Differentiable Learning Under Triage

Anonymous Author(s)
Affiliation
Address
email

Abstract

Multiple lines of evidence suggest that predictive models may benefit from algorithmic triage. Under algorithmic triage, a predictive model does not predict all instances but instead defers some of them to human experts. However, the interplay between the prediction accuracy of the model and the human experts under algorithmic triage is not well understood. In this work, we start by formally characterizing under which circumstances a predictive model may benefit from algorithmic triage. In doing so, we also demonstrate that models trained for full automation may be suboptimal under triage. Then, given any model and desired level of triage, we show that the optimal triage policy is a deterministic threshold rule in which triage decisions are derived deterministically by thresholding the difference between the model and human errors on a per-instance level. Building upon these results, we introduce a practical gradient-based algorithm that is guaranteed to find a sequence of predictive models and triage policies of increasing performance. Experiments on a wide variety of supervised learning tasks using synthetic and real data from two important applications—content moderation and scientific discovery—illustrate our theoretical results and show that the models and triage policies provided by our algorithm outperform those provided by several competitive baselines.

1 Introduction

In recent years, there have been a raising interest on a new learning paradigm which seeks the development of predictive models that operate under different automation levels—models that take decisions for a given fraction of instances and leave the remaining ones to human experts. This new paradigm has been so far referred to as learning under algorithmic triage [1], learning under human assistance [2, 3], learning to complement humans [4, 5], and learning to defer to an expert [6]. Here, one does not only has to find a predictive model but also a triage policy which determines who predicts each instance.

The motivation that underpins learning under algorithmic triage is the observation that, while there are high-stake tasks where predictive models have matched, or even surpassed, the average performance of human experts [7–9], they are still less accurate than human experts on some instances, where they make far more errors than average [1]. The main promise is that, by working together, human experts and predictive models are likely to achieve a considerably better performance than each of them would achieve on their own. While the above mentioned work has shown some success at fulfilling this promise, the interplay between the predictive accuracy of a predictive model and its human counterpart under algorithmic triage is not well understood.

One of the main challenges in learning under algorithmic triage is that, for each potential triage policy, there is an optimal predictive model, however, the triage policy is also something one seeks to optimize, as first noted by De et al. [2]. In this context, previous work on learning under algorithmic triage can be naturally differentiated into two lines of work. The first line of work has developed rather general heuristic algorithms that do not enjoy theoretical guarantees [1, 4, 5]. The second

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

has developed algorithm with theoretical guarantees [2, 3, 6], however, they have focused on more restrictive settings. More specifically, De et al. [2, 3] focuses on ridge regression and support vector machines and reduce the problem to the maximization of approximately submodular functions and Mozannar and Sontag [6] view the problem from the perspective of cost sensitive learning and introduce a convex and consistent surrogate loss of the objective function they consider.

Our contributions. Our starting point is a theoretical investigation of the interplay between the prediction accuracy of supervised learning models and human experts under algorithmic triage. By doing so, we hope to better inform the design of general purpose techniques for training differentiable models under algorithmic triage. Our investigation yields the following insights:

I. To find the optimal triage policy and predictive model, we need to take into account the amount of human expert disagreement, or expert uncertainty, on a per-instance level.

II. We identify under which circumstances a predictive model that is optimal under full automation may be suboptimal under a desired level of triage.

III. Given any predictive model and desired level of triage, the optimal triage policy is a deterministic threshold rule in which triage decisions are derived deterministically by thresholding the difference between the model and human errors on a per-instance level.

Building on the above insights, we introduce a practical gradient-based algorithm that finds a sequence of predictive models and triage policies of increasing performance subject to a constraint on the maximum level of triage. We apply our gradient-based algorithm in a wide variety of supervised learning tasks using both synthetic and real-world data from two important applications—content moderation and scientific discovery. Our experiments illustrate our theoretical results and show that the models and triage policies provided by our algorithm outperform those provided by several competitive baselines[1].

Further related work. Our work is also related to the areas of learning to defer and active learning. In learning to defer, the goal is to design machine learning models that are able to defer predictions [10–17]. Most previous work focuses on supervised learning and design classifiers that learn to defer either by considering the defer action as an additional label value or by training an independent classifier to decide about deferred decisions. However, in this line of work, there are no human experts who make predictions whenever the classifiers defer them, in contrast with the literature on learning under algorithmic triage [1–6]—they just pay a constant cost every time they defer predictions. Moreover, the classifiers are trained to predict the labels of all samples in the training set, as in full automation. In active learning, the goal is to find which subset of samples one should label so that a model trained on these samples predicts accurately any sample at test time [18–25]. In contrast, in our work, the trained model only needs to predict accurately a fraction of samples picked by the triage policy at test time and rely on human experts to predict the remaining samples.

2 Supervised Learning under Triage

Let $\mathcal{X} \subseteq \mathbb{R}^m$ be the feature domain, $\mathcal{Y}$ be the label domain, and assume features and labels are sampled from a ground truth distribution $P(\mathbf{x}, y) = P(\mathbf{x})P(y \,|\, \mathbf{x})$. Moreover, let $\hat{y} = h(\mathbf{x})$ be the label predictions provided by a human expert and assume they are sampled from a distribution $P(h \,|\, \mathbf{x})$, which models the disagreements amongst experts [26]. Then, in supervised learning under triage, one needs to find:

(i) a triage policy $\pi(\mathbf{x}) : \mathcal{X} \to \{0, 1\}$, which determines who predicts each feature vector—a supervised learning model ($\pi(\mathbf{x}) = 0$) or a human expert ($\pi(\mathbf{x}) = 1$);

(ii) a predictive model $m(\mathbf{x}) : \mathcal{X} \to \mathcal{Y}$, which needs to provide label predictions $\hat{y} = m(\mathbf{x})$ for those feature vectors $\mathbf{x}$ for which $\pi(\mathbf{x}) = 0$.

