An Audit of Historical Mortgage Approval Data in Vermont Using PAC and Metric-Fair Learning

Albert Qi
Harvard College
Cambridge, Massachusetts, USA
albertqi@college.harvard.edu

Ronak Malik
Harvard College
Cambridge, Massachusetts, USA
ronakmalik@college.harvard.edu

ABSTRACT

Decision-making algorithms are being used more often in salient situations. One area in which there is rising concern that algorithms might perpetuate or augment existing biases is evaluating credit risk and lending. The loan approval process is usually opaque and functions with impunity. We want to further investigate this topic by building a fair model that determines the risk taken on by a bank and ultimately predicts an approval or denial of a mortgage using mortgage application information.

We develop a (dis)similarity metric that uses a human arbiter. We then train an unconstrained linear classifier on historical mortgage approval data in Vermont, reaching convergence at 83% accuracy. Using our metric, we then retrain our linear classier under a set of fairness constraints, showing that there is no significant difference in accuracy on the same data. This implies a possibly fair mortgage approval process in Vermont.

All source code can be accessed at https://github.com/albertqi/pacf-hmda2022.

1 INTRODUCTION

Algorithms are increasingly being deployed in our everyday lives to make decisions where humans existed prior, from crucial circumstances like setting bail amounts and determining the chances of recidivism to seemingly simple tasks such as recognizing our faces when we unlock our phone. While humans are not known for their fair and equal judgement of individuals or groups, it is just as important that we prevent these biases from being carried over to the algorithms that we design (especially as they become more prevalent) as it is to stop them from happening in the present.

One particularly high-stakes industry for algorithm abuse is credit risk evaluation. Historically, the credit industry has had notorious cases of discriminatory practices in providing (or more accurately withholding) financial services from minority racial and ethnic groups [6]. While modern day discrimination does not happen with markers and maps, there could still be bias and discrimination embedded deep in the credit evaluation process. It is crucial that as more firms use automated approval systems, prejudice and discrimination are eliminated by decision-making algorithms and not perpetuated by them.

There do exist some essential public disclosures when it comes to salient credit approval, the primary being the Home Mortgage Disclosure Act (HMDA) [4].[1] Each time a financial institution makes a decision about a mortgage application, the application data and the approval decision must be reported to the Federal Financial Institutions Examination Council (FFIEC) and are later aggregated and publicly disclosed. Although the FFIEC performs internal and periodic inspections of the HMDA and the reporting institutions, we believe that it is the public's responsibility to additionally evaluate such data to ensure accountability and fairness (see Section 3.2).

The goal of this paper is to audit a selection of the HMDA database using modern fairness definitions and techniques. Modern definitions of fairness require being able to compare the similarity of individuals, but no such metric exists for mortgage application data (particularly using only the fields available in the HMDA). We develop such a metric that can be applied (and perhaps generalized) to our selection of the HMDA using a known metric-generation algorithm. We also provide an implementation of a fair classification framework that, along with our (dis)similarity metric, we use to determine if the mortgage approval processes used on the applications in our selection of the HMDA are fair.

Some of the main challenges in developing a PACF model on historical mortgage approval data are (1) dealing with highly irregular data, (2) the nonexistence of a distance metric, and (3) determining arbitrary approximation parameters for PACF learning.

(1) The data is formatted inconsistently. For example, percentages can be expressed as any of 40%, 40, 0.4, and >40%. Additionally, some applications are missing data; we handle this via imputation.
(2) There is no objective distance metric, making it difficult to evaluate similarity. We develop our own metric based on what we deem reasonable.

[1] Throughout the rest of this paper, we may use "HMDA" to refer to the database itself.

(3) It is challenging to determine the optimal parameters α and γ for PACF learning in our context. Again, we just use what we find reasonable.

We begin by filtering and pre-processing the 2022 HMDA dataset to only include applications from Vermont. Then, we construct a (dis)similarity metric for individual fairness via a human arbiter. This allows us to find the distance between any two individuals. Afterward, we train an unconstrained linear classifier on our dataset that reaches 83% accuracy. Finally, using our metric, we retrain our classifier under a set of fairness constraints to obtain a PACF model that reaches 81% accuracy.

Overall, we make the following contributions within this paper:

(1) We highlight the need to analyze historical mortgage approval data.
(2) We construct a (dis)similarity metric for individual fairness using a human arbiter on the HMDA dataset.
(3) We develop both unconstrained and constrained classifiers on the HMDA dataset, reaching 83% and 81% accuracy, respectively, indicating that the mortgage approval process in Vermont might be fair.

