Selection bias is a known issue in data science, but the depth of which is not fully appreciated. As is the case this particular problem, there is no simple solution to handle the bias. The bias makes a data set far from an infallible neutral source from which we can learning. To blindly run analysis on this data risks infecting the model with the same bias in the initial data and thereby creating a poor model. In this post, I hope to explain why selection bias is so challenging particularly in application data such as when applying for a financial loan.

Selection bias

The ancient Greek poet Diagoras of Melos was a famed aethist. A friend, in an effort to convince him of the existence of gods, said, "“You think the gods have no care for man? Why, you can see from all these votive pictures here how many people have escaped the fury of storms at sea by praying to the gods who have brought them safe to harbor.” tries to convince him of the existence of the gods by pointing to the paintings of men saved from storms on the sea through prayer. Diagoras countered “Yes, indeed, but where are the pictures of all those who suffered shipwreck and perished in the waves?”[^Hecht]

A selection bias occurs whenever the data are not representative of the full underlying distribution. If you want to create a sample of all people who pray, but only include those who commission paintings of themselves surviving storm, then it is not covering the full distribution by missing the those who drowned. Just as Diagoras was not convinced by the paintings, so should we be skeptical by conclusions drawn from data with a selection bias.

Once you start looking for selection bias, it is easy to spot. Right now in the U.S. we are inundated with political polls in the run-up to the presidential election. These polls try to be representative of U.S. voters, but struggle with selection bias. Pollsters put in a lot of effort to minimize this and are often open about their methodology.[^Rasmussen] Occasionally, though, the effort to reduce bias can cause problems with small sample sizes. In one notable poll, the New York Times discovered a large weight was placed on a single voter that fell into small, hard to sample demographics. [^NYTimes]

Handling this bias is a huge part of the value added by fiverthirtyeight.com and the Upshot at the New York Times. Instead of working to reduce the polling bias, they model how the selection bias in different polls is likely correlated. For example, the selection bias in polls in Ohio are likely to be strongly correlated to the selection bias in Michigan, but less correlated in North Carolina because of the different demographics. So if Donald Trump outperforms the polls and wins in Ohio, he is more likely to also outperform in Michigan than in North Carolina.[^fivethirtyeight]

Data from applications, whether for jobs, colleges, or loans, almost always have a huge selection bias. In all applications, it is always much easier to obtain data from the people that were accepted. A university that wants to assess their undergraduate admissions could run a test to see how well their selection process correlates with graduation rate or GPA of the admitted students. But the same university could never run a test to see how well a student whom they rejected would have performed. The same core problem exists in financial data.

The only way a university could fully assess their admissions would be to randomly admit some students for whom the admission process says reject. This exploration of possible candidates that are being missed is expensive to the point of absurdity. In general, anytime there is a selection bias in which only the result of positive examples are observed and the cost of exploration is high, it will be impossible to get unbiased data.

Selection Bias in Financial Data

I recently was asked to create a predictive model for financial loans. The data I was given for training consisted of about 100,000 U.K. loan applications. The data included whether each loan was either approved or denied and, if approved, whether or not it was repaid. The evaluation criteria for the predictive model I was tasked to build was simply +1 if the model gave a loan and it was repaid, -1 if the model gave a loan and it wasn’t repaid, and 0 if the model denied a loan.

Though this may seems like a standard machine learning based approach, there are fundamental problems. The data are subject to a massive selection bias in that there is only information on repayment for loans that were given. There are loan applicants for whom their ability to repay the loans is unknown because it is untested. This is apparent when evaluating the a model with the criteria above as there is no way to score what happens when the model gives a loan when the ability to pay is unknown. Handling this problem is an incredibly complex problem and key to building any prediction on loans.

The ultimate goal of loan modeling is to predict the probability that a loan would be repaid given input data x: $P(LoanRepaid|x)$. The input data includes information like current salary, savings account balance, loan purpose and other financial information. But it could also

include demographic information like age, gender, and ethnicity as well geographical information like zip code.

The challenge is that this is not what we actually have in the data. Instead, we can only learn the probability that a loan was granted by the bank: $P(LoanGranted|x)$. And if the loan was given, the probability that it was repaid *given a loan was granted* $P(LoanRepaid|LoanGranted = True, x)$. The additional conditional statement is a way of mathematically representing the selection bias.

A naive approach is to approximate $P(LoanRepaid|x)$ as $P(LoanRepaid|LoanGranted = True, x)$. Concretely, this is done by throwing out training cases in which the loan was not given and learning to classify loans as either repaid or not repaid. This has the nice feature of avoiding the pesky uncertainty of whether or not loan would be repaid whenever the loan was denied. But has the bad feature of being terrible wrong. A classifier trained only on data for which a loan was approved is worthless as that data that is not representative of the distribution of new loan applications.

