Assignment 3: Probabilistic Definitions of Fairness and Bayesian Networks

Machine Learning

Fall 2021

Learning Objectives

- Learn about the connection between probabilistic criteria for algorithmic fairness and Bayesian Networks.
- Solidify your understanding of the Naïve Bayes algorithm by applying it to movie review sentiment classification.

1 *Reflecting on COMPAS and Algorithmic Fairness*

In assignment 1 of this module we discussed how Bayesian methods can be used to reason about algorithmic fairness. In class, we discussed fairness within the context of the COMPAS algorithm and the United States justice system.

Exercise 1 (15 minutes)

Please reflect on the conversation in class about the COMPAS algorithm, the US justice system, race in the US, and algorithms and fairness. You can reflect on these topics and/or on the conversation itself (meta). We look forward to hearing your thoughts and appreciate any ideas you have to improve these types of conversations in the future. We are all learning together.
Additionally, if you would like to share something anonymously, here is an anonymous survey.

2 *Measures of Bias*

In the last assignment, we looked at confusion matrices and different metrics that can be calculated for a single confusion matrix. Now, we will look at some metrics for comparing two confusion matrices (e.g., if we break our results up to look at the outcomes for two different groups).

As we saw in the COMPAS debate, confusion matrix metrics are often used to measure fairness and bias. Note that when we say algorithmic bias and fairness, we are referring to the phenomenon of models discriminating against protected or underprivileged groups (based on attributes such as race and gender), which often arises as a factor of bias that is integrated into the dataset used for training. This is common, because datasets often measure phenomena that are already subject to systemic bias and may include additional (usually unintended) bias introduced by the collectors and analysts of the data.

When detecting algorithmic bias, we typically identify a protected class (such as people of color or women). Many of the popular measures of bias compare the confusion matrix metrics we went over in the last section between the protected and privileged class, either by taking a straightforward difference, or by taking a ratio between the values of the two metrics. These are often referred to as parity of a given metric, for example False Positive Rate Parity, which measures the difference in False Positives of the two groups. Many of the most commonly used metrics have a second name, for example the Equal Opportunity Difference is the same as True Positive Rate Parity.

There are many measures of bias, but today we will be focusing on the five common metrics. By understanding these metrics, you will have the tools to understand other, similar metrics that you may want to use in the future. It may be helpful to refer to the loan example visualization from Google Research again to wrap your head around this.

- **Statistical Parity Difference** is computed as the difference of the rate of favorable outcomes received by the unprivileged group to the privileged group.
  - The ideal value of this metric is 0.
  - Fairness for this metric is between -0.1 and 0.1.
  - Calculated as the difference between predicted positive of the privileged and unprivileged groups.
  - A value of < 0 implies higher benefit for the privileged group and a value > 0 implies higher benefit for the unprivileged group.
- **Equal Opportunity Difference** is computed as the difference of true positive rates between the unprivileged and the privileged groups.
  - The ideal value of this metric is 0.
  - A value of < 0 implies higher benefit for the privileged group and a value > 0 implies higher benefit for the unprivileged group.
  - Fairness for this metric is between -0.1 and 0.1
  - Calculated as the difference between the privileged TPR and unprivileged TPR.
- **Average Odds Difference** is computed as the average difference of false positive rate and true positive rate between unprivileged and privileged groups.
  - The ideal value of this metric is 0.
  - A value of < 0 implies higher benefit for the privileged group and a value > 0 implies higher benefit for the unprivileged group.
  - Fairness for this metric is between -0.1 and 0.1
  - Calculated as the average of the difference between the privileged FPR and unprivileged FPR and the difference between the unprivileged TPR and privileged TPR
- **Disparate Impact** is computed as the ratio between rate of favorable outcomes for the unprivileged group to that of the privileged group.
  - The ideal value of this metric is 1.0.
  - A value < 1 implies higher benefit for the privileged group and a value >1 implies a higher benefit for the unprivileged group.
  - Fairness for this metric is between 0.8 and 1.2.
  - Calculated as unprivileged predicted positive/privileged predicted positive.
- **Theil Index** is computed as the generalized entropy of benefit for all individuals in the dataset, with alpha = 1. It measures the inequality in benefit allocation for individuals. This is a measure of non-randomness or redundancy.
  - A value of 0 implies perfect fairness.
  - Fairness is indicated by lower scores, higher scores are problematic.
  - We will not focus on this metric, but Wikipedia has a decent article about it.

Sometimes these metrics conflict, so we need to choose carefully based on our application which ones to minimize.

Exercise 2 (60 Minutes)

(a) Calculate the Statistical Parity Difference, Equal Opportunity Difference, Average Odds Difference, and Disparate Impact of the confusion matrices below. What do you notice? Is this model fair?

