LEARNING TO LEND IN DYNAMIC, RESOURCE-CONSTRAINED ENVIRONMENTS

Eric Mibuari, David C. Parkes *
School of Engineering and Applied Sciences
Harvard University
mibuari@g.harvard.edu, parkes@eecs.harvard.edu

Berk Ustun
Department of Computer Science & Engineering
University of California, San Diego
berk@ucsd.edu

ABSTRACT

In lending domains, machine learning can be used to learn a predictive model of the probability of default (a "risk score"), this driving loan decisions. For simple models, this brings the benefits of transparency and explainability, as well as guidance in regard to recourse. An alternative is to use *policy learning*, that is, learning a loan policy directly, and without concern to risk scoring. This emphasizes profit and can speed up learning, but with a concomitant loss of transparency. In this paper, we study a risk-score based policy learning method, demonstrating both good profitability and transparency. We also define a *recourse effort fairness* metric, and demonstrate that a risk score-based policy approach achieves optimal profits and recourse effort fairness, along with explainability and transparency. For this risk score-based policy approach, we further define and motivate the challenge of *recourse effort fairness portability*, a desideratum in resource-scarce contexts (e.g., developing countries), as a set-up for our ongoing work.

1 INTRODUCTION

A current practice in the use of machine learning to support lending decisions is to learn a predictive model of risk to generate repayment probabilities that can be used to guide loan decisions. In particular, *credit scores* are designed with the goal of accurately predicting default rates (Siddiqi, 2012). Beyond the transparency and explainability benefits that come from these risk models, there are also regulatory frameworks, such as the U.S. *Equal Credit Opportunity Act (ECOA) Regulation B*, that are based on risk models. An explicit risk score can also help protect against predatory lending: a regulator can say "don't lend to customers with a predicted default risk below a certain threshold."

Another advantage of accurate risk models is that they support transparency, for example, to explain loan denial decisions (as required by law in some countries). Transparency of the basis for a decision also helps with sense-checking and human scrutiny of decisions. A risk model outputs a value, i.e., the predicted risk of default, and since this output is grounded, its validity can be checked by a human expert. A risk model also enables *portability*, where this is useful, whereby a model developed for one context, can be used in another context and still achieve a desired objective. For example, if we have two different lending markets, A and B, that differ in product and thus interest rate, a policy approach trained on A would implicitly bake A's interest rate within its learned policy, whereas a suitable risk model can be employed in either market. In resource-constrained settings, such data reuse may be critical to make optimal use of limited data, which may not be available across all markets where the model could be deployed.

*Also with DeepMind, London, UK.

Nevertheless, some, such as Kilbertus et al. (2019), have observed that learning to predict accurately may not maximize profit because it can attend to prediction errors that are immaterial to making the right decision for a lender. Rather, they present a pipeline that adopts negated profit as the loss functions, and directly learns a decision policy (and without learning an explicit risk model). We argue that this approach can be undesirable because, without a risk model, the additional objectives of explainability and transparency cannot be achieved. Moreover, the last decade has seen a steady growth of machine-learning enabled credit scoring systems for lending in developing countries. This has been made possible by the growth of new data streams, including from mobile phone use behavior. However, this growth has not been accompanied by a commensurate focus on fairness considerations in these resource constrained settings. In particular, we focus on *recourse* (Ustun et al., 2019) and ask whether a learning pipeline can provide for fairness in regard to the actions that an individual can take to remedy a situation in which they are declined a loan. We ask whether we can have the best of both of worlds. We study a risk-score based, policy learning method, which makes use of an explicit risk model while also driving learning primarily by the consideration of maximizing expected profit.

We introduce a new metric, which we call *recourse effort fairness*. We are motivated by Roemer's theory (Roemer & Trannoy, 2015; Beretta et al., 2021): *A ranking is said to be fair if all individuals belonging to different types have the same opportunity to reach a high ranking position, and if relevance and exposure are given by personal responsibility and not by birth circumstances*. In our context, we can take type as demographic group, relevance as credit score, and exposure as loan approval. In formalizing Roemer's framework, we relate an individual to a cumulative distribution function of their credit score, for individuals of the same type. Specifically, two individuals belonging to a different type who occupy the same quantile in their respective distribution functions have exerted the same level of effort and therefore the same level of responsibility. Given this, and in the case of a deterministic loan policy, we define the recourse effort for an individual who does not receive a loan as the minimum amount by which they must increase their quantile in this distribution in order to receive a loan. The *recourse effort error* is defined as the difference between the *actual recourse effort* due to a learned model and the *fair recourse effort*, i.e., the recourse effort that should be exerted given a ground truth model of an individual's latent quality. A similar approach is used by Ustun et al. (2019), who present cost functions for recourse based on score percentiles of an individual in the target population.