Here, similarly as in standard supervised learning, we look for the triage policy and the predictive model that result into the most accurate label predictions by minimizing a loss function $\ell(\hat{y}, y)$. More formally, let Π be the set of all triage policies, then, given a hypothesis class of predictive models $\mathcal{M}$,

[1] To facilitate research in this area, we will release an open-source implementation of our algorithm with the final version of the paper.

our goal is to solve the following minimization problem[2]:

$$\underset{\pi\in\Pi, m\in\mathcal{M}}{\text{minimize}} \quad L(\pi, m) \qquad \text{subject to} \quad \mathbb{E}_{\mathbf{x}}[\pi(\mathbf{x})] \leq b \tag{1}$$

where b is a given parameter that limits the level of triage, *i.e.*, the percentage of samples human experts need to provide predictions for, and

$$L(\pi, m) = \mathbb{E}_{\mathbf{x},y,h}\left[(1-\pi(\mathbf{x}))\,\ell(m(\mathbf{x}), y) + \pi(\mathbf{x})\,\ell(h, y)\right]. \tag{2}$$

Here, one might think of replacing h in the loss function $\ell(h, y)$ with its point estimate $\mu_h(\mathbf{x}) = \operatorname{argmin}_{\mu_h} \mathbb{E}_{h|\mathbf{x}}[\ell(h, \mu_h)]$. However, the resulting objective would have a bias term, as formalized by the following proposition[3]:

Proposition 1 *Let $\ell(\hat{y}, y)$ be a convex function with respect to $\hat{y}$ and assume there exist $\mathbf{x} \in \mathcal{X}$ for which the distribution of human predictions $P(h \mid \mathbf{x})$ is not a point mass. Then, the function*

$$\overline{L}(\pi, m) = \mathbb{E}_{\mathbf{x},y}\left[(1-\pi(\mathbf{x}))\,\ell(m(\mathbf{x}), y) + \pi(\mathbf{x})\,\ell(\mu_h(\mathbf{x}), y)\right]$$

is a biased estimate of the true average loss defined in Eq. 2.

The above result implies that, to find the optimal triage policy and predictive model, we need to take into account the amount of expert disagreement, or expert uncertainty, on each feature vector $\mathbf{x}$ rather than just an average expert prediction.

3 On the Interplay Between Prediction Accuracy and Triage

Let m_0^* be the optimal predictive model under full automation, *i.e.*, $m_0^* = \operatorname{argmin}_{m\in\mathcal{M}} L(\pi_0, m)$, where $\pi_0(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{X}$. Then, the following proposition tells us that, if the predictions made by m_0^* are less accurate than those by human experts on some instances, the model will always benefit from algorithmic triage:

Proposition 2 *If there is a subset $\mathcal{V} \subset \mathcal{X}$ of positive measure under P such that*

$$\int_{\mathbf{x}\in\mathcal{V}} \mathbb{E}_{y|\mathbf{x}}\left[\ell(m_0^*(\mathbf{x}), y)\right]\, dP > \int_{\mathbf{x}\in\mathcal{V}} \mathbb{E}_{y,h|\mathbf{x}}\left[\ell(h, y)\right]\, dP,$$

then there exists a nontrivial triage policy $\pi \neq \pi_0$ such that $L(\pi, m_0^*) < L(\pi_0, m_0^*)$.

Moreover, if we rewrite the average loss as

$$L(\pi, m) = \mathbb{E}_{\mathbf{x}}\big[(1-\pi(\mathbf{x}))\,\mathbb{E}_{y|\mathbf{x}}[\ell(m(\mathbf{x}), y)] + \pi(\mathbf{x})\,\mathbb{E}_{y,h|\mathbf{x}}\left[\ell(h, y)\right]\big],$$

it become apparent that, for any model $m \in \mathcal{M}$, the optimal triage policy $\pi_m^* = \operatorname{argmin}_{\pi\in\Pi} L(\pi, m)$ is a deterministic threshold rule in which triage decisions are derived by thresholding the difference between the model and human loss on a per-instance level. Formally, we have the following Theorem:

Theorem 3 *Let $m \in \mathcal{M}$ be any fixed predictive model. Then, the optimal triage policy that minimize the loss $L(\pi, m)$ subject to a constraint $\mathbb{E}_{\mathbf{x}}[\pi(\mathbf{x})] \leq b$ on the maximum level of triage is given by:*

$$\pi_{m,b}^*(\mathbf{x}) = \begin{cases} 1 & \textit{if } \mathbb{E}_{y|\mathbf{x}}\left[\ell(m(\mathbf{x}), y) - \mathbb{E}_{h|\mathbf{x}}\left[\ell(h, y)\right]\right] > t_{P,b,m} \\ 0 & \textit{otherwise,} \end{cases} \tag{3}$$

where $t_{P,b,m} = \underset{\tau\geq 0}{\textit{argmin}}\ \mathbb{E}_{\mathbf{x}}\big[\tau\, b + \max\left(\mathbb{E}_{y|\mathbf{x}}\left[\ell(m(\mathbf{x}), y) - \mathbb{E}_{h|\mathbf{x}}[\ell(h, y)]\right] - \tau,\, 0\right)\big]$.

Then, if we plug in Eq. 3 into Eq. 1, we can rewrite our minimization problem as:

$$\underset{m\in\mathcal{M}}{\text{minimize}} \quad L(\pi_{m,b}^*, m) \tag{4}$$

where

$$L(\pi_{m,b}^*, m) = \mathbb{E}_{\mathbf{x}}\big[\mathbb{E}_{y|\mathbf{x}}[\ell(m(\mathbf{x}), y)] - \textsc{Thres}_{t_{P,b,m}}\left(\mathbb{E}_{y|\mathbf{x}}\left[\ell(m(\mathbf{x}), y) - \mathbb{E}_{h|\mathbf{x}}[\ell(h, y)]\right], 0\right)\big] \tag{5}$$

[2] One might think that minimizing over the set of all possible triage policies is not a well-posed problem, *i.e.*, the optimal triage policy is an extremely complex function. However, our theoretical analysis will reveal that the optimal triage policy does have a simple form and its complexity depends on the considered hypothesis class of predictive models (refer to Theorem 3).