(b) The Data Science and Public Policy group at Carnegie Mellon University created a decision tree with proposed guidance about which metrics of fairness might be desirable in different situations. Take a look at their decision tree. Try to understand why each decision leads to the metrics that it does.

The Impossibility Theorem of Fairness

As we explored in the previous problems, we often cannot satisfy every fairness metric at the same time. This leads to an interesting ethical discussion about when it is appropriate to use a given metric, and when it is okay to accept a failing score on a given fairness metric. Above we show a decision tree for certain metrics that we took from Aequitas, a bias detection toolkit created by the University of Chicago Center for Data Science and Public Policy (now at Carnegie Mellon).

Choosing a fairness metric in a way that is ethical is a highly subjective task. Even though decision trees exist to help policymakers and model creators evaluate their models, this is an active area of research and discussion. One great resource for learning about up and coming developments in this field is the FAT conference, where researchers and technologists discuss and present research on Fairness, Accountability, and Transparency in ML. At the heart of these debates is the Impossibility Theorem of Fairness, which essentially says that there are many fairness metrics that are mutually exclusive in the vast majority of cases.

External Resource(s) 60 Minutes

This lecture on the tradeoffs of different fairness metrics talks about the societal impact of different fairness metrics, and explains the Impossibility Theorem of Fairness in more depth.

✓ Understanding Check

What is group fairness? What is individual fairness? Why are these two notions of fairness in conflict?

Bias Mitigation

Now that we have discussed different ways to detect bias, we want to briefly think about how we can reduce bias and algorithmic discrimination. There are various toolboxes that have been designed to help mitigate bias in algorithms and data, including Aequitas (from Carnegie Mellon - this also includes a more detailed decision tree) and the IBM AI Fairness 360 (which you can see a demo of here). There are a number of bias mitigation strategies, and each has several variations, but all bias mitigation strategies can be broken up into three categories.

Pre-processing

Pre-processing is the practice of modifying a dataset before training a model. Models learn bias from uneven representation in datasets, stemming from problems with sampling distribution, or from existing social phenomena that create uneven distribution of input situations and outcomes between demographic groups. The theory behind preprocessing is that if a dataset represents a fair distribution of outcomes between demographic groups, any model trained on that data will learn to classify in a fair manner. Preprocessing is often the preferred method of bias mitigation, since it allows for transparency in the bias mitigation process, and it does not change the model training process. It also allows for those who release datasets to ensure that models using those datasets are fair, regardless of who is making them. However, it requires foresight to use since it needs to be implemented before the model is trained, so often model creators opt for in-processing or post-processing.

In-processing

In-processing is the practice of considering notions of fairness while training a model. This is often done by integrating fairness metrics into the error or optimization function while training. In-processing is popular because it builds fairness directly into the model design, unlike pre-processing that only builds fairness into the training data.

Post-processing

Post processing is the practice of modifying an existing model's output to satisfy notions of fairness. Post-processing only focuses on group fairness, and it does not consider individual fairness. For this reason, post-processing is often less preferable that the other types of bias mitigation algorithms, which have to ability to consider both group fairness and individual fairness. However, post-processing is a good choice when a model is already trained and cannot be rebuilt.

3 Generative versus Discriminative Models (5 minutes)

So far, we have built up some machinery that allows us to work with probabilities. Next we're going to take this machinery and turn back towards machine learning. Specifically, we'll be looking at the classification problem and using probability theory to see it in a whole new light.

This short video provides a quick overview of generative versus discriminative models, which may be helpful prep for reading the next sections.

3.1 Discriminative Models: a Look Back at Logistic Regression (10 minute read)

Let's think back to the logistic regression model for binary classification that we learned about in module 1. Given an input point $\mathbf{x_i}$, the logistic regression algorithm applied a weight vector $\mathbf{w}$ to compute the probability that the corresponding output y_i was 1 via the formula $\sigma(\mathbf{w}^\top \mathbf{x_i}) = \frac{1}{1+e^{-\mathbf{w}^\top \mathbf{x_i}}}$ (recall that σ is known as the sigmoid function and serves to squash its input into a number between 0 and 1, which can serve as a valid probability). While we didn't quite have the vocabulary for it then, what we were really doing was computing a conditional probability. We can think of Y_i as a random variable that represents the output corresponding to the input $\mathbf{x_i}$ (in the case of binary classification Y_i is either 0 or 1). We can also think of the input as a random variable X_i (thinking of the input as a random variable will be helpful later in this section). Framed in this way the logistic regression algorithm computes the following conditional probability:

$$p(Y_i = 1 | X_i = \mathbf{x_i}) = \sigma(\mathbf{w}^\top \mathbf{x_i}) \ . \tag{1}$$

We defined a loss function to specify which weights were better or worse given a training set $(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)$. The details of how we did this are not important to the point we are trying to make now, so it'll suffice to say that learning in a logistic regression model meant tuning the conditional distribution of the outputs (the Y_i's) given the inputs ($\mathbf{x_i}$'s) to fit the training data the best. This type of model is what is known as a *discriminative model* (the Wikipedia article on discriminative models has more details if you are interested).

