sentencing assigned to 2, 2-3 year of sentencing assigned to 3, and sentencing years greater than 3 assigned to 4. The maximum sentencing is 12 years and the minimum sentencing is 0.

Figure 1. The percentage of recidivism based on different covariates



IV. Experiments and Results

a. Experiment Setup

Using the group fairness definition of the RobinHood algorithm, we measure whether the judicial system in the late 20th century (1989~1999) favored one group of defendants over the other. The first set of experiments choose 'Race' as a group criterion (White and Black), and the second set measures the disparity between 'Sex' (Female and Male).

For both sets, we carry out two types of experiments with different goals: 1) compare RobinHood's performance to that of Naive and Offset Tree using all 35 covariates, and 2) analyze the varying degrees of fairness influenced by the courtroom participants such as judges and assistant district attorneys (ADAs) using RobinHood. Among many parties involved, we choose 'trial judge', 'screen ADA', and 'trial ADA' to examine, following the suggestions of our mentors. To measure the effects, we use the covariate set (named 'NONE') that only has the defendant's data and crime-related features as our starting point. Then we add the information related to the three parties independently on top of the 'NONE' set (named 'JUDGE', 'SCREEN_ADA', and 'TRIAL_ADA', respectively).

When running each experiment, we keep all configurations the same as the original paper, except the number of trials ('n_trials'). We choose to proceed with 10 trials for each experiment, considering the runtime and the multitude of experiments.

b. Evaluation Metrics

Two main metrics used to compare experiments are 1) the expected reward, and 2) the difference in the false-positive rate between groups (denoted 'bqf' in the code) on the test set, which serves as our fairness metrics. The reward is the importance-weighted expected reward given by a potential policy. Since we receive a reward of -1 when recidivism is observed and 1 otherwise, a higher expected reward means that the sentencing policy is less likely to provoke recidivism. For the second metric, the closer the value is to zero, the fairer the policy is. In summary, we need to opt for a policy with the highest expected reward if our objective is to reduce the probabilities of recidivism; on the other hand, if exercising fair sentencing is the ultimate goal, we choose a policy with the smallest false-positive gap (fairness metrics) though it might entail sacrifice in rewards in some cases.

c. Results: Race as Treatment

RobinHood versus Others (Table 1) When race acts as our treatment, the RobinHood algorithm generates the fairest sentencing policy while Naive performs the best in terms of rewards. However, the difference in rewards between the two is very marginal (Δ= 0.06). Therefore, if we were to choose a single algorithm, RobinHood is the optimal choice that produces fair sentencing with a negligible trade-off in rewards. It is interesting to note that, while Offset Tree is the most computationally efficient algorithm and is not constrained by our fairness conditions, Offset Tree generates the lowest reward.

Table 1. Different Models' Performance for White versus Black

Model	Expected Reward		Fairness Metrics		Average
	Mean	SD	Mean	SD	Train Time (s)
RobinHood	0.5331	0.1583	0.0028	0.0052	25726.1556
Naive	0.5932	0.1152	0.0053	0.0053	21251.6219
Offset Tree	0.3735	0.0385	0.0418	0.0437	1.8273

Effects of Judge and ADA (Table 2) Our experiments show that recidivism is least likely (inferred by the highest reward) when we consider only defendant/crime-related information for sentencing. Yet, this should be taken with a grain of salt as the standard deviation is relatively high compared to others. Alternatively, if we wish to minimize the racial bias in the prosecutorial decisions, it is desirable to incorporate information about trial ADA (sex, race, party, age). To strengthen this argument, we may need to run more trials and consult with the field experts as our experiments are only based on 10 trials.

Table 2. RobinHood's Performance on Different Covariate Sets for White versus Black

Covariate Set	Expected Reward		Fairness Metrics		Average
	Mean	SD	Mean	SD	Train Time (s)
None	1.8410	2.4533	0.0051	0.0099	16965.5842
Judge	0.5718	0.2168	0.0054	0.0055	35453.9813
Trial ADA	0.7121	0.1111	0.0014	0.0032	18235.3556
Screen ADA	0.6751	0.1191	0.0041	0.0037	27005.0154

d. Results: Gender as Treatment

RobinHood versus Others (Table 3) The Gender experiments show an opposite trend: when the maximum reward is the goal, RobinHood performs the best while the Naive algorithm gives the fairest policy. If we allow that the difference in rewards is trivial, then the Naive algorithm is the optimal performer in terms of both rewards and fairness. However, this is by no means evidence against RobinHood; the original RobinHood paper explains that the Naive algorithm acts almost identically to RobinHood so long as enough data is given. That is, "RobinHood is a variant of Naive that includes a mechanism for determining when there is sufficient data to trust the conclusions drawn from the available data." [2]. One salient conclusion we can draw is that Offset Tree generates the lowest rewards and the worst fairness level in all experiments.

