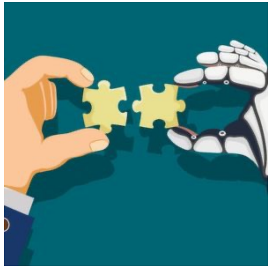


Learning under algorithmic triage

Manuel Gomez Rodriguez

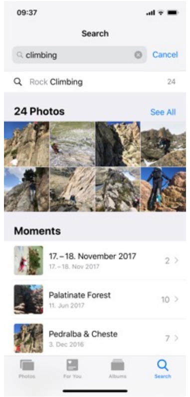
Includes joint work with Nastaran Okati, Abir De, Paramita Koley, Ali Zarezade and Niloy Ganguly

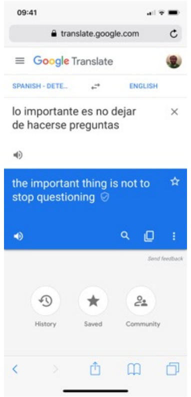
MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

Most stunning developments in computing

Machine learning (ML) has *taught* machines to...

recognize images | **translate between languages** | **play complex games**

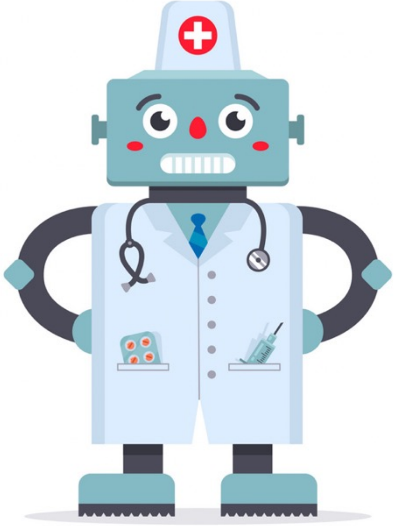




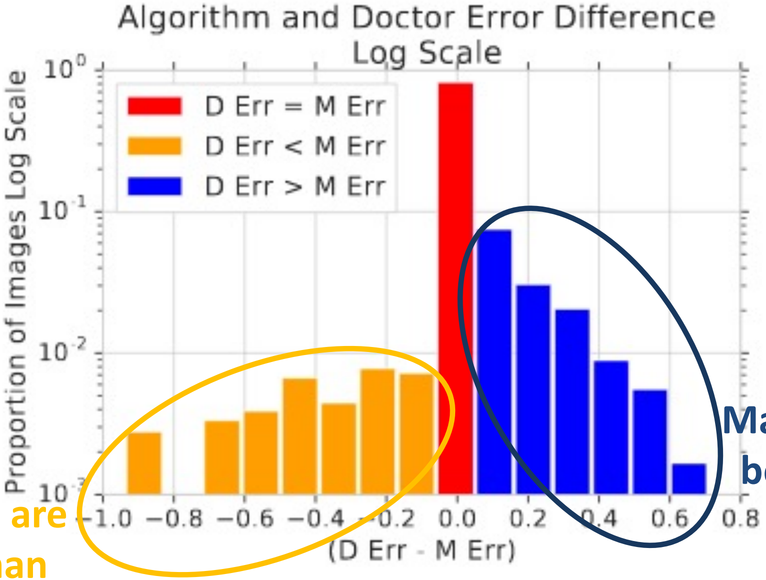
...to name a few

Machines achieve, or surpass, human performance at tasks for which intelligence is required

Machines learning is sometimes worse than humans

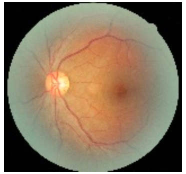


On **<u>some instances</u>, machine predictions** are still **worse** than **predictions** made by **human experts**