✔ Understanding Check

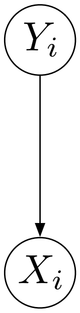
Intuitively, why does being able to accurately estimate $p(Y = 1|X = \mathbf{x})$ allow you to classify unseen test data?

3.2 Generative Models (10 minute read)

The approach outlined above is great, but it's not the only way to approach binary classification (and supervised learning in general). Since we are interested in predicting Y_i given some inputs $\mathbf{x_i}$, it of course makes sense, for example for a binary classification problem, to want to determine $p(Y_i = 1|\mathbf{x_i})$. However, instead of modeling that distribution directly, we can use Bayes' rule.

$$\begin{aligned} p(Y_i = 1|X_i = \mathbf{x_i}) &= \frac{p(X_i = \mathbf{x_i}|Y_i = 1)p(Y_i = 1)}{p(X_i = \mathbf{x_i})} \\ &= \frac{p(X_i = \mathbf{x_i}|Y_i = 1)p(Y_i = 1)}{p(X_i = \mathbf{x_i}|Y_i = 1)p(Y_i = 1) + p(X_i = \mathbf{x_i}|Y_i = 0)p(Y_i = 0)} \end{aligned} \tag{2}$$

These equations tell us is that if we have a model of the probability of the output being 1 *a priori*, $p(Y_i = 1)$, and a model of the inputs $\mathbf{x_i}$ given the output y_i, $p(X_i = \mathbf{x_i}|Y = y_i)$, then we can compute $p(Y_i = 1|X_i = \mathbf{x}_i)$. This amounts to adopting the perspective that the hidden output Y_i causes the input X_i. We call this sort of model a probabilistic generative model (PGM). The BN corresponding to this model is given below.



The natural question is *why?* Here are some potential advantages of using probabilistic generative models.

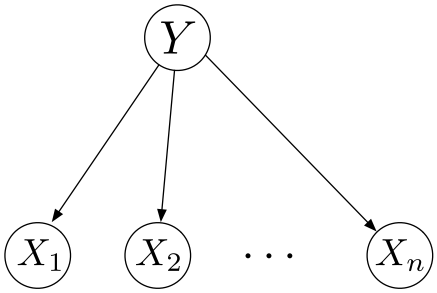
- Suppose you found out that $p(Y_i)$ changed for some reason (any thoughts on when this might happen? Post here on NB). Incorporating this change into a probabilistic graphical model would be very straightforward (just modify $p(Y_i = 1)$ in Equation 2).
- Suppose you found out that $p(X_i = \mathbf{x_i}|Y_i = y_i)$ changed for some reason. For example, if one of the elements of X_i represents a result obtained by running some sort of medical test, the sensitivity of that medical test might change (any other examples on when this might happen? Post here on NB.).
- Suppose that instead of classifying data (i.e., predicting Y_i), you instead wanted to generate samples $\mathbf{x_i}$ conditioned on a particular value of Y_i (e.g., you might want to synthesize samples of hand written digits based on training a probabilistic graphical model). This can be done naturally with a PGM. More modern versions of this idea are generative adversarial networks (GANs), which are behind such work as this person does not exist and better language models and their implications (the second link is the work of a former Oliner!).

✔ Understanding Check

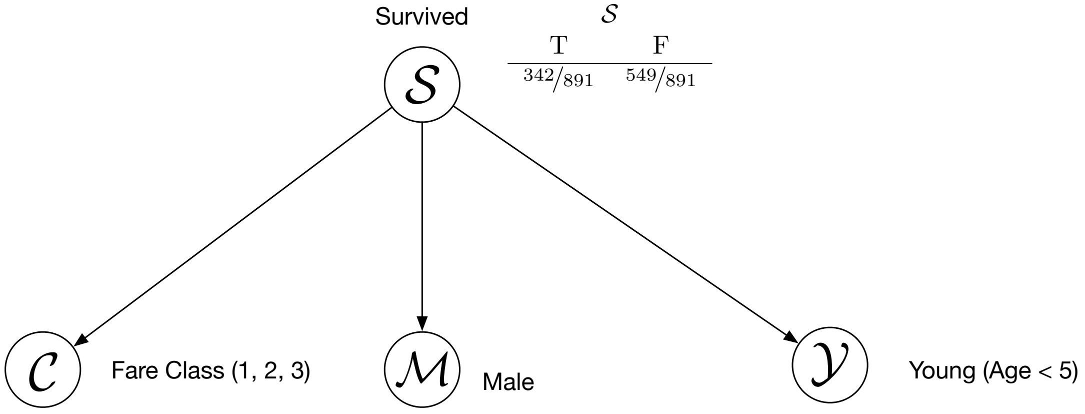
What are the probabilities needed to classify input data in a discriminative model? What are the probabilities needed to classify input data in a generative model? How does Bayes' rule connect these two models?