[3] All proofs can be found in Appendix A.

with $\textsc{Thres}_t(x, \text{val}) = \begin{cases} x & \text{if } x > t \\ \text{val} & \text{otherwise.} \end{cases}$

Here, note that, in the unconstrained case, $t_{P,b,m} = 0$ and $\textsc{Thres}_0(x, 0) = \max(x, 0)$. Next, building on the above expression, we can identify the circumstances under which the optimal predictive model under full automation $m_{\theta_0^*}$ within a hypothesis class of parameterized predictive models $\mathcal{M}(\Theta)$ is suboptimal under algorithmic triage. Formally, our main result is the following Proposition:

Proposition 4 *Let $m_{\theta_0^*}$ be the optimal predictive model under full automation within a hypothesis class of parameterized models $\mathcal{M}(\Theta)$, $\pi^*_{m_{\theta_0^*},b}$ the optimal triage policy for $m_{\theta_0^*}$ defined in Eq. 3 for a given maximum level of triage b, and $\mathcal{V} = \{\mathbf{x} \,|\, \pi^*_{m_{\theta_0^*},b}(\mathbf{x}) = 1\}$. If*

$$\int_{\mathbf{x} \in \mathcal{V}} \mathbb{E}_{y|\mathbf{x}} \left[\nabla_\theta \ell(m_\theta(\mathbf{x}), y)|_{\theta=\theta_0^*} \right] dP \neq \mathbf{0}. \tag{6}$$

then it holds that $L(\pi^*_{m_{\theta_0^*},b}, m_{\theta_0^*}) > \min_{\theta \in \Theta} L(\pi^*_{m_\theta,b}, m_\theta)$.

Finally, we can also identify the circumstances under which any predictive model $m_{\theta'}$ within a hypothesis class of parameterized predictive models $\mathcal{M}(\Theta)$ is suboptimal under algorithmic triage:

Proposition 5 *Let $m_{\theta'}$ be a predictive model within a hypothesis class of parameterized models $\mathcal{M}(\Theta)$, $\pi^*_{m_{\theta'},b}$ the optimal triage policy for $m_{\theta'}$ defined in Eq. 3 for a given maximum level of triage b, and $\mathcal{V} = \{\mathbf{x} \,|\, \pi^*_{m_{\theta'},b}(\mathbf{x}) = 0\}$. If*

$$\int_{\mathbf{x} \in \mathcal{V}} \mathbb{E}_{y|\mathbf{x}} \left[\nabla_\theta \ell(m_\theta(\mathbf{x}), y)|_{\theta=\theta'} \right] dP \neq \mathbf{0}. \tag{7}$$

then it holds that $L(\pi^*_{m_{\theta'},b}, m_{\theta'}) > \min_{\theta \in \Theta} L(\pi^*_{m_\theta,b}, m_\theta)$.

The above results will lay the foundations for our practical gradient-based algorithm for differentiable learning under triage in the next section.

4 How to Learn Under Triage

In this section, our goal is to find the policy m_{θ^*} within a hypothesis class of parameterized predictive models $\mathcal{M}(\Theta)$ that minimizes the loss $L(\pi^*_{m_\theta,b}, m_\theta)$ defined in Eq. 4.

To this end, we now introduce a general purpose gradient-based algorithm that first approximates m_{θ^*} given a desirable maximum level of triage b and then approximates the corresponding optimal triage policy $\pi^*_{m_{\theta^*},b}$[4]. To approximate m_{θ^*}, the main obstacle we face is that the threshold value t_{P,b,m_θ} in the average loss $L(\pi^*_{m_\theta,b}, m_\theta)$ given by Eq. 5 depends on the predictive model m_θ which we are trying to learn. To overcome this challenge, we proceed sequentially, starting from the triage policy π_0, with $\pi_0(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{X}$, and build a sequence of triage policies and predictive models $\{(\pi^*_{m_{\theta_t},b}, m_{\theta_t})\}_{t=0}^T$. More specifically, in each step t, we find the parameters of the predictive model m_{θ_t} via stochastic gradient descent (SGD) [27], *i.e.*,

$$\begin{aligned}
\theta_t^{(j)} &= \theta_t^{(j-1)} - \alpha^{(j-1)} \nabla_\theta \, L(\pi^*_{m_{\theta_{t-1}},b}, m_\theta)\Big|_{\theta=\theta_t^{(j-1)}} \\
&= \theta_t^{(j-1)} - \alpha^{(j-1)} \nabla_\theta \mathbb{E}_{\mathbf{x}} \big[\pi^*_{m_{\theta_{t-1}},b}(\mathbf{x}) \, \mathbb{E}_{y,h|\mathbf{x}} \left[\ell(h, y) \right] + (1 - \pi^*_{m_{\theta_{t-1}},b}(\mathbf{x})) \, \mathbb{E}_{y|\mathbf{x}} [\ell(m_\theta(\mathbf{x}), y)] \big] \Big|_{\theta=\theta_t^{(j-1)}} \\
&= \theta_t^{(j-1)} - \alpha^{(j-1)} \mathbb{E}_{\mathbf{x}} \big[(1 - \pi^*_{m_{\theta_{t-1}},b}(\mathbf{x})) \times \mathbb{E}_{y|\mathbf{x}} [\nabla_\theta \, \ell(m_\theta(\mathbf{x}), y)|_{\theta=\theta_t^{(j-1)}}] \big],
\end{aligned} \tag{8}$$

where $\alpha^{(j)}$ is the learning rate at iteration j. Moreover, the following proposition shows that, under mild conditions, the performance of the triage policies and predictive models improves in each step:

Proposition 6 *Assume that $\nabla^2_\theta \ell(m_\theta(\mathbf{x}), y) \preccurlyeq \Lambda \mathbb{I}$ for all $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$ and $\alpha^{(j)} < 1/\Lambda$ for all $j > 0$ for some constant $\Lambda > 0$. If, in each step t, we find the parameters of the predictive model m_{θ_t} using Eq. 8, with $\theta^{(0)} = \theta_{t-1}$, then, it holds that $L(\pi^*_{m_{\theta_{t-1}},b}, m_{\theta_{t-1}}) < L(\pi^*_{m_{\theta_t},b}, m_{\theta_t})$.*

[4] At test time, given a predictive model m_θ and an unseen sample $\mathbf{x}$, we cannot directly evaluate (or, more precisely, estimate using Monte-Carlo) the value of the optimal triage policy $\pi^*_{m_{\theta^*},b}(\mathbf{x})$, given by Eq. 3, since it depends on $\mathbb{E}_{y \,|\, \mathbf{x}}[\cdot]$ and $\mathbb{E}_{h \,|\, \mathbf{x}}[\cdot]$.