In simulation, we introduce a new policy learning algorithm that makes use of an explicit risk model, and adopt the evaluation metrics of lender profit, predictive accuracy, and recourse effort fairness. We show that our *model-based policy learning* approach obtains near optimal profits across different environments, whereas an accuracy driven, model-based approach is sub-optimal in more complex environments. Further, model-based policy learning has higher *recourse effort fairness* and achieves better accuracy than the other two approaches, also achieving better explainability.

Related work. Our paper is most closely related to that of Kilbertus et al. (2019), who consider lending as a problem with censored training data, where the repayment or default is only observed for applicants who are approved loans. These authors adopt off-policy learning for this problem, making use of importance weighting to give an unbiased profit estimator, and they adopt a randomized decision policy and show how to directly learn a profit-maximizing loan policy over time. Our work extends this approach to make it model-based while retaining loss defined as negated profit, so that the policy is defined on the basis of an explicit risk model.

Our work also ties into the larger literature on machine learning and fairness (Dwork et al., 2011; Hardt et al., 2016; Chouldechova, 2016). In particular, some papers have made an explicit distinction between predictions and decisions (Corbett-Davies & Goel, 2018; Valera et al., 2018). Our work follows this stream of treating label predictions and decisions as distinct, and broadly relates to research on fairness for learning in dynamic environments e.g., (Hu & Chen, 2017).

2 MODEL

In this section, we formalize the lending problem. We consider a loan approval problem where a bank provides loans to applicants over each of $T \geq 1$ periods for $t \in \{0, 1, \ldots, T\}$. We represent the bank's lending decision for an applicant using a *policy function*. Formally, a *policy* $\pi : \mathcal{X} \to [0, 1]$

is a function that takes as input a vector of d features from a specific applicant, $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ (e.g., $\boldsymbol{x} = (\texttt{age}, \texttt{income}, \texttt{n\_credit\_cards})$, and returns the probability with which the individual is assigned a loan. $y \in \{0, 1\}$ is a repayment indicator, where $y = 1$ if loan will be repaid, $k \in \{0, 1\}$ is the loan amounts index set, $l \in \mathbb{R}$ is the loan amount, $\theta \in \mathbb{R}^d$ are the policy parameters, and $p(\boldsymbol{x})$ is repayment probability $\Pr(y = 1|\boldsymbol{x})$. x and y are sampled from a ground truth distribution $P(X, Y)$. The loan award decision $k \in \{0, 1\}$ is sampled from a policy $\pi(\boldsymbol{x})$.

The bank considers a class of policy functions Π_θ. Each policy π_θ is specified by a parameter vector $\theta \in \mathbb{R}^d$. The optimal policy for a bank is $\pi^*_\theta = \arg\max_{\pi_\theta \in \Pi_\theta} u(\pi_\theta)$, where $u : \Pi \to \mathbb{R}$ is a utility function that measures a bank's *expected profit*. In a setting where a bank lends $\$l$ at an interest rate α, it receives $(1 + \alpha) \cdot l$ if an applicant repays their loan, and loses $-l$ if an applicant defaults on the loan. Thus, the expected profit from making a loan is

$$u(\pi_\theta) = \mathbb{E}_{x,y \sim P, k \sim \pi_\theta(x)} \left[\mathbb{1}(k = 1)(\mathbb{1}(y = 1) \cdot \alpha \cdot l) - \mathbb{1}(y = 0) \cdot l] + [\mathbb{1}(k = 0) \cdot 0\right] \quad (1)$$

The bank starts with an *initial dataset* of n_0 examples $(\boldsymbol{x}_i, y_i)_{i=1}^{n_0}$, for example by making loans for a short period of time in pure exploration mode. At period t, the bank receives applications from $n_t \geq 1$ consumers, and samples a lending decision from $\pi_t(k = 1|\boldsymbol{x}_i)$. At the end of period t the bank updates the policy function using the dataset of n_t training examples $(\boldsymbol{x}_i, y_i)_{i=1}^{n_t}$, where $\boldsymbol{x}_i \in \mathcal{X} \subseteq R^d$ is a vector of $\boldsymbol{x}_i$ for applicant i, and $y_i \in \{0, 1\}$ is a label that is set as 1 if applicant i repays the loan. The bank observes feature vectors $\boldsymbol{x}_i$ for every applicant, but only observes y_i for applicants who receive a loan.

