Introduction to Fairness

Ygor Canalli

PESC - UFRJ

Table of contents

Problem characterization

Introducion

- Discrimination-Aware classification was first introduced[1,2] to avoid unwanted dependencies between the attributes.
- Given a set of *sensitive attributes*, such as race, color, religion, sex, age, pregnancy
- The goal is to learn statistical models avoiding bias, discrimination or prejudice with respect to the sensitive attributes.
- Due to historical issues, human data may have, intentionally or not, harmful bias against some groups.

---

[1] Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. "Discrimination-aware data mining". In: *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 08* (2008), p. 560.

[2] Faisal Kamiran and Toon Calders. "Classifying without discriminating". In: *2009 2nd International Conference on Computer, Control and Communication, IC4 2009* (2009).

Causes of unfairness

Causes of unfairness

- There are, at least, three causes of unfairness[3]
- Prejudice
  - Direct prejudice
  - Indirect prejudice
- Underestimation
- Negative legacy

[3]Toshihiro Kamishima et al. “Fairness-aware classifier with prejudice remover regularizer”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7524 LNAI.PART 2 (2012), pp. 35–50.

Fairness criterias

Fairness criterias

- Anti-classification
    - Protected attributes and their proxies are not explicitly used to make decisions
- Classification parity
    - Measures of predictive performance (e.g., false positive and false negative rates) are equal across groups defined by the protected attributes
- Calibration
    - Outcomes are independent of protected attributes af- ter controlling for estimated risk

Fairness criterias

- Threshold rules[4]
  - Treat similarly risky people similarly
- Rich subgroup fairness[5]
  - Fairness constraint (say, equalizing false positive rates across protected groups), hold over an exponentially or infinitely large subgroups

[4] Sam Corbett-Davies et al. *The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning **. Tech. rep. 2018.
[5] Michael Kearns et al. *An Empirical Study of Rich Subgroup Fairness for Machine Learning*. Tech. rep. 2018.

Redlining effect

Redlining effect

- Simply removing sensitive attributes from training data is not enough to solve this problem
- The statistical model could indirectly learn bias through related features, phenomena known as *redlining effect*[6].
- For example, ethnicity may be strongly related to zip code.

[6] Gregory D. Squires. "Racial Profiling, Insurance Style: Insurance Redlining and the Uneven Development of Metropolitan Areas". In: *Journal of Urban Affairs* 25.4 (2003), pp. 391–410.

Approaches

α-protector

- Algorithm based on association and classification rules[7]
- Manually define a α-protector threshold for increasing confidence
- Direct and indirect discrimination versions

[7] Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. "Discrimination-aware data mining". In: *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 08* (2008), p. 560.

Data massaging

- *Massaging* the data to remove discrimination[8].
- Instances are ranked by learning a model trained without sensitive attributes
- Select candidates for *promotion* and *demotion*
- Invert labels to avoid discrimination

[8] Faisal Kamiran and Toon Calders. "Classifying without discriminating". In: *2009 2nd International Conference on Computer, Control and Communication, IC4 2009* (2009).

Bayesian models

- Proposes three models based on Naive Bayes[9]
  1. *Post-processing* phase that modify the probability of the decision being positive by changing the probabilities in the model
  2. Train *one model by value* of every sensitive attribute and *balance* them
  3. Add a *latent variable* in the Bayesian model that represents an unbiased, discrimination-free label and optimize the model parameters for likelihood using expectation maximization

[9] Toon Calders and Sicco Verwer. "Three naive Bayes approaches for discrimination-free classification". In: *Data Mining and Knowledge Discovery* 21.2 (2010), pp. 277–292.

Decision tree

- Proposes two models based on Decision Tree[10]
    1. Evaluates the *discrimination caused by each split*, not only its contribution to accuracy
    2. *Leaf relabeling* to lower discrimination

[10] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. "Discrimination aware decision tree learning". In: *Proceedings - IEEE International Conference on Data Mining, ICDM* (2010), pp. 869–874.