2 BACKGROUND

We build upon some foundational concepts in the field of algorithmic fairness. Below is brief overview of the work of which we build off.

2.1 Fairness Through Awareness

To audit the fairness of historical mortgage approval data, we need to first understand what fairness itself means. Dwork et al. introduce the concept of individual fairness, whereby similar people are treated similarly [3]. We use this idea to determine whether or not historical data is fair (i.e., do similar individuals have similar approval rates?). This does, however, require us to also know how similar or dissimilar any two individuals are (see Section 4).

3 DATASET

We begin by filtering and pre-processing our dataset. There are some irregularities and edge cases that we address as well.

3.1 HMDA Mortgage Applications

We use the 2022 HMDA [4] dataset of mortgage applications, filtered to the state of Vermont in order to narrow down our scope and obtain a more specific audit result (i.e., banks acting across states may act differently).

We carefully choose our features according to what we deem to be most valuable in evaluating mortgage applications. For example, we did not think that the type of entity purchasing a covered loan from the institution would directly impact individual mortgage approvals that much. Additionally, we one-hot encode our categorical features before using them in training. Finally, note that we utilize derived features, which aggregates information about the applicant and co-applicant together. This allows us to capture a more holistic view of the combined situations of both parties, potentially increasing our model's accuracy by reducing the number of features and aggregating related information into a single feature.

Table 1: Age Demographics for Full Dataset and Subset with Empty Values

Age	Full Dataset	Subset
<25	2.26%	3.10%
25-34	17.84%	14.26%
35-44	24.59%	20.98%
45-54	21.56%	20.06%
55-64	17.61%	16.16%
65-74	10.25%	10.32%
>74	3.50%	4.44%
Not Available	2.39%	10.68%

Table 2: Sex Demographics for Full Dataset and Subset with Empty Values

Sex	Full Dataset	Subset
Male	26.86%	31.19%
Female	19.27%	19.93%
Joint	39.45%	26.65%
Not Available	14.42%	22.23%

Table 3: Ethnicity Demographics for Full Dataset and Subset with Empty Values

Ethnicity	Full Dataset	Subset
Hispanic or Latino	1.43%	1.67%
Not Hispanic or Latino	76.45%	68.00%
Joint	1.31%	1.09%
Not Available	20.81%	29.24%

3.2 Missing Data

A major, increasingly pressing, problem with the HMDA dataset is the presence of missing values in specific application listings. Around 18% of entries in the 2022 HMDA have race demographic information for the borrower listed as "Not Available", and an even higher 20% are not listed in

Table 4: Race Demographics for Full Dataset and Subset with Empty Values

Race	Full Dataset	Subset
White	76.27%	68.63%
Asian	1.30%	1.08%
Black or African American	0.74%	0.86%
American Indian or Alaska Native	0.31%	0.43%
Native Hawaiian or Other Pacific Islander	0.10%	0.17%
Two or More Minority Races	0.05%	0.07%
Joint	1.57%	1.12%
Not Available	19.66%	27.64%

Vermont. Furthermore, many rows just contain cells with empty values, omitting income, debt-to-income ratio, and even loan amounts in some cases. While this is sometimes because the information is not available to the financial institution at the time of reporting [2], it seems like more and more banks make little attempt at accurately collecting/reporting demographic information. This makes it extremely difficult to know if the trends that researchers and auditors see in the data truly exist and can obscure real trends that are actually there. The waning fidelity of this essential resource is a detriment to many. Consumers are less protected against predatory loans and the inconsistent information makes it difficult to hold financial institutions accountable for acting unfairly.

We choose to not drop rows with null values as this would remove a significant number of the application denials, changing the denial rate in the dataset from 17.33% to 0.43%. Instead, we choose to impute our missing values, using the median for numerical features and the mode for categorical features. The demographic breakdown of rows with missing data is shown in Tables 1 to 4. Interestingly, there is no major increase in any one specific group's proportion, but rather rows with null values tend to also have more data points listed as "Not Available".

4 METRIC LEARNING

4.1 Background

Recall that we want to test whether or not similar individuals are treated similarly. In order to actually test this, though, we need to first come up with a (dis)similarity metric. That is, given any two individuals, how can we determine how similar or dissimilar they are? Christina Ilvento proposes an algorithm that returns a metric by utilizing a human arbiter in the process [5].

The algorithm takes advantage of the fact that triplet queries (i.e., which one of x and y is closer to r?) are much easier than exact distance queries (i.e., how far is x from y?). Thus, we can pick a representative r and run a modified merge sort algorithm that makes use of these triplet queries to sort individuals based on their distance to r.