In practice, given a set of samples $\mathcal{D} = \{(\mathbf{x}_i, y_i, h_i)\}$, we can use the following finite sample Monte-Carlo estimator for the gradient $\nabla_\theta L(\pi^*_{m_{\theta_{t-1}},b}, m_\theta)$:

$$\nabla_\theta L(\pi^*_{m_{\theta_{t-1}},b}, m_\theta) = \nabla_\theta \left[\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \ell(m_\theta(\mathbf{x}_i), y_i) - \textsc{Thres}_{t_{P,b,m_{\theta_{t-1}}}} \left(\ell(m_\theta(\mathbf{x}_i), y_i) - \ell(h_i, y_i), 0 \right) \right]$$
$$= \frac{1}{|\mathcal{D}|} \sum_{i=1}^{\max(\lceil (1-b)\,|\mathcal{D}| \rceil, p)} \nabla_\theta \ell(m_\theta(\mathbf{x}_{[i]}), y_{[i]})$$

where $\cdot_{[i]}$ denotes the i-th sample in increasing value of the difference between the model and the human loss[5] $\ell(m_{\theta_{t-1}}(\mathbf{x}_{[i]}), y_{[i]}) - \ell(h_{[i]}, y_{[i]})$ and p is the number of samples with $\ell(m_{\theta_{t-1}}(\mathbf{x}_{[i]}), y_{[i]}) - \ell(h_{[i]}, y_{[i]}) < 0$.

In the above, we do not have to explicitly compute the threshold $t_{P,b,m_{\theta_{t-1}}}$ nor the triage policy $\pi^*_{m_{\theta_{t-1}},b}(\mathbf{x}_i)$ for every sample $\mathbf{x}_i$ in the set $\mathcal{D}$, we just need to pick the $\max(\lceil (1-b)\,|\mathcal{D}| \rceil, p)$ samples with the lowest value of the model loss minus the human loss $\ell(m_{\theta_{t-1}}(\mathbf{x}_{[i]}), y_{[i]}) - \ell(h_{[i]}, y_{[i]})$ using the predictive model $m_{\theta_{t-1}}$ fitted in step $t-1$. To understand why, note that, as long as $t_{P,b,m_{\theta_{t-1}}} > 0$, by definition, $t_{P,b,m_{\theta_{t-1}}}$ needs to satisfy that

$$\frac{d}{d\tau} \left[\sum_{i \in \mathcal{D}} [\tau b + \max(\ell(m_{\theta_{t-1}}(\mathbf{x}_i), y_i) - \ell(h_i, y_i) - \tau, 0)] \right] \Bigg|_{\tau = t_{P,b,m_{\theta_{t-1}}}} = 0$$

and this can only happens if $\ell(m_{\theta_{t-1}}(\mathbf{x}_i), y_i) - \ell(h_i, y_i) - t_{P,b,m_{\theta_{t-1}}} > 0$ for $\lfloor b\,|\mathcal{D}| \rfloor$ out of $|\mathcal{D}|$ samples. Here, we are implicitly estimating the optimal triage policies using the observed training labels and human predictions—we are not approximating them using a parameterized model—and, due to Proposition 6, the implementation of the above procedure with Monte-Carlo estimates is guaranteed to converge to a local minimum of the empirical loss. Moreover, note that we can think of the procedure as a particular instance of disciplined parameterized programming [28, 29], where the differentiable convex optimization layer is given by the minimization with respect to the triage policy.

While training each of the predictive models m_{θ_t}, we can implicitly compute the optimal triage policy $\pi^*_{\theta_t,b}$ as described above, however, at test time, we cannot do the same since we would need to observe the label and human prediction of each unseen sample $\mathbf{x}$. To overcome this, after training the last machine model m_{θ_T}, we also fit a model $\hat{\pi}_\gamma(\mathbf{x})$ to approximate $\pi^*_{m_{\theta_T},b}(\mathbf{x})$ using SGD, *i.e.*,

$$\gamma^{(j)} = \gamma^{(j-1)} - \alpha^{(j-1)} \nabla_\gamma \left[\sum_{i=1}^{|\mathcal{D}|} \ell'(\hat{\pi}_\gamma(\mathbf{x}_i), \pi^*_{m_{\theta_T},b}(\mathbf{x}_i)) \right] \Bigg|_{\gamma = \gamma^{(j-1)}}, \tag{9}$$

where $\alpha^{(j)}$ is the learning rate at iteration j and the choice of loss $\ell'(\cdot)$ depends on the model class chosen for $\hat{\pi}_\gamma$. In our experiments, we have found that, using the above procedure, we can approximate well the optimal triage policy $\pi_{m_{\theta_T},b}$. However, we would like to note that this problem can also be viewed as finding an estimator for the α-superlevel set $C^\alpha(f) = \{\mathbf{x} \in \mathcal{X} : f(\mathbf{x}) \geq \alpha\}$ of the function $f(\mathbf{x}) = \mathbb{E}_{y \mid \mathbf{x}}[\ell(m_{\theta_T}(\mathbf{x}), y) - \mathbb{E}_{h \mid \mathbf{x}}[\ell(h(\mathbf{x}), y)]]$ with $\alpha = t_{P,b,m_{\theta_T}}$ from a set of noisy observations. Under this view, it might be possible to derive estimators with error performance bounds building upon recent work on level set estimation [30, 31]. This is left as future work.

Refer to Appendix B for a pseudocode implementation of the overall gradient-based algorithm, which returns θ_T and γ, and a detailed scalability analysis, which suggests that our algorithm does not significantly increase the computational complexity of vanilla SGD.

5 Experiments on Synthetic Data

In this section, our goal is to shed light on the theoretical results from Section 3. To this end, we use our gradient-based algorithm in a simple regression task in which the optimal predictive model under full automation is suboptimal under algorithmic triage.[6]

[5] Note that, if the set of samples contains several predictions $h_{[i]}$ by different human experts for each sample $\mathbf{x}_{[i]}$, we would use all of them to estimate the (average) human loss.

[6] All algorithms were implemented in Python 3.7 and ran on a V100 Nvidia Tesla GPU with 32GB of memory.