Table 3. Different Models' Performance for Male versus Female

Model	Expected Reward		Fairness Metrics		Average Train Time (s)
	Mean	SD	Mean	SD	
RobinHood	0.6141	0.1113	0.0071	0.0069	37239.3144
Naive	0.6123	0.1489	0.0025	0.0037	37020.2797
Offset Tree	0.4689	0.0796	0.0541	0.0346	1.5221

Effects of Judge and ADA (Table 4) Based on our results, the optimal sentencing can be made in terms of both rewards and fairness when the characteristics of the judge are considered. (It is also true that the reward prediction should be considered with caution as the standard deviation is significantly higher than others). There is interesting research from social psychology, which claims that judges may be just as biased or "even more biased" than the general public in deciding court cases where traditional gender roles are challenged [4]. This coincides with our finding that knowing about the judge, probably the sex of the judge, would help the decision-making to be fairer toward the opposite sex.

Table 4. RobinHood's Performance on Different Covariate Sets for Male versus Female

Covariate Set	Expected Reward		Fairness Metrics		Average Train Time (s)
	Mean	SD	Mean	SD	
None	0.5873	0.1940	0.0050	0.0121	13079.1747
Judge	2.5341	5.3976	0.0022	0.0026	16303.8111
Trial ADA	0.7033	0.1149	0.0110	0.0136	38709.7369
Screen ADA	0.6793	0.1019	0.0094	0.0108	13571.7738

V. Conclusions and Future Improvements

In this project, we successfully applied RobinHood to the NODA dataset. The results show that, when ensuring fairness between white and black defendants, RobinHood can achieve the fairest sentencing policy while sacrificing little on rewards. Additionally, considering the trial ADAs associated with each case and defendant helps establish fairness between races. However, the opposite trend is observed when we try to ensure fairness between male and female defendants. In this set of experiments, Naive appears to be the fairest algorithm while returning relatively high rewards. For gender, the most helpful additional covariates are those that are related to the judge in charge of each case and defendant.

Some of the challenges we have encountered include understanding criminal justice practices and the complicated RobinHood algorithm, cleaning and preparing a large NODA dataset, and training the model for an extensive amount of time. Since we lack background and knowledge in the U.S. criminal justice system, the team faced a lot of unclear definitions of legal terms and the complex relationship between practices like confinement, probation, and condition. In addition, the RobinHood algorithm is poorly documented with little explanation and comments about the code implementation and outputs. However, the biggest challenge comes from the complicated process of cleaning and preparing the NODA dataset. We started the project with more than 300 covariates and narrowed them down to 35 after extensive research and discussion with our advisors on each covariate.

For the next steps in this project, we would like to experiment with other fairness constraints. Currently, we borrowed the fairness constraints that the original RobinHood paper designed for the ProPublica dataset. However, RobinHood allows users to define their own fairness constraints, and, in the future, we can experiment with other fairness constraints like CATE or constraints other than the difference between false-positive rates between the two groups. Additionally, we are interested in comparing RobinHood with other baselines like POEM and exploring actions other than confinement sentences such as bond information.

VI. Acknowledgments

We would like to acknowledge Daniel Li Chen and Ryan Carroll from the World Bank Group for giving all the helpful suggestions and guidance throughout the project, as well as providing the NODA dataset.

References

[1] Gendreau, P., Goggin, C., and Cullen, F. T. (1999). "The Effects of Prison Sentences on Recidivism". Public Works and Government Services Canada.
[2] Metevier, B., Giguere, S., Brockman, S., Kobren, A., Brun, Y., Brunskill, E., and Thomas, P. S. (2019). "Offline Contextual Bandits with High Probability Fairness Guarantees". Neural Information Processing Systems 2019.
[3] Beygelzimer, A. and Langford, J. (2016). "The Offset Tree for Learning with Partial Labels". Association for Computing Machinery 2009.
[4] Miller, A. L. (2018). "Expertise Fails to Attenuate Gendered Biases in Judicial Decision-Making". Social Psychological and Personality Science.