3 METHODOLOGY

In this section, we describe different approaches to learn a lending policy.

3.1 MODEL-BASED LEARNING – MODEL

The lender learns a repayment model, and acts myopically to maximize profit in each period given the current model. In particular, the decision rule applies a hard threshold to the repayment probability; i.e., $\pi_\theta(k = 1 \mid x) = \mathbb{1}(p(y = 1|x) > Th)$ where Th denotes the minimum predicted risk needed to approve a loan. Setting $Th = 1/(1 + \alpha)$ maximizes expected profit.

3.2 POLICY LEARNING – POLICY

The lender learns a loan approval policy that maps features to loan approval decisions (Kilbertus et al., 2019). This approach avoids the need to learn a model to predict the risk of repayment, and uses the data to train a parameterized policy function: $\pi_\theta(k = 1 \mid x) = \sigma(\beta \cdot \theta^\top \phi(x)) \in [0, 1]$. The expected profit under this policy can be determined by the ground-truth distribution P as in Equation 1. The policy is updated across periods through stochastic gradient ascent with learning rate $\eta > 0$. Given that the data collected in a given lending period is induced by policies in prior periods, the data suffers from selection bias. In turn, Kilbertus et al. (2019) compute an unbiased estimate of the gradient on expected profit for policy π_θ using weighted importance sampling as:

$$\nabla_{\theta_t} u(\pi_{\theta_t}) = \mathbb{E}_{x,y \sim P_{\theta_{t-1}}} \left[\frac{\nabla_{\theta_t} \pi_{\theta_t}(k = 1|x)}{\pi_{\theta_{t-1}}(k = 1|x)} v_1(y_1) \right] \approx \frac{1}{n_{t-1}} \sum_{i=1}^{n_{t-1}} \left[\frac{\nabla_{\theta_t} \pi_{\theta_t}(k = 1|x_i)}{\pi_{\theta_{t-1}}(k = 1|x_i)} v_1(y_i) \right] \quad (2)$$

Here n_{t-1} is the number of loans approved by policy $\pi_{\theta t-1}$ at period $t - 1$. This gradient is then used to update the policy as: $\nabla \leftarrow \nabla + \eta \nabla_\theta u(\pi_\theta)$.

3.3 MODEL-BASED POLICY LEARNING – MBPOL

The lender learns an explicit risk model with which to parameterize a decision policy, with learning again adopting loss as negated expected profit. We refer to this new approach as *model-based policy learning* and summarize the procedure in Algorithm 1 in the Appendix. We model the repayment probability as $p(y = 1|\boldsymbol{x}, k = 1) = \sigma(\theta^\top \phi(\boldsymbol{x}))$, where $\sigma(a) = \frac{1}{1+\exp(-a)}$ and $\phi(\boldsymbol{x})$ is a basis function transformation of $\boldsymbol{x}$. Given this, we approve a loan with a policy decision that is defined as a logit function on the expected utility (Equation 11),

$$\pi_\theta(k = 1|x) = \sigma(\beta \cdot v) = \sigma\big(\beta \cdot l \cdot \big(\sigma(\theta^\top \phi(x))(\alpha + 1) - 1\big)\big) \in [0, 1], \quad (3)$$

where $\beta > 0$ is a hyperparameter that controls the degree of exploration. A similar gradient computation with weighted sampling is done, as in the case of policy learning, but as appropriate to this new parametric form for the policy and as detailed in the Appendix. In particular, a model accuracy penalization is applied to the SGA gradient update step: $\nabla \leftarrow \nabla + \eta(\nabla_\theta u(\pi_\theta) - \gamma \nabla_\theta(P_\theta))$, with γ as the penalization rate.

4 EXPERIMENTAL RESULTS

In this section, we present results from experiments on synthetic datasets where we compare these different approaches to learning lending models. We evaluate the performance of each approach in terms of multiple performance metrics, and across different kinds of data distributions that may arise in practice.

4.1 METRICS

We consider the following performance metrics, evaluated after the final training period:

Profit: The lender utility is the average profit made per applicant.

Model Accuracy: We measure *balanced accuracy,* which is the average between the sensitivity and the specificity; i.e., the average of the decision accuracy on positive and negative examples.