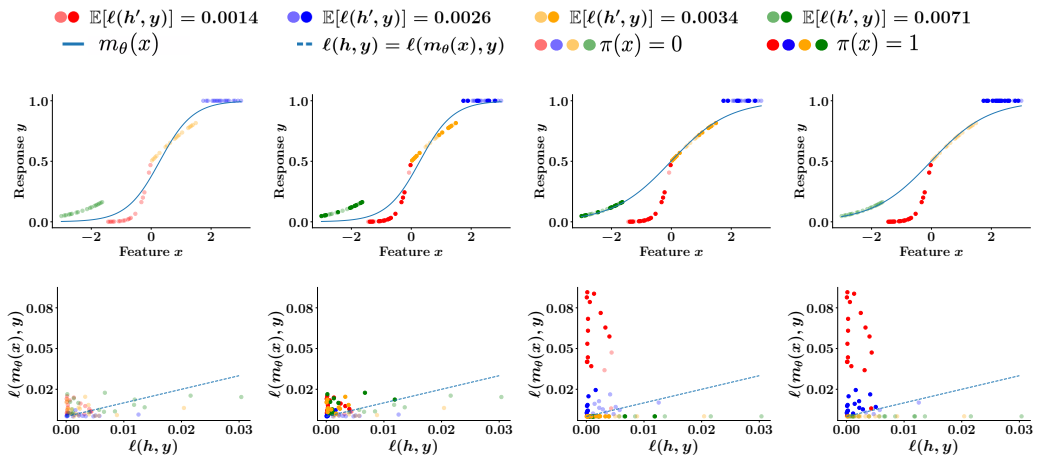


Figure 1: Interplay between the per-instance accuracy of predictive models and experts under different triage policies. In both panels, the first row shows the training samples (x, y) along with the predictions made by the models $m(x)$ and the triage policy values $\pi(x)$ and the second row shows the predictive model loss $\ell(m(x), y)$ against the human expert loss $\ell(m(x), y)$ on a per-instance level. The triage policy $\pi^*_{m_{\theta_0},b}$ is optimal for the predictive model m_{θ_0} and the triage policy $\pi^*_{m_\theta,b}$ is optimal for the predictive model m_θ. Each point corresponds to one instance and, for each instance, the color indicates the amount of noise in the predictions by experts, as given by Eq. 10, and the tone indicates the triage policy value. In all panels, we used $\ell(\hat{y}, y) = (\hat{y} - y)^2$ and the class of predictive models parameterized by sigmoid functions, *i.e.*, $m_\theta(x) = S_\theta(x)$.

Experimental setup. We generate $|\mathcal{D}| = 72$ samples, where we first draw the features $x \in \mathbb{R}$ uniformly at random, *i.e.*, $x \sim \mathrm{U}[-3, 3]$, and then obtain the response variables y using two different sigmoid functions $S_\theta(x) = \frac{1}{1+\exp(-\theta x)}$. More specifically, we set $y = S_1(x)$ if $x \in [-3, -1.5) \cup [0, 1.5)$ and $y = S_5(x)$ if $x \in [-1.5, 0) \cup [1.5, 3]$. Moreover, we assume human experts provide noisy predictions of the response variables, *i.e.*, $h(x) = y + \epsilon(x)$, where $\epsilon(x) \sim \mathcal{N}(0, \sigma^2_\epsilon(x))$ with

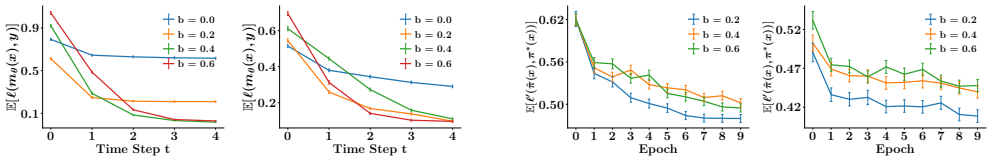
$$\sigma^2_\epsilon(\mathbf{x}) = \begin{cases} 8 \times 10^{-3} & \text{if} \quad x \in [-3, -1.5) \\ 1 \times 10^{-3} & \text{if} \quad x \in [-1.5, 0) \\ 4 \times 10^{-3} & \text{if} \quad x \in [0, 1.5) \\ 2 \times 10^{-3} & \text{if} \quad x \in [1.5, 3] \end{cases} \tag{10}$$

In the above, we are using heteroscedastic noise motivated by multiple lines of evidence that suggest that human experts performance on a per instance level spans a wide range [1, 2, 26]. Then, we consider the hypothesis class of predictive models $\mathcal{M}(\Theta)$ parameterized by sigmoid functions, *i.e.*, $m_\theta(x) = S_\theta(x)$, and utilize the sum of squared errors on the predictions as loss function, *i.e.*, $\ell(\hat{y}, y) = (\hat{y} - y)^2$, to train the following models and triage policies:

1. Predictive model trained under full automation m_{θ_0} without algorithmic triage, *i.e.*, $\pi_{m_{\theta_0},b}(\mathbf{x}) = \pi_0(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{X}$.
2. Predictive model trained under full automation m_{θ_0} with optimal algorithmic triage $\pi^*_{m_{\theta_0},b}$.
3. Predictive model trained under algorithmic triage m_θ, with $b = 1$, with suboptimal algorithmic triage $\pi^*_{m_{\theta_0},b}$. Here, we use the triage policy that is optimal for the predictive model trained under full automation.
4. Predictive model trained under algorithmic triage m_θ, with $b = 1$, with optimal algorithmic triage $\pi^*_{m_\theta,b}$.

In all the cases, we train the predictive models m_{θ_0} and m_θ using our method with $b = 0$ and $b = 1$, respectively. Finally, we investigate the interplay between the accuracy of the above predictive models and the human experts and the structure of the triage policies at a per-instance level.

Results. Figure 1 shows the training samples (x, y) along with the predictions made by the predictive models m_{θ_0} and m_θ and the values of the triage policies $\pi_0(x)$, $\pi^*_{m_{\theta_0},b}$ and $\pi^*_{m_\theta,b}$, as well as the



Hatespeech Galaxy zoo Hatespeech Galaxy zoo

(a) Training losses by the predictive models m_{θ_t} (b) Training losses by the triage policies $\hat{\pi}_{\gamma^{(i)}}$

Figure 2: Average training losses achieved by the predictive models m_{θ_t} and the triage policies $\hat{\pi}_{\gamma^{(i)}}$ on the Hatespeech and Galaxy zoo datasets during training. In Panel (a), each predictive model m_{θ_t} is the output of TRAINMODEL$(\cdot)$ at step t and in Panel (b), each triage policy $\hat{\pi}_{\gamma^{(i)}}$ is the output of TRAINTRIAGE$(\cdot)$ at epoch i. Both functions are defined in Appendix B where we provide the pseudocode of our Algorithm. Error bars correspond to plus and minus one standard error.

losses achieved by the models and triage policies (1-4) on a per-instance level. The results provide several interesting insights.