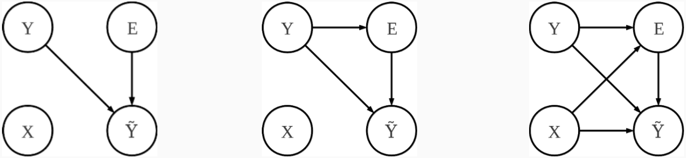
Heuristics for rich subgroups

- Two-player zero-sum game[11]
    1. Learner (the primal player)
    2. Auditor (the dual player)

[11] Michael Kearns et al. “Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness”. In: (Nov. 2017).

Noise Taxonomy

Noise Taxonomy



(a) NCAR (b) NAR (c) NNAR

Figure 1: Noisy Models

Noisy Completely at Random The occurrence of an error E is independent of the other random variables, including the true class itself. So called uniform label noise.

Noisy at Random The probability of occurrence of na error is dependent on true lable. Different classes may have different label error rates.

Noisy Not at Random The occurrence of an error may depend not only on true label, but even on features.

Proposal

Proposal

- Handle fairness as a *waekly supervised learning problem*
- Apply Loss Factorization[12]
- Given a *transition matrix*, use forward and backward to correct prediction[13]
- Given a transition matrix T we can define forward correction loss as

$$\ell^{\rightarrow}(p(y|x)) = \ell(T^{T} p(y|x))$$

[12] Giorgio Patrini et al. *Loss Factorization, Weakly Supervised Learning and Label Noise Robustness*. Tech. rep. 2016. URL: `http://proceedings.mlr.press/v48/patrini16.pdf`.

[13] Filipe Braida do Carmo. "Considerando o ruído no aprendizado de modelos preditivos robustos para a filtragem colaborativa". PhD thesis. Universidade Federal do Rio de Janeiro.

Adult Income Dataset characterization

Adult Income Dataset characterization

Target

Size

Total	48842
Train	32561 (66.66%)
Test	16281 (33.33%)

Positive > 50k

train	7841 (24.08%)
test	3846 (23.62%)

Negative $<= 50$k

train	24720 (75.91%)
test	12435 (76.37%)

Features

age	continuous.
workclass	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
fnlwgt	continuous.

Adult Income Dataset characterization

education Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.

education-num continuous.

marital-status Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

occupation Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

relationship Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

Adult Income Dataset characterization

race White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.

sex Female, Male.

capital-gain continuous.

capital-loss continuous.

hours-per-week continuous.

native-country United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

NAR Experiment on Adult Income

NAR Experiment on Adult Income

Given

False positive rate: fp

False negative rate: fn

We define transition matrix T as

$$T = \begin{bmatrix} 1 - fp & fp \\ fn & 1 - fn \end{bmatrix}$$

Also, forward correction as

```
def forward_categorical_crossentropy(T):
    def loss(y_true,y_pred):
        pred = dot(transpose(T), y_pred)
        return categorical_crossentropy(y_true, pred)
    return loss
```

NAR Experiment on Adult Income

Pre-processing
- One-hot-encoding
- Normalization (Min-Max Scaler)

Randomness
- Fixed seed

Network architecture
- $108 \to 128 \to ReLU \to 2 \to softmax$

```
traning_epochs = 6
model = keras.Sequential([
        keras.layers.Flatten(input_shape=(108,)),
        keras.layers.Dense(128, activation=tf.nn.relu),
        keras.layers.Dense(2, activation=tf.nn.softmax)
    ])
model.compile(optimizer='adam',
              loss=categorical_crossentropy,
              metrics=['accuracy'])
```