Model Calibration: We measure calibration using the root mean square error of the probability predictions of a model over a sample with n points. The value is computed as $\sqrt{\frac{1}{n}\sum_{i=0}^{n}(\hat{p}_i - p_i)^2}$ where $\hat{p}_i$ and p_i denote the predicted and true values of the probability of repayment for person i, respectively. Calibration can only be computed for methods that output probability predictions, and is therefore marked "N/A" for Policy approach in Table 1.

Recourse Effort Fairness. This fairness metric entails calculating the *recourse effort error* defined as the absolute difference of *actual recourse effort* and *fair recourse effort*. This difference is only computed among individuals who are denied loans – i.e., not approved (under a deterministic policy), or who have a probability of approval $\pi(\boldsymbol{x}) \leq 50\%$ (under a stochastic policy).

We can define these terms for model-based methods as follows. Given an individual with a predicted probability of approval $\pi(\boldsymbol{x}) \in [0, 1]$, where $\pi(\boldsymbol{x}) := \mathbb{1}[g(\boldsymbol{x}) \geq Th]$, where Th is the loan approval threshold, we define actual recourse effort$(x) := \min \delta(x', x)$ s.t. $\pi(x') > 0.5$, where $\pi(x) \in [0, 1]$ is the probability of approval for a person with features x, and $\delta(x', x) := F(x') - F(x)$, $F(x) :=$ Pr$(f(X) \leq f(x))$ is the CDF of true repayment probability f with respect to the random variable X. We define *fair recourse effort*$(x) = \min \delta(x', x)$ s.t. $p(x') \geq Th$, where $\delta(x', x) = F(x') - F(x)$, and $p(x')$ is the true repayment probability for x'.

Methods. We consider the three methods, *model-based learning, policy learning*, and *model-based policy learning*, over eight training periods, and with 1,000 new individuals per period.

Data Regimes. We evaluate each method in three *data regimes* shown in Fig. 1. Each regime assumes that the conditional probability of repayment Pr$(y = 1 \mid x)$ follows a specific functional form.: (1): LINEAR: Pr$(y = 1 \mid x)$ is a linear function of x. Here, x is Uniform$[-2, 2]$; (2): KINKED: Pr$(y = 1 \mid x)$ is a generic function of x; i.e., non-monotonic with multiple optima and crossing the profitability threshold multiple times. Here $x \sim$ Uniform$[-4, 4]$. $y = 1/(1 + \exp(-a))$, where $a = -(x - (x^3/3!)) + 1.4$; (3) KINKED CENSORED: This is the same as the kinked data regime, but with a region ($x > 2$) of the data distribution explicitly censored (unavailable) to the model in the first lending period. This simulates the failure modes that happen with a non-exploring approach, such as the *model-based* approach, when data may be missing over a region of feature values.

Training Protocol. We generate $n = 1000$ data points per period, and subject them to the training protocol detailed in Algorithm 1 in the Appendix.[1]

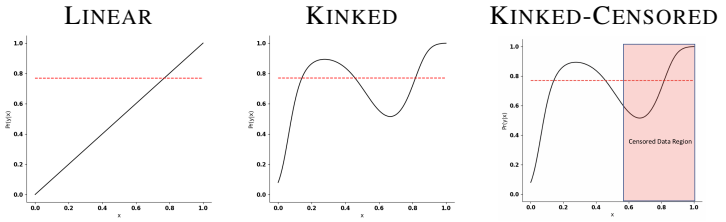

Figure 1: Overview of true model $Pr(y = 1|x)$ for different data regimes. Here x-axis is normalized to 1. The dotted horizontal red line is the profit threshold; i.e., the probability of repayment above which expected profit is non-negative.

	Linear			Kinked			Censored		
Metrics	Model	Policy	MBPol	Model	Policy	MBPol	Model	Policy	MBPol
Profit	$ 56	$ 55	$ 41	$ 65	$ 117	$ 117	$ 58	$ 120	$ 116
CAL	8.2%	N/A	22.8%	49.5%	N/A	23.4%	62.3%	N/A	24.1%
B_Acc	0.7	0.697	0.715	0.94	0.81	0.87	0.997	0.84	0.87

Table 1: Profit (on $1000 Principal), Calibration Error (CAL), and Balanced Accuracy (B_Acc) achieved by each method (MBPol = Model-based Policy) for three data regimes: Linear, Kinked, and Censored.