Since the predictive model trained under full automation m_{θ_0} seeks to generalized well across the entire feature space, the loss it achieves on a per-instance level is never too high, but neither too low, as shown in the left column of Panel (a). As a consequence, this model under no triage achieves the highest average loss among all alternatives, $L(\pi_0, m_{\theta_0}) = 0.5377$. This may not come as a surprise since the mapping between feature and response variables does not lie within the hypothesis class of predictive models used during training. However, since the predictions by human experts are more accurate than those provided by the above model in some regions of the feature space, we can deploy the model with the optimal triage policy $\pi^*_{\theta_0,b}$ given by Theorem 3 and lower the average loss to $L(\pi^*_{\theta_0,b}, m_{\theta_0}) = 0.2065$, as shown in the right column of Panel (a) and suggested by Proposition 2.

In contrast with the predictive model trained under full automation m_{θ_0}, the predictive model trained under triage m_θ learns to predict very accurately the instances that lie in the regions of the feature space colored in green and yellow but it gives up on the regions colored in red and blue, where its predictions incur a very high loss, as shown in Panel (b). However, these latter instances where the loss would have been the highest if the predictive model had to predict their response variables y are those that the optimal triage policy $\pi^*_{m_\theta,b}$ hand in to human experts to make predictions. As a result, this predictive model under the optimal triage policy does achieve the lowest average loss among all alternatives, $L(\pi^*_{\theta,b}, m_\theta) = 0.0964$, as suggested by Propositions 4 and 5.

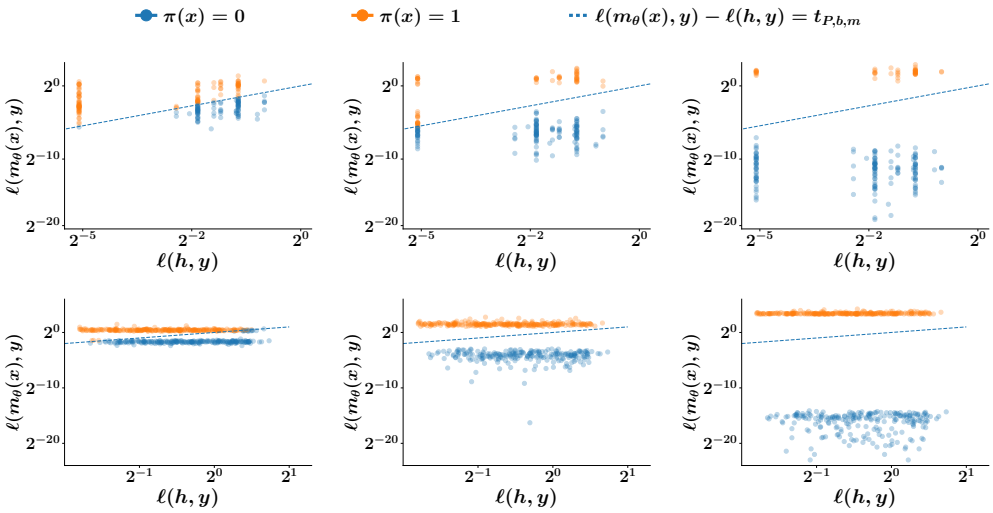
Finally, our results also show that deploying the predictive model m_θ under a suboptimal triage policy may actually lead to a higher loss $L(\pi^*_{\theta,b}, m_\theta) = 0.3111$ than the loss achieved by the predictive model trained under full automation m_{θ_0} with its optimal triage policy $\pi^*_{\theta_0,b}$. This happens because the predictive model m_θ is trained to work well *only* on the instances x such that $\pi^*_{m_\theta,b}(x) = 0$ and not necessarily on those with $\pi^*_{m_0,b}(x) = 0$.

6 Experiments on Real Data

In this section, we use our gradient-based algorithm in two binary and multi-class classification tasks in content moderation and scientific discovery,. We first investigate the interplay between the accuracy of the predictive models and human experts and the structure of the optimal triage policies at different steps of the training process. Then, we compare the performance of our algorithm with several competitive baselines.

Experimental setup. We use two publicly available datasets [32, 33], one from an application in content moderation and the other for scientific discovery[7]:

[7] We chose these two particular applications because the corresponding datasets are among the only publicly available datasets that we found containing multiple human predictions per instance, necessary to estimate the human loss at an instance level, and a relatively large number of instances. The Hatespeech dataset is publicly available under MIT license and the Galaxy zoo dataset is publicly available under Creative Commons Attribution-Noncommercial-No Derivative Works 2.0 license. Since our algorithm is general, it may be useful in other applications in which predictive models trained under full automation performs worse than humans in some instances (see Proposition 2).



(a) Step $t = 1$ (b) Step $t = 2$ (c) Step $t = 3$

Figure 3: Predictive model and expert losses at a per-instance level on a randomly selected subset of 500 samples of the Hatespeech (top row) and Galaxy zoo (bottom row) datasets throughout the execution of our method during training. The maximum level of triage is set to $b = 0.4$ for Hatespeech and $b = 1.0$ for Galaxy zoo dataset. Each point corresponds to an individual instance and, for each instance, the color pattern indicates the triage policy value.

— *Hatespeech:* It consists of $|\mathcal{D}| = 24{,}783$ tweets containing lexicons used in hate speech. Each tweet is labeled by three to five human experts from Crowdflower as "hate-speech", "offensive", or "neither".

— *Galaxy zoo:* It consists of $|\mathcal{D}| = 10{,}000$ galaxy images[8]. Each image is labeled by 30+ human experts as "early type" or "spiral".

For each tweet in the Hatespeech dataset, we first generate a 100 dimensional feature vector using fasttext [34] as $\mathbf{x}$, similarly as in De et al. [2]. For each image in the Galaxy zoo dataset, we use its corresponding pixel map[9] as $\mathbf{x}$. Given an instance with feature value $\mathbf{x}$, we estimate $P(h \,|\, \mathbf{x}) = \frac{n_{\mathbf{x}}(h)}{\sum_{h' \in \mathcal{Y}} n_{\mathbf{x}}(h')}$, where $n_{\mathbf{x}}(h)$ denotes the number of human experts who predicted label h and we set its true label to $y = \operatorname{argmax}_{h \in \mathcal{Y}} P(h \,|\, \mathbf{x})$. Moreover, at test time, for each instance that the triage policy assigns to humans, we sample $h \sim P(h \,|\, \mathbf{x})$.