In the data I analyzed (which will be shown in a later post), one binary feature whether or not the applicant `is_employed`. Intuitively, this feature should be useful in determining if a loan would be repaid. However, when following the naive approach of only analyzing applications for which a loan was granted, `is_employed` has almost no predictive power for the simple reason that damn near every loan was given to someone employed. In other words, `is_employed` is very predictive of if a loan is given. But it is not useful for predicting if a loan is repaid because we have nearly no data on an unemployed applicant receiving a loan.

A different approach would be to model $P(LoanRepaid|LoanGranted = True, x)$ the probability that a loan is repaid given that the bank would grant the loan and information x and $P(LoanGranted|x)$, the probability that the bank would grant the loan given the information x. This means building two separate models. The first model is exactly the same as the naive model above. The second model is trained to predict whether or not the bank from whom the training data came would issue a loan. Once The decision to grant a loan would occur only if there is a high probability that the loan was granted by the bank *and* a high probability that a loan would be repaid given it was granted. The exact decision process depends on the bank's necessary repayment rate and on estimates of how likely people are to repay loans for whom the bank is likely to not grant a loan as well.^[math]

Why this is not value neutral

The learning we are doing on the financial data is heavily dependent upon $P(LoanGranted|x)$, the probability that the bank from whom we gained the data would grant a loan for an application with values x. This means that even if we want to be as diligent as possible to build a model that is fair to potential loan applicants, we are dependent upon the loan history of the bank. Whatever method the bank was using to determine loans will necessarily infect out model.

Consider a situation in which a bank was explicitly discriminatory in its loan practice and refused to give loans to immigrants. There are correlations between immigrants and other features. These correlating features would then correlated to whether or not a loan is granted. For example, a postal code with a large immigrant population would then become strongly correlated with a loan being denied.

Since our new model needs to learn in part from how the bank gave loans, these correlations would also be learned. That means that, if we do not work to correct the situation, our new model would 'learn' to discriminate against immigrants! This kind of infection of a loan decision, to the data, to the next model to make loan decisions means that discriminatory practices from years ago can still leave fingerprints in the data.

What can we do?

This is not an easy problem to solve. One possible solution is to intentionally include some randomization. If a small percentage of the time, a loan that would be denied is randomly approved. This would provided crucial data to explore the full distribution of loan repayments, but the cost of giving out loans with a potential higher rate of default.

Randomization of financial loans is not without precedence. In 2011, Nigeria launched the YouWin! competition for $50,000 cash grants to start a new business or expand an existing one. 729 of the grants were given out to a random selection of 1,841 semifinalist. The randomization of the award provided a powerful data for making future decisions about how effective the loans are and which potential recipients are most likely to succeed.[^McKenzie]

I want to be clear that this post is not meant to be an indictment of the loan industry. There are very smart people who work on exactly these problems. But people who are not well versed with thinking about data can forget to consider how the data are not created in a vaccuum. Whenever there is a strong selection bias such as occurs job applications or loan applications, the data will always be biased. And these biases can have long lasting and severe consequences. Data scientists need to recognize these risks and mitigate them as much as possible.

[^Rasmussen] Rasumssen Methodology

[^Scientific American] Scientific American Blog

[^Hecht] Hecht, Jennifer Michael (2003). “Whatever Happened to Zeus and Hera?, 600 BCE-1 CE”. Doubt: A History. Harper San Francisco. pp. 9–10

[^fivethirtyeight] fivethirtyeight

[^NYTimes] http://www.nytimes.com/2016/10/13/upshot/how-one-19-year-old-illinois-man-is-distorting-national-polling-averages.html

[^McKenzie] http://blogs.worldbank.org/impactevaluations/what-happens-when-you-give-50000-aspiring-nigerian-entrepreneur

[^math] Mathematically, we can be more precise. The desired quantity is $P(LoanRepaid|x)$. This can be factored in terms $P(LoanRepaid|LoanGiven, x)$ and $P(LoanGiven|x)$:

$$P(LoanRepaid|x) = \sum_{LoanGiven} P(LoanRepaid|LoanGiven, x) * P(LoanGiven|x)$$

$$¸P(LoanRepaid|x) = P(LoanRepaid|LoanGiven = True, x) * P(LoanGiven = True|x) + P(LoanRepaid|LoanGiven = False, x) * P(Loan$$

Since there is no knowledge of the the probability that a loan is repaid if a loan was not given the second term is impossible to determine from the data.