4.2 RESULTS

In Table 1,we show the performance of the policies learned for each method over each data regime. We see that the model-based policy learning approach, as well as the direct policy obtain near optimal profit, for each of the Linear and Kinked environments, respectively. This includes the setting where the data are censored, unlike the traditional model-based approach whose profits are suboptimal in the Kinked settings. Figure 2 shows that the model-based approach has a bigger recourse effort error in the Kinked setting, both with and without censoring, because it fails to explore over a sequence of multiple lending periods. Both the direct policy and the model-based policy handle this problem well. However, as Table 1 shows, model-based policy learning achieves a higher accuracy than the other approaches in the Kinked environment, meaning it enables better explainability. In the same vein, model-based policy learning has lower calibration error than the model approach in these complex settings.

5 CONCLUDING REMARKS

One of the present limitations of our recourse effort fairness metric is that it is lacking portability; i.e., it is not yet evaluated in an environment with multiple markets, for example, where there are different profitability thresholds due to multiple interest rates. A natural extension of this work, therefore, is to examine the model-based policy approach for a band of thresholds in which the actual threshold could fall. This is the subject of our ongoing research. For a learning approach to achieve *recourse effort fairness portability* across markets, it needs to learn an accurate model in the region where the profitability thresholds might fall. This guarantees that our definition of the recourse effort fairness, which relies on minimizing the recourse effort fairness error, as detailed in Section 3, will also be achieved in diverse markets.

[1]As explained there, the relevant hyperparameters are the learning rate η, the steepness of the softmax slope β, and the model-accuracy penalization rate γ. We set these through a grid-search with twenty data seeds and maximizing profit, which is arguably the most important desideratum for the lender.

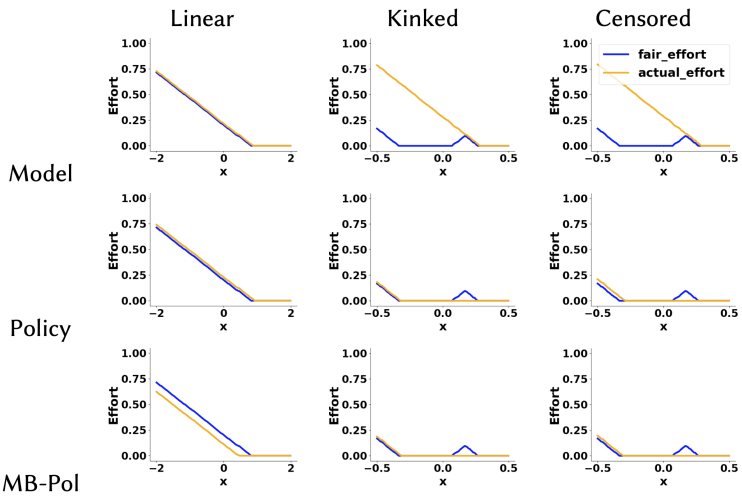

Figure 2: Actual effort juxtaposed against fair effort for different methods and data regimes, and for a representative seed, showing larger divergence for the model-based approach compared to policy and MBPolicy for the Kinked environment (whether censored or not).

REFERENCES

Elena Beretta, Antonio Vetro, and Juan Carlos Lepri, Bruno abd De Martin. Equality of opportunity in ranking: a fair-distributive model, 2021. URL https://nexa.polito.it/nexacenterfiles/bias2021_camera_ready.pdf.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *arXiv e-prints*, art. arXiv:1610.07524, Oct 2016.

Sam Corbett-Davies and Sharad Goel. The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning. *arXiv e-prints*, art. arXiv:1808.00023, Jul 2018.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Rich Zemel. Fairness Through Awareness. *arXiv e-prints*, art. arXiv:1104.3913, Apr 2011.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of Opportunity in Supervised Learning. *arXiv e-prints*, art. arXiv:1610.02413, Oct 2016.

Lily Hu and Yiling Chen. A Short-term Intervention for Long-term Fairness in the Labor Market. *arXiv e-prints*, art. arXiv:1712.00064, Nov 2017.

Niki Kilbertus, Manuel Gomez-Rodriguez, Bernhard Schölkopf, Krikamol Muandet, and Isabel Valera. Fair Decisions Despite Imperfect Predictions. *arXiv e-prints*, art. arXiv:1902.02979, Feb 2019.