Machines are worse than humans

Machines are better than humans

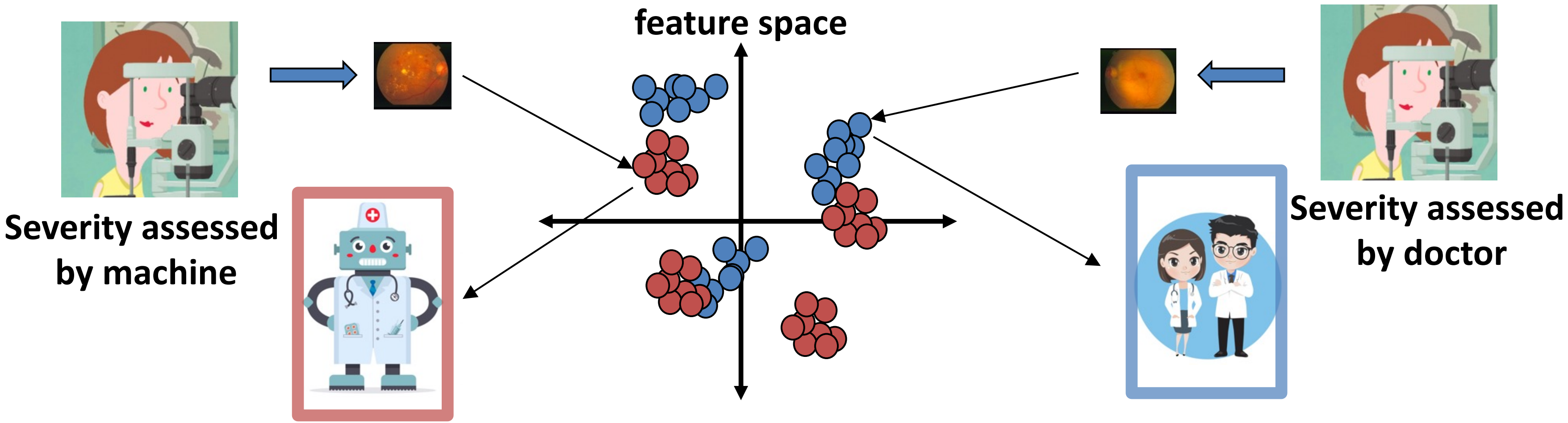


Task: estimating severity of diabetic retinopathy

[Raghu et al., Arxiv 2019]

Learning under algorithmic triage

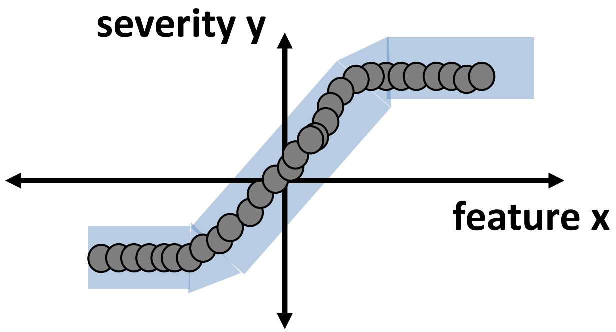
Develop machine learning models that are **optimized** to operate under **different automation levels**



They take **decisions** for a given **fraction of the instances** and **leave the remaining ones to humans**

Isn’t learning under triage just learning to defer?

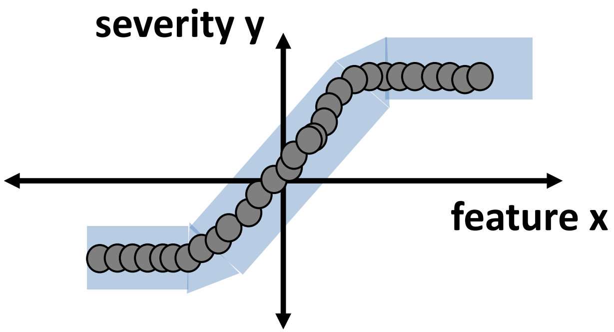
Regression task



1. The machine model is a linear function
2. We can defer some samples to humans, as dictated by a *triage policy*

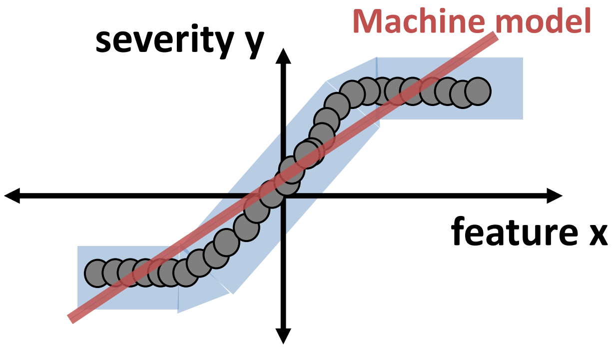
Isn't learning under triage just learning to defer?