After obtaining this sorted list, we want to actually attach real-valued distances between individuals and find an α-submetric $\mathcal{D}'$ of $\mathcal{D}$ for representative r. To do so, we approximate each individual's distance to r in a recursive manner with an α margin of error. This allows for a sublinear number of exact distance queries.

We repeat this process for a random subset of representatives, giving us multiple α-submetrics. We then combine these into one cohesive α-submetric by performing a max merge. This allows us to achieve a parallax effect and gives us a metric to use for our evaluation.

4.2 Modifications

Ilvento's metric learning algorithm for individual fairness requires $O(n \log n)$ number of triplet queries. While these queries are easier to perform than exact distance queries, they still require human intervention, and it is still infeasible for a human to manually perform $20000 \log 20000 \approx 285754$ queries. Thus, we automate these triplet queries by returning whichever vector has a lower Euclidean distance to the representative vector, under the reasoning that we still obtain an approximate sorting of individuals.

Additionally, it is challenging to choose enough representatives for us to produce a nontrivial submetric with probability at least $1-\delta$. Given our constraints as humans, we thus choose a random sample of only five representatives. While this might not be enough for nontriviality, it still should allow us to achieve some sort of parallax effect.

Lastly, we note that hard queries are, unsurprisingly, very hard. In deciding the real-valued distance between two individuals, we only have our best judgments and intuitions, which may be incorrect or inconsistent. We are not oracles that know the objective truth, meaning our metric's accuracy is limited by our lacking knowledge of the subject area and inconsistencies as humans.

5 PACF LEARNING

5.1 Background

With this (dis)similarity metric, we can now look into developing a classifier under a set of fairness constraints. Probably approximate correct and fair (PACF) learning is a PAC learning framework originally introduced by Guy Rothblum and Gal Yona [7] based on the seminal work of Dwork et al. [3] which describes a classifier that can be subjected to fairness constraints and can be trained in polynomial time.

As discussed in the PACF paper, perfect metric fairness is often unattainable in practice due to the inherent complexity and variability of real-world data and the specific problem being solved. Rothblum and Yona show that it is

intractable (and sometimes even unsolvable) to find a classifier that fairly treats all individuals who are similar in all circumstances (with respect to some metric), and using a relaxed version of fairness offers better generalization properties. Specifically, Rothblum and Yona prove that a PACF classifier generalizes in fairness from the training sample to the underlying population.

The main property we aim to satisfy with a PACF classifier is a relaxed fairness constraint that allows the classifier to be unfair in *some* cases and be *slightly* inaccurate in its estimations of the similarity between people. PACF defines an (α, γ)-approximately metric-fair classifier as one that treats less than an α fraction of pairs of individuals unfairly with a γ slack in estimating how similar two individuals are. More formally, let $M(S) \in \mathcal{D}$ be a sample of individuals from the distribution that defines the population and d be the real similarity metric (developed from the metric creation algorithm in Section 4). For a classifier h to be (α, γ)-approximately metric-fair, the following must hold

$$\Pr_{x,x' \sim M(S)}[|h(x) - h(x')| > d(x, x') + \gamma] \leq \alpha$$

while solving the following optimization problem

$$\arg\min_{h \in H} err_S(h) \text{ s.t. } h \in \hat{H}^{\alpha,\gamma}$$

where $\hat{H}^{\alpha,\gamma}$ is a class of (α, γ)-approximately metric-fair classifiers. However, even for linear classifiers, the set $\hat{H}^{\alpha,\gamma}$ is non-convex, so this is a non-trivial optimization problem to solve. The PACF paper introduces an additional constraint on the set of approximately fair classifiers to convexify the set by also requiring that classifiers have an ℓ_1 metric-fairness violation loss bounded by $\tau = \alpha \cdot \gamma$. The formulation for the constraint is given below:

$$\xi(h) = \sum_{(x,x') \in M(S)} \max(0, |h(x) - h(x')| - d(x, x'))$$
$$\xi(h) < \tau$$

The proof for this convexification is given in the PACF [7] paper (only holds for linear classifiers). This new constrained set is labeled $H^{\tau}_{\ell_1}$, with $H^{\tau}_{\ell_1} \subseteq \hat{H}^{\alpha,\gamma} \subseteq H$, where H is the set of all unconstrained classifiers.

5.2 Modifying the Constraint

While the paper's authors claim that a convex set is enough to efficiently solve the aforementioned optimization problem, the non-differentiablility of the constraint creates some difficulty in numerically optimizing a classifier (e.g. with gradient descent). However, if we slightly reformulate the metric-fairness violation constraint, we can maintain similar properties with a differentiable function, making it easier to use standard numerical methods to find a satisfying classifier.