In all our experiments, we consider the hypothesis class of probabilistic predictive models $\mathcal{M}(\Theta)$ parameterized by softmax distributions, *i.e.*,

$$m_\theta(\mathbf{x}) \sim p_{\theta;\mathbf{x}} = \text{Multinomial}\left(\left[\exp\left(\phi_{y,\theta}(\mathbf{x})\right)\right]_{y \in \mathcal{Y}}\right),$$

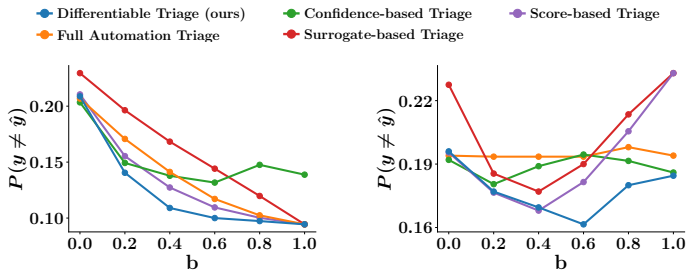
where, for the Hatespeech dataset, $\phi_{\bullet,\theta}$ is the convolutional neural network (CNN) by Kim [35] and, for the Galaxy zoo dataset, it is the deep residual network by He et al. [36]. During training, we use a cross entropy loss on the observed labels, *i.e.*, $\ell(\hat{y}, y) = -\log P(\hat{y} = y \,|\, \mathbf{x})$. Here, if an instance is assigned to the predictive model, we have that

$$P(\hat{y} = y \,|\, \mathbf{x}) = \frac{\exp\left(\phi_{y,\theta}(\mathbf{x})\right)}{\sum_{y'} \exp\left(\phi_{y',\theta}(\mathbf{x})\right)},$$

and, if an instance is assigned to a human expert, we have that $P(\hat{y} = y \,|\, \mathbf{x}) = \log P(h = y \,|\, \mathbf{x})$. For the function $\hat{\pi}_\gamma(\mathbf{x})$, we use the class of logistic functions, *i.e.*, $\hat{\pi}_\gamma(\mathbf{x}) = \frac{1}{1+\exp(-\phi_\gamma(\mathbf{x}))}$, where ϕ_γ is the same CNN and deep residual network as in the predictive model, respectively. Here, we also use the cross entropy loss, *i.e.*, $\ell'(\hat{\pi}_\gamma(\mathbf{x}), \pi^*_{m_{\theta_T},b}(\mathbf{x})) = -\pi^*_{m_{\theta_T},b}(\mathbf{x}) \log \hat{\pi}_\gamma(\mathbf{x}) - (1 - \pi^*_{m_{\theta_T},b}(\mathbf{x})) \log(1 - \hat{\pi}_\gamma(\mathbf{x}))$. In each experiment, we used 60% samples for training, 20% for validation and 20% for testing. Refer to Appendix C for additional details on the experimental setup.

[8] The original Galaxy zoo dataset consists of 61,577 images, however, we report results on a randomly chosen subset of 10,000 images due to scalability reasons. We found similar results in other random subsets.

[9] The pixel maps for each image are available at `https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge`



(a) Hatespeech (b) Galaxy zoo

Figure 4: Misclassification test error $P(\hat{y} \neq y)$ against the triage level b on the Hatespeech and Galaxy zoo datasets for our algorithm, confidence-based triage [5], score-based triage [1], surrogate-based triage [6] and full automation triage. Appendix C contains more details on the baselines.

Results. First, we look at the average loss achieved by the predictive models m_{θ_t} and triage policies $\hat{\pi}_{\gamma^{(i)}}$ throughout the execution of our method during training. Figure 2 summarizes the results, which reveal several insights. For small values of the triage level, the models m_{θ_t} aim to generalize well across a large portion of the feature space. As a result, they incur a large training loss, as shown in Panel (a). In contrast, for $b \geq 0.4$, the models m_{θ_t} are trained to generalize across a smaller region of the feature space, which leads to a considerably smaller training loss. However, for such a high triage level, the overall performance of our method is also contingent on how well $\hat{\pi}_\gamma$ approximates the optimal triage policy. Fortunately, Panel (b) shows that, as epochs increase, the average training loss of $\hat{\pi}_{\gamma^{(j)}}$ decreases. Appendix D validates further the trained approximate triage policies $\hat{\pi}_\gamma$.

Next, we compare the predictive model and the human expert losses per training instance throughout the execution of the our method during training. Figure 3 summarizes the results. At each step t, we find that the optimal triage policies $\pi^*_{m_{\theta_t},b}$ hands in to human experts those instances (in orange) where the loss would have been the highest if the predictive model had to predict their response variables y. Moreover, at the beginning of the training process (*i.e.*, low step values t), since the predictive model m_{θ_t} seeks to generalize across a large portion of the feature space, the model loss remains similar across the feature space. However, later into the training process (*i.e.*, high step values t), the predictive models m_{θ_t} focuses on predicting more accurately the samples that the triage policy hands in to the model, achieving a lower loss on those samples, and gives up on the remaining samples, where it achieves a high loss.

Finally, we compare the performance of our method against four baselines in terms of test misclassification error $P(\hat{y} \neq y)$. Refer to Appendix C for more details on the baselines, which we refer to as confidence-based triage [5], score-based triage [1], surrogate-based triage [6] and full automation triage[10] . Figure 4 summarizes the results, which show that the predictive models and triage policies found by our algorithm outperform the baselines across the majority of maximum triage levels b.

7 Conclusions

In this paper, we have contributed towards a better understanding of supervised learning under algorithmic triage. We have first identified under which circumstances predictive models may benefit from algorithmic triage, including those trained for full automation. Then, given a predictive model and desired level of triage, we have shown that the optimal triage policy is a deterministic threshold rule in which triage decisions are derived deterministically from the model and human per-instance errors. Finally, we have introduced a practical algorithm to train supervised learning models under triage and have shown that it outperforms several competitive baselines.