Regression task



1. The machine model is a linear function
2. We can defer some samples to humans, as dictated by a *triage policy*

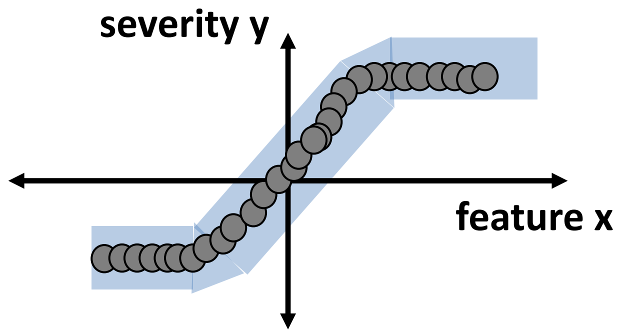
Learning to defer



Machine model is not optimized during training

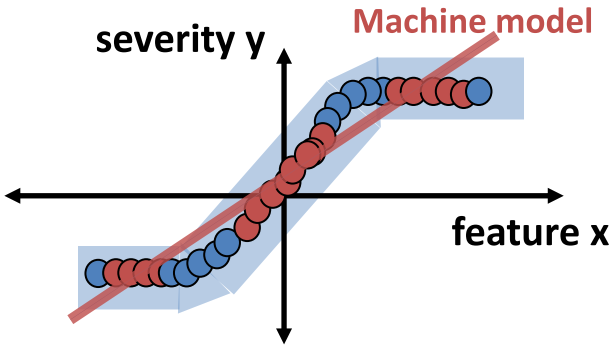
Isn't learning under triage just learning to defer?

Regression task



1. The machine model is a linear function
2. We can defer some samples to humans, as dictated by a *triage policy*

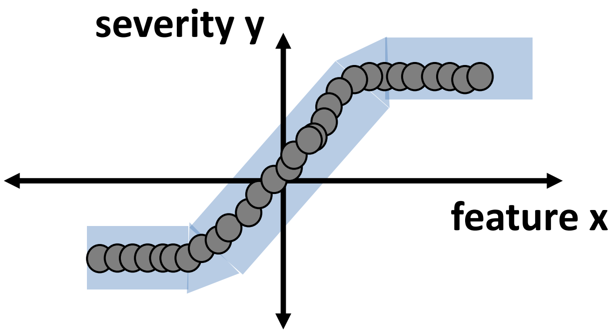
Learning to defer



Machine model is not optimized during training

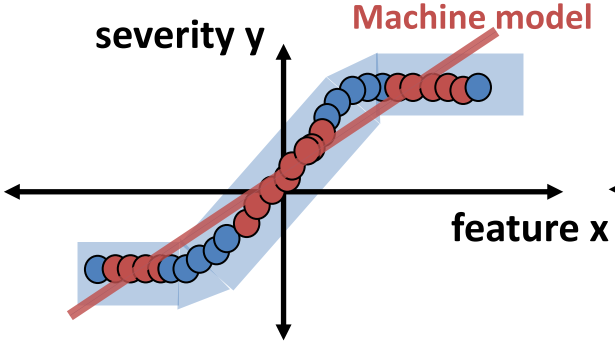
Isn’t learning under triage just learning to defer?

Regression task



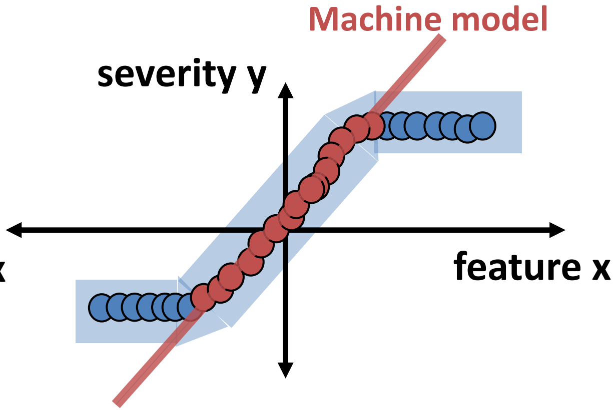
1. The machine model is a linear function
2. We can defer some samples to humans, as dictated by a *triage policy*