The two operations that are non-differentiable are the max and absolute value functions. Let us re-write the function for convenience:

$$\xi(h) = \sum_{(x,x') \in M(S)} \max(0, |D(x, x')| - d(x, x'))$$
$$D(x, x') = h(x) - h(x')$$

We can remove the max function by simply taking the two sided distance error which is strictly greater than the one sided error (we scale up distances to prevent the square from reducing error), resulting in a sum of squares:

$$\xi(h) = \sum_{(x,x') \in M(S)} (|D(x, x')| - d(x, x'))^2$$

Similarly, we square the classifier estimated distance and the metric distance to remove the absolute value. Since the values of distance are intended to be non-negative, the minimas and maximas are retained and optimization will produce the same results. The resulting differentiable constraint function is as follows:

$$\xi(h) = \sum_{(x,x') \in M(S)} (D(x, x')^2 - d(x, x')^2)^2$$

This does increase the strictness of fairness on the classifiers, but this does not affect our final result. It is noteworthy that this also changes the definition of fairness defined by Dwork et al. to enforce that dissimilar individuals be treated dissimilarly.

5.3 Implementation

Now that we have a differentiable constraint, we implement projected gradient descent (PGSD) to numerically find a satisfying classifier for our dataset. PGSD takes place in two steps: (1) standard gradient descent on the loss function to optimize for correctness, and (2) a projection step to constrain the new classifier to the set of points defined by the constraint. Let $f(h)$ be the loss function with respect to model correctness and δ be the step size. The process we follow is described in a paper by Anderson Ang [1]:

(1) Pick an initial point $h_0 \in H$
(2) Repeat until complete:
  (a) Compute $\nabla f(h_i)$
  (b) Let $y_j = h_i - \delta \nabla f(h_i)$
  (c) While $\xi(y_j) > \tau$:
    (i) Compute $\nabla \xi(y_j)$
    (ii) Let $y_{j+1} = y_j - \delta \nabla \xi(y_j)$
  (d) Let $h_{i+1} = y_{j_n}$

This is slight modification of the process described in the paper. Here, we perform projection (step c), by running an additional gradient descent step until our current classifier is within $H^{\tau}_{\ell_1}$. Since $\xi(h)$ is a continuous function, this is equivalent to projection. At each point of the domain, subtracting

the current gradient with respect to the classifier will bring us closer to the nearest point where $\xi(h) < \tau$.

6 EVALUATION

In order to determine historical fairness of the Vermont HMDA, we first train a standard linear classifier to see if we can replicate the loan approval decisions in the region. We then train a PACF linear classifier with fairness constraints to see how the accuracy changes. By first training a classifier that models the existing data, a statistically significant accuracy drop in the fair model would indicate unfairness in the original data.

We train a linear classifier because the PACF framework requires a linear classifier for tractable training, and a linear classifier seems to have sufficient predective power for our use case.

6.1 Experimental Setup

We run everything on a 16-core, 32 thread AMD EPYC 7R32 with excess memory. All of our training and testing is performed with a batch size of 64. Training and testing data are split at a 3:1 ratio.

We first run the pre-processing step described in Section 3. We then generate our evaluation metric from the overall dataset and train the linear and PACF classifiers on the same data. Finally, we evaluate the results from the two classifiers.

We evaluate the performance of the classifiers based on how accurately they are able to predict the outcome of the application approval process. Fairness is used as an overall evaluation metric (i.e., comparing the accuracy of the unfair and fair classifiers) but is not an individual metric for the PACF classifier because it is a constraint and must hold for a potential classifier to be considered.

6.2 Linear vs. PACF Classifier

When training a classifier without fairness constraints, our model converges to 83% accuracy. This indicates that our model has reasonably strong predictive power. We then retrain the model under our fairness constraints, and the model converges to 81% accuracy. This is still fairly strong, and its accuracy is not too far from our unconstrained model given the number of data points ($\approx$ 20000).

6.3 Analysis

The drop in accuracy is fairly insignificant. This could mean that (1) the historical mortgage approval data is actually fair, (2) our linear classifier is not complex enough to highlight the impact of our fairness constraints, (3) our metric itself is too weak, or (4) our results are skewed by missing data.

6.3.1 Possibility 1: The Data is Actually Fair. It is always possible that the historical data that we analyze is actually fair. If this were the case and similar individuals were indeed treated similarly, then it makes perfect sense that our classifier does not lose much accuracy when training under fairness constraints. Because the data itself is already fair, the fairness constraints would not impact the model's convergence.