Our work also opens many interesting venues for future work. For example, we have assumed that each instance is predicted by either a predictive model or a human expert. However, there may be many situations in which human experts predict all instances but their predictions are informed by a predictive model [37]. We have shown that our algorithm is guaranteed to converge to a local minimum of the empirical risk, however, it would be interesting to analyze the convergence rate and the generalization error. Finally, it would be valuable to assess the performance of supervised learning models under algorithmic triage using interventional experiments on a real-world application.

[10] Among all the baselines we are aware of [1–6], we did not compare with De et al. [2, 3] because they only allow linear models and SVMs and we did not compare with Wilder et al. [4] because they did not provide enough details to implement their method.

References

[1] Maithra Raghu, Katy Blumer, Greg Corrado, Jon Kleinberg, Ziad Obermeyer, and Sendhil Mullainathan. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220*, 2019.

[2] Abir De, Paramita Koley, Niloy Ganguly, and Manuel Gomez-Rodriguez. Regression under human assistance. In *AAAI*, 2020.

[3] Abir De, Nastaran Okati, Ali Zarezadeh, and Manuel Gomez-Rodriguez. Classification under human assistance. In *AAAI*, 2021.

[4] Bryan Wilder, Eric Horvitz, and Ece Kamar. Learning to complement humans. In *IJCAI*, 2020.

[5] Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S. Weld. Optimizing AI for Teamwork. In *AAAI*, 2021.

[6] Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *ICML*, 2020.

[7] Justin Cheng, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. Antisocial behavior in online discussion communities. In *ICWSM*, 2015.

[8] Michael Pradel and Koushik Sen. Deepbugs: A learning approach to name-based bug detection. In *OOPSLA*, 2018.

[9] Eric Topol. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25(1):44, 2019.

[10] Peter Bartlett and Marten Wegkamp. Classification with a reject option using a hinge loss. *JMLR*, 9(Aug):1823–1840, 2008.

[11] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *ALT*, 2016.

[12] Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. 2019.

[13] Harish Ramaswamy, Ambuj Tewari, and Shivani Agarwal. Consistent algorithms for multiclass classification with an abstain option. *Electronic J. of Statistics*, 12(1):530–554, 2018.

[14] Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *ICML*, 2019.

[15] Ziyin Liu, Zhikang Wang, Paul Pu Liang, Russ R Salakhutdinov, Louis-Philippe Morency, and Masahito Ueda. Deep gamblers: Learning to abstain with portfolio theory. In *NeurIPS*, 2019.

[16] Sunil Thulasidasan, Tanmoy Bhattacharya, Jeff Bilmes, Gopinath Chennupati, and Jamal Mohd-Yusof. Combating label noise in deep learning using abstention. *arXiv preprint arXiv:1905.10964*, 2019.

[17] Liu Ziyin, Blair Chen, Ru Wang, Paul Pu Liang, Ruslan Salakhutdinov, Louis-Philippe Morency, and Masahito Ueda. Learning not to learn in the presence of noisy labels. *arXiv preprint arXiv:2002.06541*, 2020.

[18] David Cohn, Zoubin Ghahramani, and Michael Jordan. Active learning with statistical models. In *NeurIPS*, 1995.

[19] Steven Hoi, Rong Jin, Jianke Zhu, and Michael R Lyu. Batch mode active learning and its application to medical image classification. In *ICML*, 2006.

[20] Masashi Sugiyama. Active learning in approximately linear regression based on conditional expectation of generalization error. *JMLR*, 7(Jan):141–166, 2006.

[21] Rebecca Willett, Robert Nowak, and Rui M Castro. Faster rates in regression via active learning. In *NeurIPS*, 2006.

[22] Yuhong Guo and Dale Schuurmans. Discriminative batch mode active learning. In *NeurIPS*, 2008.

[23] Sivan Sabato and Remi Munos. Active regression by stratification. In *NeurIPS*, 2014.

[24] Xue Chen and Eric Price. Active regression via linear-sample sparsification. In *COLT*, 2017.

[25] Abolfazl Hashemi, Mahsa Ghasemi, Haris Vikalo, and Ufuk Topcu. Submodular observation selection and information gathering for quadratic models. In *ICML*, 2019.

[26] Maithra Raghu, Katy Blumer, Rory Sayres, Ziad Obermeyer, Bobby Kleinberg, Sendhil Mullainathan, and Jon Kleinberg. Direct uncertainty prediction for medical second opinions. In *ICML*, 2019.

[27] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[28] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *ICML*, 2017.

[29] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and Zico Kolter. Differentiable convex optimization layers. In *NeurIPS*, 2019.

[30] Rebecca M Willett and Robert D Nowak. Minimax optimal level-set estimation. *IEEE Transactions on Image Processing*, 16(12):2965–2979, 2007.

[31] Aarti Singh. *Nonparametric Set Estimation Problems in Statistical Inference and Learning*. PhD thesis, University of Wisconsin–Madison, 2008.

[32] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. ICWSM, pages 512–515.

[33] Steven P Bamford, Robert C Nichol, Ivan K Baldry, Kate Land, Chris J Lintott, Kevin Schawinski, Anže Slosar, Alexander S Szalay, Daniel Thomas, Mehri Torki, et al. Galaxy zoo: the dependence of morphology and colour on environment. *Monthly Notices of the Royal Astronomical Society*, 393(4):1324–1352, 2009.

[34] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

[35] Yoon Kim. Convolutional neural networks for sentence classification, 2014.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[37] Brian Lubars and Chenhao Tan. Ask not what ai can do, but what ai should do: Towards a framework of task delegability. 2020.

[38] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[39] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 7.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Our work contributes to empower, rather than replace, humans with machine learning. In particular, our results contribute towards developing general purpose techniques for training differentiable models under triage. More broadly, we do believe that machine learning algorithms that are optimized to operate under different automation levels may have a positive societal impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] All proofs can be found in Appendix A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We have included the code in the supplementary material; also, see Appendix B.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We provide additional details about the experimental setup in the Appendix C.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We provide standard-error bars in Figure 2.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Footnote 6.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See Footnote 7. The Hatespeech dataset is publicly available under MIT license. The Galaxy zoo dataset is publicly available under Creative Commons Attribution-Noncommercial-No Derivative Works 2.0 license.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The data does not contain personally identifiable information. The dataset Hatespeech contains examples of hate speech.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]