NAR Experiment on Adult Income

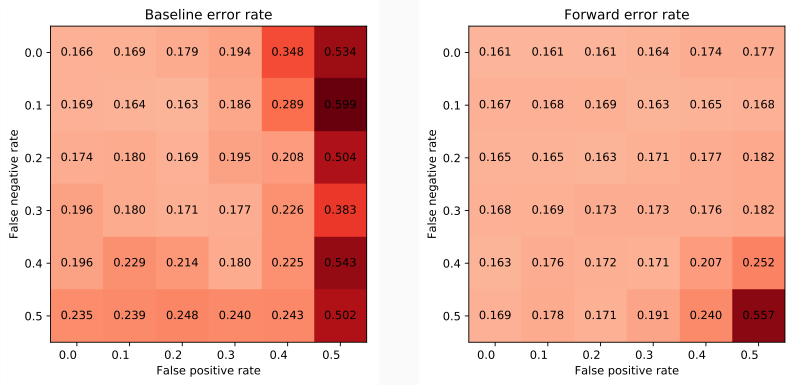
Confusion matrix without noise

		Predicted outcome		
		<= 50k	> 50k	total
Actual value	<= 50k	69.53% True neg	6.84% False pos	76.37%
	> 50k	8.27% False neg	15.34% True pos	23.62%
	total	77.80%	22.19%	

NAR Experiment on Adult Income

```python
false_positive_rates = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5]
false_negative_rates = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5]

for fp in false_negative_rates:
    for fn in false_negative_rates:
        T = np.array([[1-fp,fp],[fn,1-fn]])

        polluted_y_train = pollute(y_train, T)
        forward_loss = forward_categorical_crossentropy(T)

        baseline_acc = evaluate(X_train, X_test,
                    polluted_y_train, y_test,
                    loss_function=categorical_crossentropy)

        forward_acc = evaluate(X_train, X_test,
                    polluted_y_train, y_test,
                    loss_function=forward_loss)
```

NAR Experiment on Adult Income



(a) Cross-entropy loss (baseline) **(b)** Forward loss

Figure 2: Error rate by pollution level

NAR Experiment on Adult Income

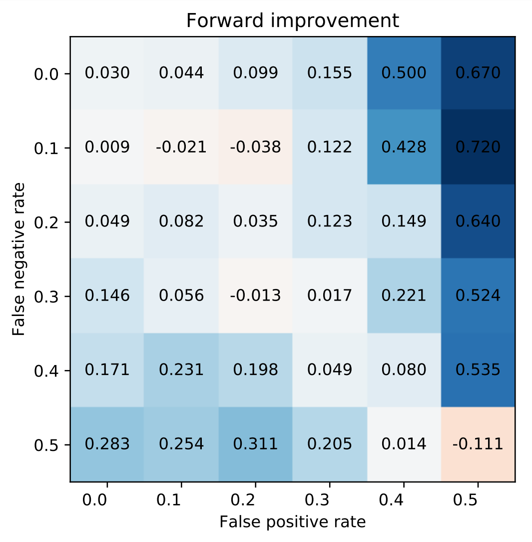


Figure 3: Improvement: $(baseline - forward)/baseline$

NNAR Experiment on Adult Income

NNAR Experiment on Adult Income

Confusion matrix on test without noise

		Predicted outcome		
		$<= 50k$	$> 50k$	total
Actual value	$<= 50k$	69.53% True neg	6.84% False pos	76.37%
	$> 50k$	8.27% False neg	15.34% True pos	23.62%
	total	77.80%	22.19%	

NNAR Experiment on Adult Income

Confusion matrix on test [male only] without noise

Predicted outcome

		$<= 50$k	> 50k	total
Actual value	$<= 50$k	60.81% True neg	9.2% False pos	70.01%
	> 50k	9.78% False neg	20.19% True pos	29.98%
	total	70.59%	29.40%	

NNAR Experiment on Adult Income

Confusion matrix on test [female only] without noise

Predicted outcome

Actual value		$<= 50$k	> 50k	total
	$<= 50$k	87.01% True neg	2.1% False pos	89.11%
	> 50k	5.23% False neg	5.64% True pos	10.88%
	total	92.25%	7.74%	

NNAR Experiment on Adult Income

Given

Male false positive rate: fp_{male}

Nale false negative rate: fn_{male}

Female false positive rate: fp_{female}

Female false negative rate: fn_{female}

We define transition matrix as

$$T_{male} = \begin{bmatrix} 1 - fp_{male} & fp_{male} \\ fn_{male} & 1 - fn_{male} \end{bmatrix},$$

$$T_{female} = \begin{bmatrix} 1 - fp_{female} & fp_{female} \\ fn_{female} & 1 - fn_{female} \end{bmatrix}$$