6.3.2 Possibility 2: Our Classifier is Not Complex Enough. Another possibility is that our classifier is simply not complex enough to demonstrate the impacts of the fairness constraints. Because both our classifier and our constraints need to be linear, we lose out on some flexibility. A more complex model may result in a higher accuracy when trained normally (e.g., 95%) and the same accuracy when trained under fairness constraints (e.g., $\approx$ 80%). Such a model would clearly demonstrate the effects of our fairness constraints, but our model might not be that complex enough.

Additionally, the features on which we choose to train may be suboptimal. A better selection of features could result in the aforementioned results and improve the model's base accuracy. There may also be some features that HMDA uses in its decision to approve or deny a mortgage application that are hidden from the public (e.g., credit score, which could have its own set of biases). This would further limit the power of our model to show the impacts of our fairness constraints.

6.3.3 Possibility 3: Our Metric is Too Weak. It is possible that our metric is too weak to capture the effects of our fairness constraints. If our metric were indeed too weak or wildly incorrect, then the classifier might always believe itself to be fair and never optimize for fairness. In other words, if our metric is incorrect, then our fairness constraints are essentially meaningless as well.

We are neither experts nor oracles, so it is entirely possible that our human errors translate into our metric, too.

6.3.4 Possibility 4: Skewed by Missing Data. As noted in Section 3.2, many applicant entries are missing data. This could have skewed our results because the classifier is unable to pick up on significant trends that would be hidden behind missing data.

7 FUTURE WORK

In the future, we would like to perform more tests in order to isolate the problem at hand and figure out which possibility is most likely.

First, we would change our feature selection in order to see how a different set of features might affect our model's accuracy. This could involve increasing or decreasing the number of features we include from the HMDA.

Additionally, we would want to strengthen our metric by max merging with more even more submetrics. Doubling or even tripling the amount of representatives might drastically improve our metric's performance, which would help highlight the effects of our fairness constraints. We may even become more and more consistent in our real-valued queries, which would enhance the metric even further.

Lastly, we could try to increase the complexity of our model in order to improve its accuracy. This would involve drifting away from linear classifiers, meaning we would need to re-adapt our methodology for non-linear classifiers.

Ultimately, all of our future work involves attempting to further demonstrate the impacts of our fairness constraints.

8 CONCLUSION

When we utilize algorithms in our daily life, we often do not stop to think about whether or not the underlying processes are fair. However, given how more and more algorithms are being increasingly utilized, fairness in these algorithms has become increasingly critical. We do not want these algorithms to exacerbate any inequalities or biases that currently exist.

The credit industry is no exception to this. There have been historical cases of discriminatory practices, and the evaluation process might still be biased in some way. Thus, we audit a selection of the HMDA database and see whether or not similar individuals are treated similarly. This gives us an idea about how fair the mortgage approval process actually is.

We filter and pre-process the HMDA dataset to only include Vermont, and use Ilvento's metric learning algorithm in order to develop a (dis)similarity metric between individuals; this allows us to find the distance between any pair of individuals. We then train an unconstrained linear classifier on the dataset, giving us an accuracy of 83%. Finally, we retrain our classifier under a set of constraints, resulting in a PACF model that converges at 81% accuracy, perhaps indicating a fair mortgage application evaluation process in Vermont.

REFERENCES

[1] Anderson Ang. 2024. Projected Gradient Algorithm. (2024). https://angms.science/doc/CVX/CVX_PGD.pdf

[2] Compli(cated). 2011. App Withdrawn - Do Not Have Income. https://www.bankersonline.com/forum/ubbthreads.php/topics/2202622/re-app-withdrawn-do-not-have-income

[3] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness Through Awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. Association for Computing Machinery, 214–226. https://doi.org/10.1145/2090236.2090255

[4] FFIEC. 2022. Snapshot National Loan Level Dataset. (2022). https://ffiec.cfpb.gov/data-publication/snapshot-national-loan-level-dataset/2022

[5] Christina Ilvento. 2020. Metric Learning for Individual Fairness. (2020). arXiv:1906.00250 http://arxiv.org/abs/1906.00250

[6] Jonathan Rose. 2023. Redlining. (2023). https://www.federalreservehistory.org/essays/redlining

[7] Guy Rothblum and Gal Yona. 2018. Probably Approximately Metric-Fair Learning. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 5680–5688. https://proceedings.mlr.press/v80/yona18a.html