Learning to defer



Machine model is not optimized during training

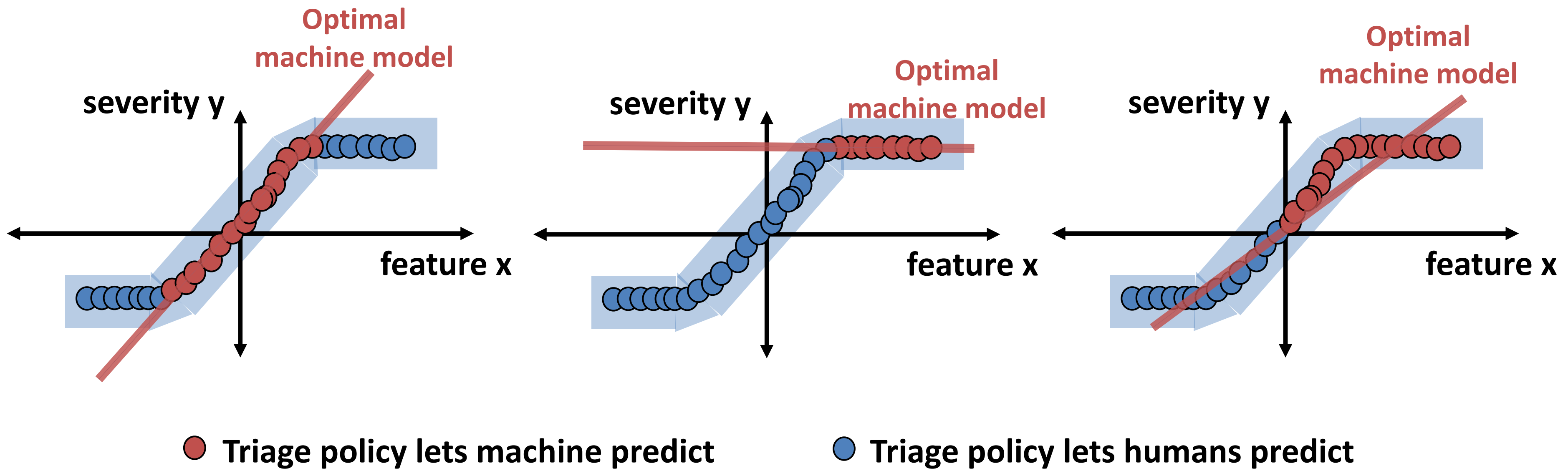
Learning under triage



Machine model is optimized during training

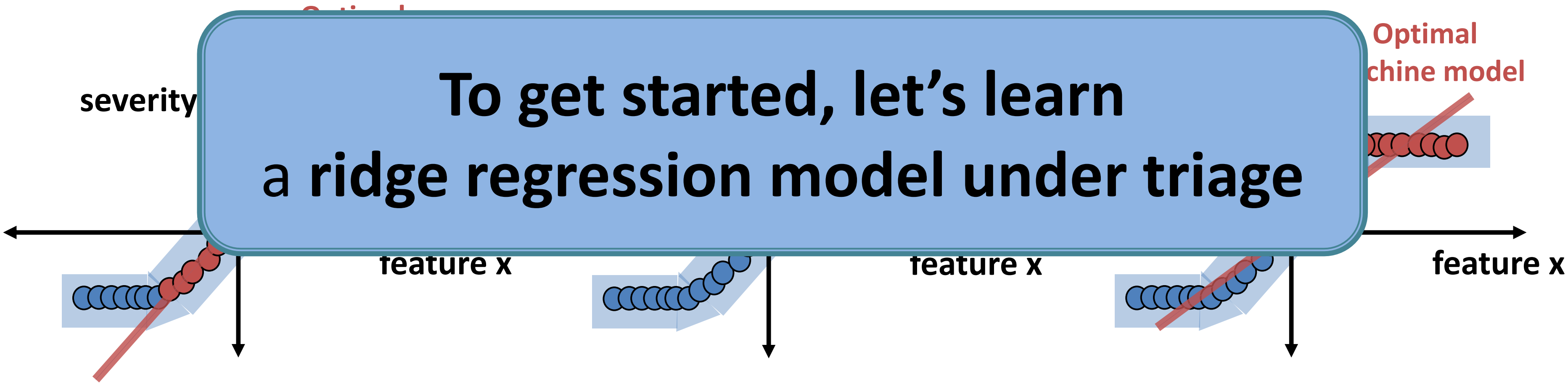
Main challenge of learning under algorithmic triage

For each **triage policy**, there is an **optimal machine model.**
However, the **triage policy** is also something **one seeks to optimize**.



Main challenge of learning under algorithmic triage

For each **triage policy**, there is an **optimal machine model.**
However, the **triage policy** is also something **one seeks to optimize**.