NNAR Experiment on Adult Income

We define, male and female forward correction as

```
T_male = np.array([[1-fp_male, fp_male  ],
                   [ fn_male  , 1-fn_male]])

T_female = np.array([[1-fp_female, fp_female  ],
                     [ fn_female  , 1-fn_female]])

male_loss = forward_categorical_crossentropy(T_male)
female_loss = forward_categorical_crossentropy(T_female)
```

And evaluate on folowing error rates

```
fp_male_rates   = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5]
fn_male_rates   = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5]
fp_female_rates = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5]
fn_female_rates = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5]
```

NNAR Experiment on Adult Income

Alternating training

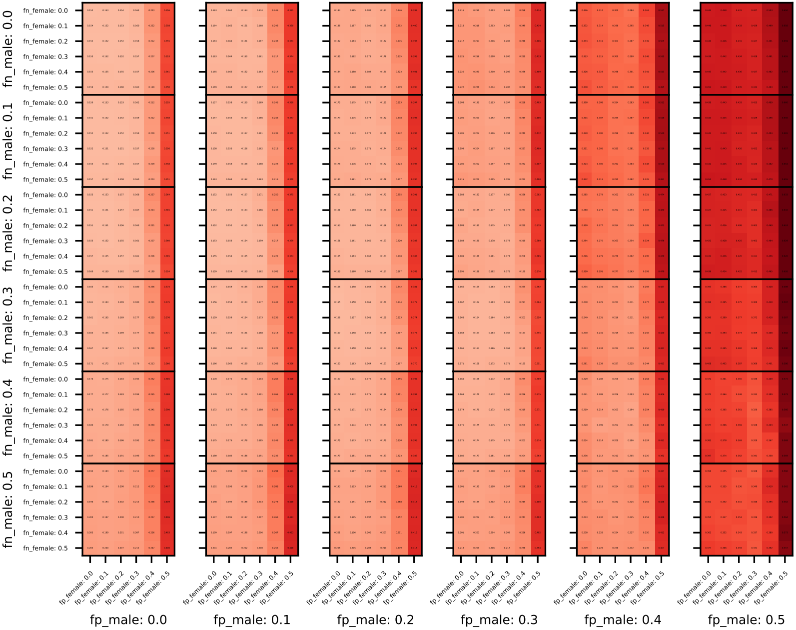
```
for i in range(6):

    model.compile(optimizer='adam',
                  loss=female_loss,
                  metrics=['accuracy'])

    model.fit(X_train_female, polluted_female_labels,
              epochs=1)

    model.compile(optimizer='adam',
                  loss=male_loss,
                  metrics=['accuracy'])

    model.fit(X_train_male, polluted_male_labels,
              epochs=1)

loss, acc = model.evaluate(X_test, y_test,)

```

NNAR Experiment on Adult Income

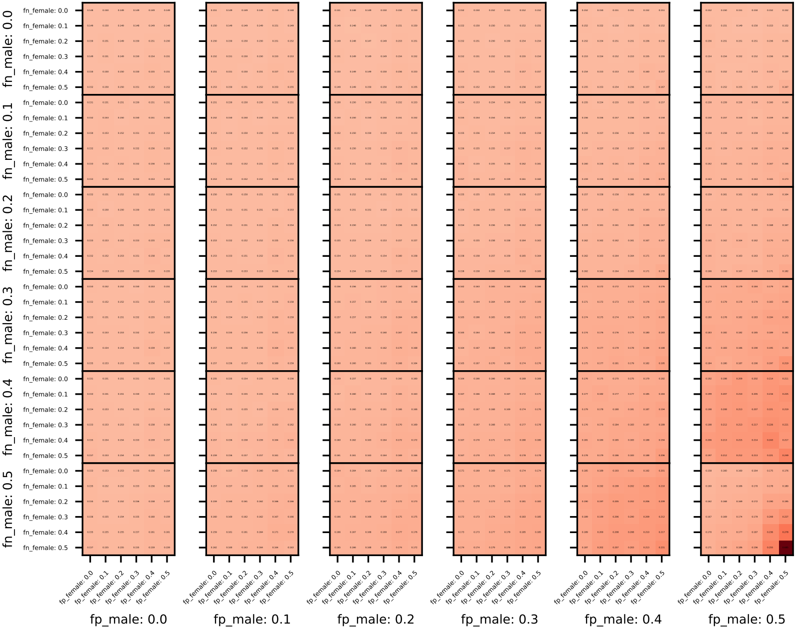
Two step training

```
model.compile(optimizer='adam',
              loss=forward_female_loss,
              metrics=['accuracy'])

model.fit(X_train_female, polluted_female_labels,
              epochs=6)

model.compile(optimizer='adam',
              loss=forward_male_loss,
              metrics=['accuracy'])

model.fit(X_train_male, polluted_male_labels,
              epochs=6)

loss, acc = model.evaluate(X_test, y_test)
```