4 Meet the Naïve Bayes Algorithm

Now that we've learned the basic concept of a probabilistic graphical model, let's make things concrete and learn about a specific PGM. For our first PGM, we're going to learn about the Naïve Bayes algorithm (we'll be doing some work with Naïve Bayes in the next assignment as well, so you'll have some time to get accustomed to it). The reason it is called Naïve Bayes is that it assumes that all of the observed data $(X_1, X_2, \ldots, X_n)$ are conditionally independent given $\mathcal{Y}$. The BN for the Naïve Bayes algorithm is shown below.



As a motivating example, let's look back at the Titanic dataset from the last module. A potential BN for the Titanic dataset is shown below.



$\mathcal{S}$: T	$\mathcal{S}$: F
$342/891$	$549/891$

$\mathcal{S}$	$\mathcal{C}$: 1	2	3
T	$136/342$	$87/342$	$119/342$
F	$80/549$	$97/549$	$372/549$

$\mathcal{S}$	$\mathcal{M}$: T	F
T	$109/342$	$233/342$
F	$468/549$	$81/549$

$\mathcal{S}$	$\mathcal{Y}$: T	F
T	$27/342$	$315/342$
F	$13/549$	$536/549$

The probabilities in this BN were computed by looking at the training set and counting the appropriate passengers that fell into each category. For instance, to compute $p(\mathcal{Y}|\mathcal{S})$ since $p(\mathcal{Y}|\mathcal{S}) = \frac{p(\mathcal{Y},\mathcal{S})}{p(\mathcal{S})}$, we can approximate this probability by counting the number of passengers under 5 who survived and dividing by the total number who survived (note that there are some subtle and important modifications to this method of fitting these probabilities that we'll discuss in the next assignment). This process was repeated for each conditional probability. Since we assume that all of the features are conditionally independent given the output ($\mathcal{S}$ in this case), this process is done independently for each feature.

4.1 Inference

While the Naïve Bayes Algorithm might sound fancy, once we have the BN, all we need to do to run the algorithm is to use Bayes' rule. We'll let you work through this on your own via an exercise.

Exercise 3 (45 minutes)

(a) Using the BN shown above, what is the probability that a young, male in first class would survive the Titanic disaster? Hint: write this as a conditional probability and then use Bayes' rule. Hint 2: leverage the fact that $\mathcal{C}, \mathcal{Y}, \mathcal{M}$ are all conditionally independent of each other given $\mathcal{S}$.

You have just derived the Naïve Bayes inference rule!

(b) Naïve Bayes is often more conveniently expressed using odds ratios. Instead of computing $p(\mathcal{S}|\mathcal{Y}, \mathcal{C} = 1, \mathcal{M})$ let's compute the following.

$$
\begin{aligned}
\frac{p(\mathcal{S}|\mathcal{Y}, \mathcal{C} = 1, \mathcal{M})}{p(\neg\mathcal{S}|\mathcal{Y}, \mathcal{C} = 1, \mathcal{M})} &= \frac{\frac{p(\mathcal{Y}, \mathcal{C}=1, \mathcal{M}|\mathcal{S})p(\mathcal{S})}{p(\mathcal{Y}, \mathcal{C}=1, \mathcal{M})}}{\frac{p(\mathcal{Y}, \mathcal{C}=1, \mathcal{M}|\neg\mathcal{S})p(\neg\mathcal{S})}{p(\mathcal{Y}, \mathcal{C}=1, \mathcal{M})}} \\
&= \frac{p(\mathcal{Y}, \mathcal{C} = 1, \mathcal{M}|\mathcal{S})p(\mathcal{S})}{p(\mathcal{Y}, \mathcal{C} = 1, \mathcal{M}|\neg\mathcal{S})p(\neg\mathcal{S})} \\
&= \frac{p(\mathcal{Y}|\mathcal{S})p(\mathcal{C} = 1|\mathcal{S})p(\mathcal{M}|\mathcal{S})p(\mathcal{S})}{p(\mathcal{Y}|\neg\mathcal{S})p(\mathcal{C} = 1|\neg\mathcal{S})p(\mathcal{M}|\neg\mathcal{S})p(\neg\mathcal{S})}
\end{aligned}
$$

What must be true about this odds ratio in order to predict that the passenger survived?