To get started, let's learn a ridge regression model under triage

● Triage policy lets machine predict ● Triage policy lets humans predict

Ridge regression, revisited

Training

Regularization parameter

$$\underset{\boldsymbol{w},\,\mathcal{S}}{\text{minimize}} \quad \sum_{i\in\mathcal{S}} \overbrace{c(\boldsymbol{x}_i, y_i)}^{\text{Human error per sample}} + \sum_{j\in\mathcal{S}^c} \left[(y_j - \boldsymbol{x}_j^\top \boldsymbol{w})^2 + \lambda||\boldsymbol{w}||_2^2\right]$$

$$\text{subject to} \quad |\mathcal{S}| \le n$$

Training samples assigned to humans

Training samples assigned to machines

Machine model

Max. number of samples that can be assigned to humans

Ridge regression, revisited

Training

$$\underset{\boldsymbol{w},\,\mathcal{S}}{\text{minimize}} \quad \sum_{i\in\mathcal{S}} \overbrace{c(\boldsymbol{x}_i, y_i)}^{\text{Human error per sample}} + \sum_{j\in\mathcal{S}^c} \left[(y_j - \boldsymbol{x}_j^\top \boldsymbol{w})^2 + \lambda||\boldsymbol{w}||_2^2\right]$$

$$\text{subject to} \quad |\mathcal{S}| \leq n$$

Regularization parameter

Machine model

Training samples assigned to humans

Training samples assigned to machines

Max. number of samples that can be assigned to humans

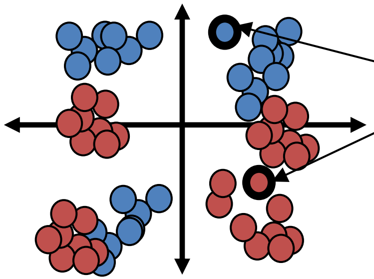
Test

We **assign** a **test sample** with **features** $\boldsymbol{x}$ to **humans** if

$$\text{argmin}_{i\in\mathcal{V}} ||\boldsymbol{x}_i - \boldsymbol{x}|| \in \mathcal{S}^*$$

Training samples assigned to humans

Training samples $\mathcal{V}$



Ridge regression becomes a combinatorial problem

Given a fixed **set** $\mathcal{S}$**,** the **optimal machine model** is given by

$$\boldsymbol{w}^*(S) = \left(\lambda|\mathcal{S}^c|\mathbb{I} + \boldsymbol{X}_{\mathcal{S}^c}\boldsymbol{X}_{\mathcal{S}^c}^\top\right)^{-1}\boldsymbol{X}_{\mathcal{S}^c}\boldsymbol{y}_{\mathcal{S}^c}$$

Ridge regression becomes a combinatorial problem

Given a fixed **set** $\mathcal{S}$**,** the **optimal machine model** is given by

$$\boldsymbol{w}^*(S) = \left(\lambda|\mathcal{S}^c|\mathbb{I} + \boldsymbol{X}_{\mathcal{S}^c}\boldsymbol{X}_{\mathcal{S}^c}^\top\right)^{-1}\boldsymbol{X}_{\mathcal{S}^c}\boldsymbol{y}_{\mathcal{S}^c}$$

Then, we can **rewrite** the **ridge regression problem** as a purely **combinatorial maximization problem**

$$\underset{\mathcal{S}}{\text{maximize}} \quad - \underbrace{\log \ell(\mathcal{S})}_{} \qquad \text{subject to} \quad |\mathcal{S}| \leq n$$

$$\textstyle\sum_{i\in S} c(\boldsymbol{x}_i, y_i) + \boldsymbol{y}_{\mathcal{S}^c}^\top \boldsymbol{y}_{\mathcal{S}^c} - \boldsymbol{y}_{\mathcal{S}^c}^\top \boldsymbol{X}_{\mathcal{S}^c}^\top \left(\lambda|\mathcal{S}^c|\mathbb{I} + \boldsymbol{X}_{\mathcal{S}^c}\boldsymbol{X}_{\mathcal{S}^c}^\top\right)^{-1}\boldsymbol{X}_{\mathcal{S}^c}\boldsymbol{y}_{\mathcal{S}^c}$$