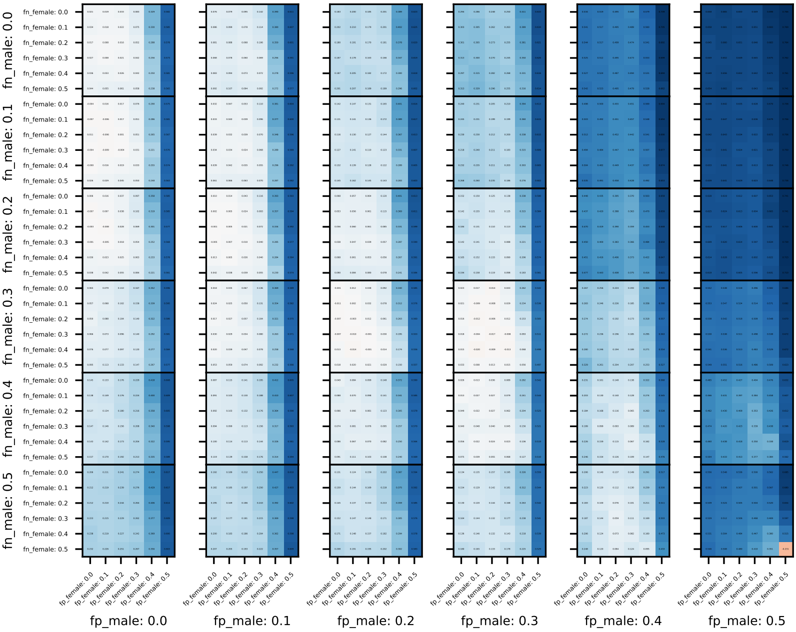
NNAR Experiment on Adult Income: Baseline error rate



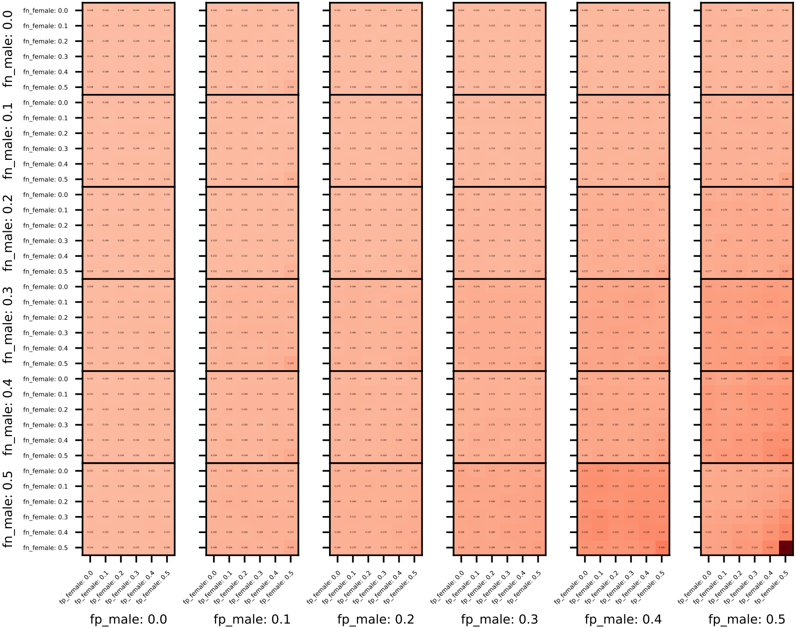
NNAR Experiment on Adult Income: Alternating forward error rate