A. Rupam Mahmood, Hado P van Hasselt, and Richard S Sutton. Weighted importance sampling for off-policy learning with linear function approximation. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3014–3022. Curran Associates, Inc., 2014.

J.E. Roemer and A Trannoy. Equality of opportunity. handbook of income distribution 2(2), 2015.

Naeem Siddiqi. *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*, volume 3. John Wiley & Sons, 2012.

Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, jan 2019. doi: 10.1145/3287560.3287566. URL https://doi.org/10.1145%2F3287560.3287566.

Isabel Valera, Adish Singla, and Manuel Gomez Rodriguez. Enhancing the accuracy and fairness of human decision making, 2018. URL https://arxiv.org/abs/1805.10318.

6 APPENDIX

6.1 MODEL-BASED POLICY LEARNING ALGORITHM

Input: Interest rate α, number of time steps T, number of loan applicants in each step N, number of iterations per step M, mini-batch ratio B, loan levels L, learning rate $\eta \geq 0$, model loss penalty rate $\gamma \geq 0$, stochasticity reduction $\beta \geq 1$.
Output: $\{\theta_t\}_{t=0}^{T}$, $\{\pi_{\theta_t}\}_{t=0}^{T}$
initialization;
$\theta_0 \leftarrow$ `InitializeParameters()`;
while $t \leq T$ **do**
  $\mathcal{D} \leftarrow$ `RunLendingCycle()` ;
  $\theta_{t+1} \leftarrow$ `UpdateModelandPolicy()`;
end
return $\{\theta_t\}_{t=0}^{T}$, $\{\pi_{\theta_t}\}_{t=0}^{T}$;
Function `RunLendingCycle`(L, N, θ, β)**:**
  $\mathcal{D} \leftarrow \varnothing$;
  for $i = 1...N$ **do**
    $(x_i, y_i) \sim P(x, y)$;
    $k_i \sim \pi_\theta(\beta, x_i)$;
    $l_i = L[k_i]$;
    if $l_i > 0$ **then**
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{x_i, a_i, k_i, y_i\}$;
    end
  end
  return $\mathcal{D}$;
;
Function `UpdateModelandPolicy`$(\theta^{'}, \mathcal{D}, M, B, \eta, \gamma)$**:**
  $\theta^0 \leftarrow \theta^{'}$;
  for $j = 1...M$ **do**
    $\mathcal{D}^j \leftarrow$ `MiniBatch`$(\mathcal{D}, B)$;
    $\nabla \leftarrow 0, n_j \leftarrow 0$;
    for $(x_i, k_i, y_i) \in \mathcal{D}^j$ **do**
      if $L[k_i] > 0$ **then**
        $n_j + +$;
        $\nabla \leftarrow \nabla + \eta\Big(\nabla_\theta u(\pi_\theta) - \gamma \nabla_\theta c(P_\theta)\Big)$;
      end
    end
    $\theta^{j+1} = \theta_j + \frac{\nabla}{n_j}$
  end
  return θ^M
;

Algorithm 1: Model-based policy learning

Table 2 summarizes the notation.

SYMBOL	MEANING
$y \in \{0, 1\}$	repayment indicator, $y = 1$ if loan will be repaid
$\boldsymbol{x} = (x_1, \ldots, x_d)$	vector of d features
$\alpha \geq 0$	interest rate
$t \in \{0, 1, \ldots, T\}$	time period
$k \in \{0, 1\}$	credit limit levels index set
$l \in \mathbb{R}$	credit limit amount
$\theta \in \mathbb{R}^d$	policy parameters
p	repayment probability $\Pr(y = 1 \vert \boldsymbol{x})$

Table 2: Notation

6.2 POLICY LEARNING

The lender learns a profit-maximizing policy directly from borrower and loan features without learning a repayment predictive model. This is the approach adopted by Kilbertus et al. (2019). This approach parametrizes the policy directly, $\pi_\theta(k = 1|x) = \sigma(\beta \cdot \theta^\top \phi(x)) \in (0, 1)$, where $\beta > 0$ is a parameter that adjusts the steepness of the logistic function, allowing us to control the desired degree of exploration.

