Foundations of Fairness in Machine Learning

Sampling and Fairness

Jakub Filipek

Introduction

My project was looking at the impact of under- and over- sampling methods on False Positive Rates (FPR) across different populations. Those methods are used when datasets have an imbalanced distribution of positive and negative examples, which is often the case in social datasets.
Most of oversampling and undersampling methods work by removing or adding points close to what algorithm thinks is a decision boundary in binary classification. However, if there are some groups of people (for example distinguished by their sex) that leave close to that boundary they might be either removed from the dataset entirely, or become a minority in relation to points artificially added to dataset. In both of these cases the resulting new, and supposedly improved dataset will portray them in an unfair, skewed way.

Past Work

There is not a lot of literature on this topic. However, there are 2 papers, which are closely related to my project.
Celis et al. in [1] describe a new method of subsampling a large dataset in such a way that it is fair and diverse. This is a very interesting notion, since most of such work does not take a significant interest in fairness, but rather in performance, and some methods will look at diversity of outputs.
Their approach is to modify a $k - DPP$ algorithm, by ensuring that these subset are also fair. This way, when they're later used for sampling only a few examples from them they are more likely to be fair. Simultaneously, the amount of subsets ensures diversity of the outcome.
However, the problem with this work is that it does not evaluate how such method affects performance, which is often the most important evaluation criterion for other under- and oversampling methods. This is something that I want investigate, and something that I believe would be a good follow-up to [1].
Another paper, is more recent, but also quite different in the goal. [2] describes a scenario in which a decision-maker can create a dataset, which seems fair to anyone who interacts with it, but is indeed biased.
While it might seem much different from what I am trying to attempt, the role of decision-maker in a dataset creation is not much different from oversampling algorithms. We can consider a decision-maker an oversampler, with an additional adversary goal. This is interesting due to the fact that oversampling (and undersampling) methods can lead to such problem even when there is no adversial goal. Their focus on overall accuracy can lead to huge sacrifices on smaller minority groups, or even larger ones, if they live close enough to boundary decision.

Experimental setup

I used a Census Income Dataset from UCI Machine Learning Repository, which defines a task to predict whether a person earns less or more that 50 thousand dollars per year. Since this is a binary task it is perfect from under- and over-sampling methods which often were developed based on such problems. Additionally, for every person it has their race and sex features, which I hid away from a predictive, over- and undersampling algorithms, but which were

used to define groups. These groups can be seen on Tables 1 and 2. There are only 7841 (24.08%) people with salary over 50 thousand dollars, out of 32561 in the whole dataset.
Every categorical feature in dataset was converted to a one-hot encoding for all of the possible values of that feature. This, along with removing race and sex resulted in 98 dimensional vector.
Model to make a binary prediction was a simple logistic regression model, with 98-dimensional input and 2-dimensional decision.
Along with the vanilla (unbalanced dataset) I tested 6 different data augmentation algorithms. All of the implementation were using a package from [3], available publicly:

- Random Oversampling: Every point from minority group (in our case 1 - people over 50k) has the same chance of being duplicated. Duplication are performed until minority and majority group are the same size.
- SMOTE: All points in minority group they all can be linked other points from minority group close by (typically within 3 neighbors). Linking means, that two points are taken, and an artificial one is generated somewhere on the line between the two. Note that this is the most basic implementation of SMOTE, and more advanced versions have been built on top of it. However, it is still popular, and worth investigating.
- BorderSMOTE: Uses SMOTE algorithm to generate new points. However, only points which have neighbors from different classes, but majority of its neighbors are still in the minority class. In other words, a minority point with 2 minority and 1 majority neighbor can be used, while point with all minority or all majority neighborhoods cannot be used.
- ADASYN: Similar to SMOTE, however each point will be used to generate amount of new points proportional to amount of majority neighbors.
- Random Undersampling: Every point from majority group (in our case 0 - people under 50k) has the same chance of being removed. Removals are performed until minority and majority group are the same size.
- Tomek: Removes a point in majority group if and only if it's nearest neighbor is a point in a minority class, and it is a nearest neighbor to that point too.

Every algorithm was used to produce a new dataset a 100 times (using different random seeds), and a model was trained for 5 epochs each time, using Adam optimizer.

Results

Due to space efficiency (tables were going over the edges of the page) I was only able to put these two algorithms and a unbalanced dataset for reference. However, all of the plots, along with some more discussion can be found it the final presentation. Additionally all of the code that can be used to reproduce these results can be found on this github repository.
Table 1 shows the FPRs of the best undersampling and oversampling algorithms on the Census dataset. We can see that both methods tend to keep Male FPR more stable than Female, when compared to unbalanced dataset. This might be related to the fact that there are twice as many males as there are females and hence they can much more likely to stay, or to enforce even further themselves with new points added around them.

	Random Oversampling	Tomek	Vanilla
Male	12.53% +/- 0.21	12.35% +/- 0.07	12.17% +/- 0.42
Female	6.23% +/- 0.19	5.54% +/- 0.16	4.98% +/- 0.39

Table 1: False Positive Rates of different sexes, when the best oversampling (Random Oversampling) and undersampling (Tomek) methods are used. Vanilla (unbalanced) dataset is also shown as a baseline performance.

However, when investigated we can see that this is not the case for SMOTE and ADASYN (In particular Male group FPR explodes to 20-24, while Female stays around 6-8), hinting that females are probably closer to the label boundary (based on salary, not race or sex), and hence they are getting a small boost from these two methods.
Table 2 shows similar results but with regards to race. Here we can see that both of these methods significantly increase FPR of American Indian/Eskimo and Black groups, while significantly lowering Asian/Pacific Islander. This can hint us that the first two groups are closer to the boundary and the last one is further away.
The reason for that is that Tomek links tend to remove points close to boundary, hence decreasing number of examples of these points, and possibly performance of the model on them. Since some point were removed model will perform better on the remaining ones, which is why Asian/Pacific Islander group benefits from this algorithm.

	Random Oversampling	Tomek	Vanilla
American Indian / Eskimo	13.07% +/- 0.58	10.07% +/- 0.17	8.36% +/- 0.94
Asian / Pacific Islander	10.34% +/- 0.55	12.71% +/- 2.07	13.63% +/- 1.58
Black	7.03% +/- 0.20	5.04% +/- 0.08	4.86% +/- 0.39
Other	7.45% +/- 0.67	6.11% +/- 0.19	6.01% +/- 1.21
White	11.21% +/- 0.19	10.60% +/- 0.08	10.26% +/- 0.40

Table 2: False Positive Rates of different races, when the best oversampling (Random Oversampling) and undersampling (Tomek) methods are used. Vanilla (unbalanced) dataset is also shown as a baseline performance.

A last observation is that FPR are skewed towards undersampling in our case. In particular, all of the oversampling methods have performed worse than Tomek links. The best intuition I can give for such trend is that when we remove negative examples with Tomek, we are removing negative samples close to the boundary, which are probably these false positives. However, when we add new positive examples close to the boundary they can push the boundary in such a way that includes more false positives, which causes oversampling methods to have worse FPR than unbalanced dataset.

References

[1] L. Elisa Celis, Amit Deshpande, Tarun Kathuria, and Nisheeth K. Vishnoi. How to be fair and diverse? *CoRR*, abs/1610.07183, 2016.

[2] Kazuto Fukuchi, Satoshi Hara, and Takanori Maehara. Faking fairness via stealthily biased sampling. *arXiv preprint arXiv:1901.08291*, 2019.

[3] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.