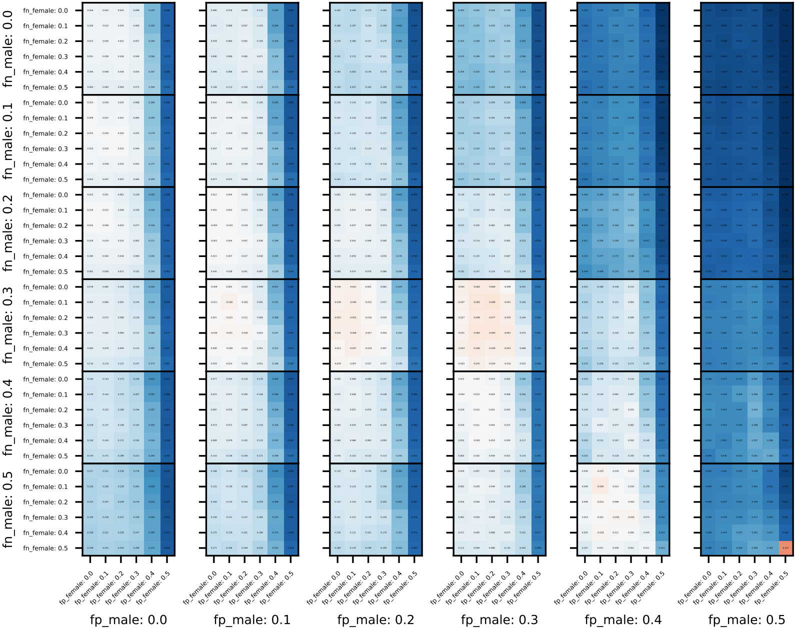
NNAR Experiment on Adult Income: Alternating forward improvement



NNAR Experiment on Adult Income: Two step forward error rate



NNAR Experiment on Adult Income: Two step forward improvement



NNAR Experiment on Adult Income

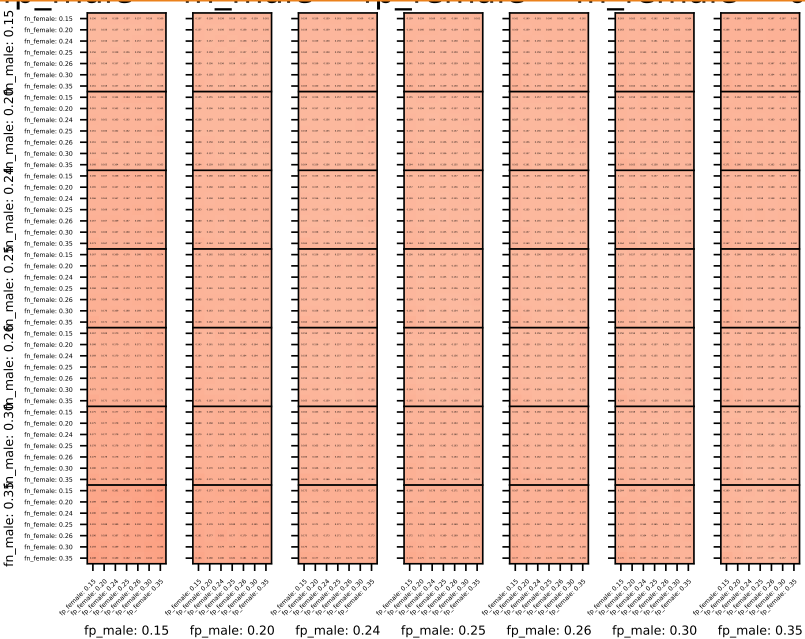
To measure method sensibility to bad T estimation, we will fix pollution matrix in whole train set to