Appendix A. Replication Results on ProPublica

Model	Expected Reward		Fairness Metrics		Average
	Mean	SD	Mean	SD	Train Time (s)
RobinHood	0.6253	0.1811	0.0033	0.0216	220.7806
Naive	0.6225	0.1824	0.0041	0.0196	210.9959
OffsetTree	0.6946	0.2201	0.0703	0.0823	0.0087

* 50 trials × 11 different training set sizes = 550 experiments per model

Appendix B. Pseudo Code of RobinHood Algorithm

Algorithm 1 RobinHood $(D, \Delta = \{\delta_i\}_{i=1}^{k}, \hat{Z}\{\hat{z}_j^i\}_{j=1}^{d}, \mathcal{E} = \{E_i\}_{i=1}^{k})$

$D_c, D_s = \texttt{partition}(D)$
$\theta_c = \arg\max_{\theta \in \Theta} \texttt{CandidateUtility}(\theta, D_c, \Delta, \hat{Z}, \mathcal{E})$ ▷ Candidate Selection
$[U_1, ..., U_k] = \texttt{ComputeUpperBounds}(\theta_c, D_s, \Delta, \mathcal{E}, \texttt{inflateBounds=False})$ ▷ Safety Test
if $U_i \leq 0$ for all $i \in \{1, ..., k\}$ **then return** θ_c **else return** `NSF`

Algorithm 2 CandidateUtility$(\theta, D_c, \Delta, \hat{Z}, \mathcal{E})$

$[\hat{U}_1, ..., \hat{U}_k] = \texttt{ComputeUpperBounds}(\theta, D_c, \Delta, \hat{Z}, \mathcal{E}, \texttt{inflateBounds=True})$
$r_{\min} = \min_{\theta' \in \Theta} \hat{r}(\theta', D_c)$
if $\hat{U}_i \leq -\xi$ for all $i \in \{1, ..., k\}$ **then return** $\hat{r}(\theta, D_c)$ **else return** $r_{\min} - \sum_{i=1}^{k} \max\{0, \hat{U}_i\}$

Algorithm 3 ComputeUpperBounds$(\theta, D, \Delta, \hat{Z}, \mathcal{E}, \texttt{inflateBounds})$

$\texttt{out} = [\,]$
for $i = 1, ..., k$ **do**
    $\hat{Z}_i = \{\hat{z}_j^i\}_{j=1}^{d_i} \subseteq \hat{Z}$
    $L_i, U_i = \texttt{Bound}(E_i, \theta, D, \delta_i / d_i, \hat{Z}_i, \texttt{inflateBounds})$
    $\texttt{out.append}(U_i)$
return `out`

Appendix C. Covariate List

Defendant and Crime Details :
DISP_CODE_SENT
SENTENCE_TYPE
HABITUAL_OFFENDER_FLAG_SENT
SENTENCE_LOCATION
CUSTODY_CODE
CRIMINAL_FLAG
SEX_DFDN
RACE_DFDN
JUVENILE_INVOLVED_FLAG
BOND_MADE_FLAG
CHARGE_CAT
BOND_TYPE_CODE
FINE_AMOUNT
NBR_OF_DFDN
AGE_DFDN
AGE_DFDN_ISNA
BOND_MADE_AMOUNT
BOND_SET_AMOUNT
CHARGE_CLASS
CHARGE_CLASS_ISNA

Judge Details:
SEX_JUDGE
RACE_JUDGE
PARTY_JUDGE
AGE_JUDGE
AGE_JUDGE_ISNA

Trial ADA Details:
RACE_TRIAL_ADA
SEX_TRIAL_ADA
PARTY_TRIAL_ADA
AGE_TRIAL_ADA
AGE_TRIAL_ADA_ISNA

Screen ADA Details:
RACE_SCREEN_ADA
SEX_SCREEN_ADA
PARTY_SCREEN_ADA
AGE_SCREEN_ADA
AGE_SCREEN_ADA_ISNA