The expected utility/profit (based on ground-truth distribution P) is:

$$u(\pi_\theta) = \mathbb{E}_{x,y\sim P, k\sim\pi_\theta(x)} \left[\mathbb{1}(k = 1)(\mathbb{1}(y = 1) \cdot \alpha \cdot l) - \mathbb{1}(y = 0) \cdot l] + [\mathbb{1}(k = 0) \cdot 0\right]$$

The *realized profit* from a loan decision is:

$$v_k(y_k) = \begin{cases} \alpha l & \text{if } k = 1, y_1 = 1 \\ -l & \text{if } k = 1, y_1 = 0 \\ 0 & \text{if } k = 0 \end{cases} \tag{4}$$

To compute the expected profit gradient of a given policy π_θ, and recognizing that the data observed in a lending period is from a distribution induced by the current policy, $\pi_{\theta_{t-1}}$, we use weighted importance sampling. In particular, we have:

$$\nabla_{\theta_t} u(\pi_{\theta_t}) = \mathbb{E}_{x,y\sim P_{\theta_{t-1}}} \left[\frac{\nabla_{\theta_t}\pi_{\theta_t}(k = 1|x)}{\pi_{\theta_{t-1}}(k = 1|x)} v_1(y_1)\right] \approx \frac{1}{n_{t-1}} \sum_{i=1}^{n_{t-1}} \left[\frac{\nabla_{\theta_t}\pi_{\theta_t}(k = 1|x_i)}{\pi_{\theta_{t-1}}(k = 1|x_i)} v_1(y_i)\right] \tag{5}$$

where n_{t-1} is the number of loans that were assigned a positive decision by the existing policy $\pi_{\theta t-1}$, i.e., the policy from period $t - 1$.

We need to compute the value of $\nabla_{\theta_t}\pi_{\theta_t}(k = 1|x_i)$ inside $equation$ 11. Reading the value of $\pi_\theta(k = 1|x)$ from $equation$ 8, and remembering that the derivative of the logistic sigmoid function can be written in terms of itself as: $\frac{\partial\sigma(z)}{\partial z} = \sigma(z)(1 - \sigma(z))$ [1] we can rewrite this gradient as:

$$\begin{aligned} \nabla_{\theta_t}\pi_{\theta_t}(k = 1|x_i) &= \nabla_\theta \sigma\Big(\beta \cdot \theta^\top \phi(x_i)\Big) \\ &= \beta \cdot \pi_{\theta_t}(k_i = 1|x_i) \cdot (1 - \pi_{\theta_t}(k_i = 1|x_i)) \cdot \phi(x_i) \end{aligned}$$

This gradient is used inside the UpdateModelandPolicy function in Algorithm , and can be written out in full as:

$$\nabla_{\theta_t} u(\pi_{\theta_t}) = \frac{1}{n_{t-1}} \sum_{i=1}^{n_{t-1}} \Bigg[\frac{\pi(k = 1|x_i) \cdot (1 - \pi(k = 1|x_i)}{\pi_{\theta_{t-1}}(k = 1|x_i)} v_1(y_i)) \cdot \beta \cdot \phi(x_i)\Bigg]$$

[1] https://dustinstansbury.github.io/theclevermachine/derivation-common-neural-network-activation-functions

This gradient is used to do the model update via SGA

One difference in implementation from what is shown in Algorithm 1 is that in Step 28, there's no penalization for model inaccuracy. That is:

$$\nabla \leftarrow \nabla + \eta \nabla_\theta u(\pi_\theta)$$

As in the *model-based policy learning* approach, the policy here is exploring(stochastic).

6.3 MODEL-BASED POLICY LEARNING

The lender learns both an accurate model and a profit-maximizing policy simultaneously with the learning primarily driven by profit-maximizing gradients for multiple thresholds.

Let's illustrate the with binary decisions $k \in \{0, 1\}$. We first parametrize the model for the repayment probability using a logistic (logit) function.

$$p(y = 1|x, k = 1) = \sigma(\theta^\top \phi(\boldsymbol{x})) \tag{6}$$

where $\sigma(a) = \frac{1}{1+\exp(-a)}$.

Denoting the probability that a loan is repaid as p, the *expected utility* v from a loan of size $l = 1$ is

$$v = p \cdot (\alpha l) + (1 - p) \cdot (-l) = l \cdot (p(\alpha + 1) - 1). \tag{7}$$

Assume, for ease of exposition, that $l = 1$. In a deterministic threshold approach we would assign a loan if this expected utility exceeds zero. In our model-based policy learning approach, we allocate loans using a logistic policy, defined as a logit function of the expected utility:

$$\pi_\theta(k = 1|x) = \sigma(\beta v) = \sigma\Big(\beta \cdot l \cdot \big(\sigma(\theta^\top \phi(x))(\alpha + 1) - 1\big)\Big) \in (0, 1) \tag{8}$$

where $\beta > 0$ is a parameter that adjusts the steepness of the logistic function, allowing us to control the desired degree of exploration.