$$\begin{bmatrix} 1 - 0.25 & 0.25 \\ -0.25 & 1 - 0.25 \end{bmatrix},$$

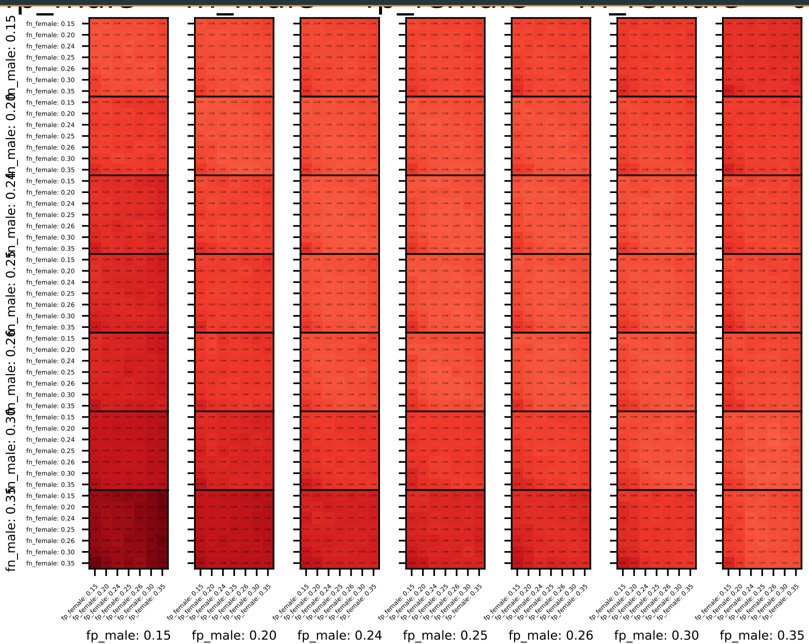
and evaluate varying each error rate on ± 0.01, ± 0.05 and ± 0.1, giving the folowing error rates:

```

fp_male_rates   = [0.15, 0.20, 0.24, 0.25, 0.26, 0.30, 0.35]
fn_male_rates   = [0.15, 0.20, 0.24, 0.25, 0.26, 0.30, 0.35]
fp_female_rates = [0.15, 0.20, 0.24, 0.25, 0.26, 0.30, 0.35]
fn_female_rates = [0.15, 0.20, 0.24, 0.25, 0.26, 0.30, 0.35]

```

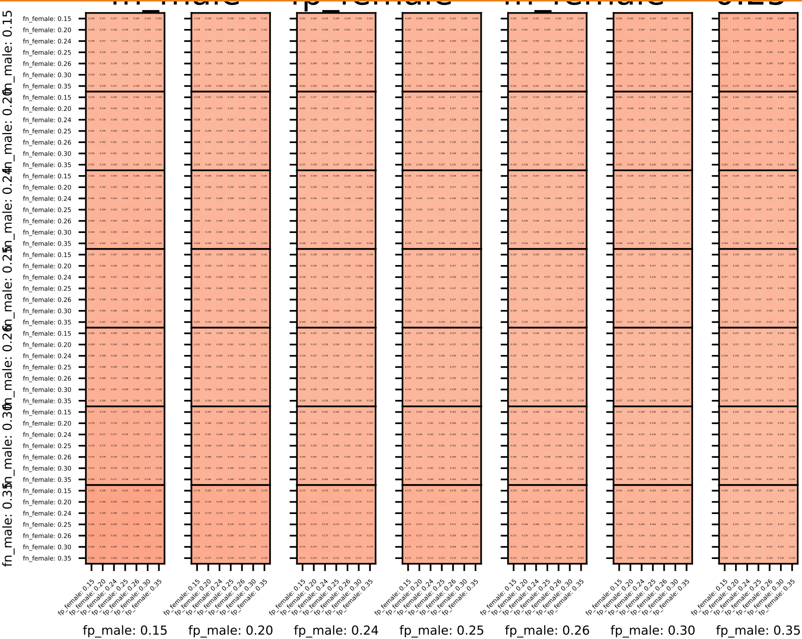
Alternating forward sensibility



Alternating forward sensibility (reduced color scale)



Two step forward sensibility



Two step forward sensibility (reduced color scale)