In the single threshold case, there's only one interest rate α The *realized profit* from a loan decision is:

$$v_k(y_k) = \begin{cases} \alpha l & \text{if } k = 1, y_1 = 1 \\ -l & \text{if } k = 1, y_1 = 0 \\ 0 & \text{if } k = 0 \end{cases} \tag{9}$$

To compute the expected utility of a given policy π_θ, and recognizing that the data observed in a lending period is from a distribution induced by the current policy, $\pi_{\theta_{t-1}}$, we use weighted importance sampling (Mahmood et al., 2014). In particular, we have:

$$u(\pi_\theta) =$$

$$\mathbb{E}_{x,y\sim P_{\theta_{t-1}}, k\sim\pi_{\theta_t}(x)}\left[\frac{\mathbb{1}(k=0)\cdot 0}{\pi_{\theta_{t-1}}(k=0|x)} + \frac{\mathbb{1}(k=1)\cdot v_1(y_1)}{\pi_{\theta_{t-1}}(k=1|x)}\right]$$

$$= \mathbb{E}_{x,y\sim P_{\theta_{t-1}}}\left[\frac{\pi_{\theta_t}(k=0|x)\cdot 0}{\pi_{\theta_{t-1}}(k=0|x)} + \frac{\pi_{\theta_t}(k=1|x)\cdot v_1(y_1)}{\pi_{\theta_{t-1}}(k=1|x)}\right]$$

$$= \mathbb{E}_{x,y\sim P_{\theta_{t-1}}}\left[\frac{\pi_{\theta_t}(k=1|x)\cdot v_1(y_1)}{\pi_{\theta_{t-1}}(k=1|x)}\right] \tag{10}$$

where

$$P_{\theta_{t-1}} \propto P(y|x) \cdot \pi_{\theta_{t-1}}(k = 1|x) \cdot P(x)$$

We take the gradient of the utility function:

$$\begin{aligned}\nabla_{\theta_t} u(\pi_{\theta_t}) &= \mathbb{E}_{x,y\sim P_{\theta_{t-1}}}\left[\frac{\nabla_{\theta_t}\pi_{\theta_t}(k=1|x)}{\pi_{\theta_{t-1}}(k=1|x)} v_1(y_1)\right] \\ &\approx \frac{1}{n_{t-1}}\sum_{i=1}^{n_{t-1}}\left[\frac{\nabla_{\theta_t}\pi_{\theta_t}(k=1|x_i)}{\pi_{\theta_{t-1}}(k=1|x_i)} v_1(y_i)\right]\end{aligned} \tag{11}$$

where n_{t-1} is the number of loans that were assigned a positive decision by the existing policy $\pi_{\theta t-1}$, i.e., the policy from period $t-1$.

We need to compute the value of $\nabla_{\theta_t}\pi_{\theta_t}(k=1|x_i)$ inside $equation$ 11. Reading the value of $\pi_\theta(k=1|x)$ from $equation$ 8, we can rewrite this gradient as:

$$\begin{aligned}&\nabla_{\theta_t}\pi_{\theta_t}(k=1|x_i) \\ &= \nabla_\theta \sigma\Big(\beta\cdot l\cdot\big(\sigma(\theta^\top\phi(x_i)+\theta_{a_i})(\alpha+1)-1\big)\Big) \\ &= (\alpha+1)\cdot\pi_{\theta_t}(k_i=1|x_i)\cdot(1-\pi_{\theta_t}(k_i=1|x_i))\cdot\beta\cdot l\cdot\hat{p}(1-\hat{p})\cdot\phi(x_i)\end{aligned}$$

where $\hat{p}=\sigma(\theta^\top\phi(x_i))$.

This gradient is used to do the model update via SGA.

$$\nabla_{\theta_t} u(\pi_{\theta_t}) = \frac{1}{n_{t-1}}\sum_{i=1}^{n_{t-1}}\left[\frac{v_1(y_i)\cdot(\alpha+1)}{\pi_{\theta_{t-1}}(k=1|x_i)}\pi(k=1|x_i)\cdot(1-\pi(k=1|x_i))\cdot\beta\cdot l\cdot\hat{p}\cdot(1-\hat{p})\cdot\phi(x_i)\right]$$