Ridge regression under human assistance is hard

Finding the solution to

$$\underset{\mathcal{S}}{\text{maximize}} \quad -\log \ell(\mathcal{S}) \qquad \text{subject to} \quad |\mathcal{S}| \leq n$$

is a **NP-hard problem**

Ridge regression under human assistance is hard

Finding the solution to

$$\underset{\mathcal{S}}{\text{maximize}} \quad -\log \ell(\mathcal{S}) \quad \text{subject to} \quad |\mathcal{S}| \leq n$$

is a **NP-hard problem**

Proof sketch

k-sparse noise vector

Assume $c(\boldsymbol{x}_i, y_i) = 0$, $\lambda = 0$ and $\boldsymbol{y} = \boldsymbol{X}^\top \boldsymbol{w}^* + \boldsymbol{b}^*$

Then, **the problem** can be viewed as the **robust least square (RLSR) problem,** which **has been shown to be NP-hard:**

$$\underset{\boldsymbol{w}, \mathcal{S}}{\text{minimize}} \sum_{i \in \mathcal{S}} (y_i - \boldsymbol{x}_i^\top \boldsymbol{w})^2 \text{ subject to } |\mathcal{S}| = |\mathcal{V}| - n$$

A simple greedy algorithm

The **greedy algorithm** proceeds **iteratively.**

At each iteration, it **assigns to a human** the **sample in the training set** that provides the **largest marginal gain**, i.e.,

$$k^* \leftarrow \text{argmax}_{k \in \underbrace{\mathcal{V} \backslash \mathcal{S}}_{\text{Points not yet assigned to humans}}} \overbrace{- \log \ell(\mathcal{S} \cup k) + \log \ell(\mathcal{S})}$$

$$\mathcal{S} \leftarrow \mathcal{S} \cup \{k^*\}$$

Does this simple greedy algorithm has approximation guarantees? ☺

The greedy algorithm has approximation guarantees

The function $-\log \ell(\mathcal{S})$ satisfies an **approximate notion of submodularity**

$$-\log \ell(\mathcal{S} \cup k) + \log \ell(\mathcal{S}) \geq (1-\alpha)\left[-\log \ell(\mathcal{T} \cup k) + \log \ell(\mathcal{T})\right]$$

for all $\mathcal{S} \subseteq \mathcal{T} \subset \mathcal{V}$

where $\alpha \leq \alpha^*$ is the **generalized curvature**

Data dependent constant

We can conclude that the greedy algorithm will find a set $\mathcal{S}$ such that

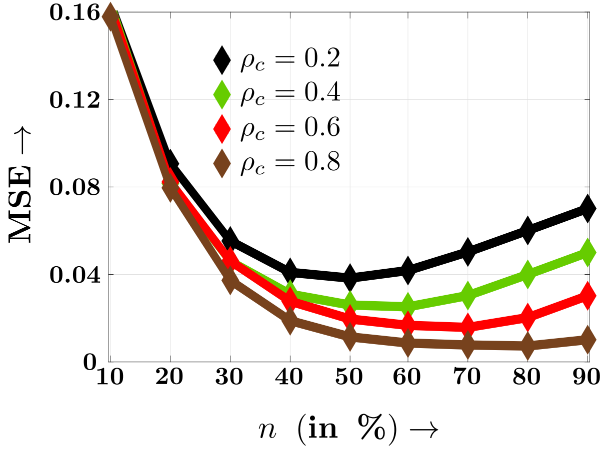
Optimal value

$$\boxed{-\log \ell(\mathcal{S}) \geq \left(1 + \frac{1}{1-\alpha}\right)^{-1} OPT}$$

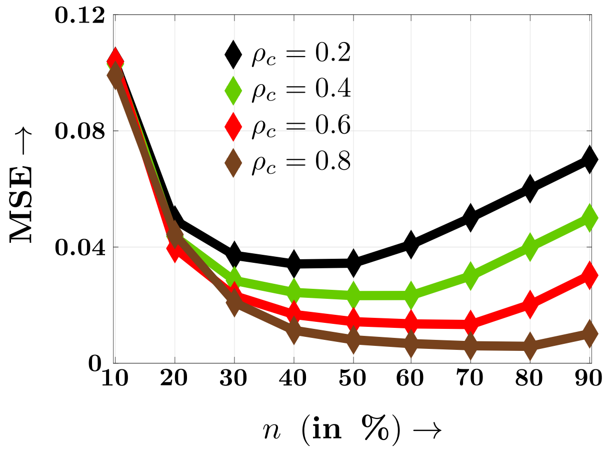
[Gatmiry & Gomez Rodriguez, ACM TOIS 2021]

The greedy algorithm spots samples where humans are accurate

Stare-H dataset



Stare-D dataset

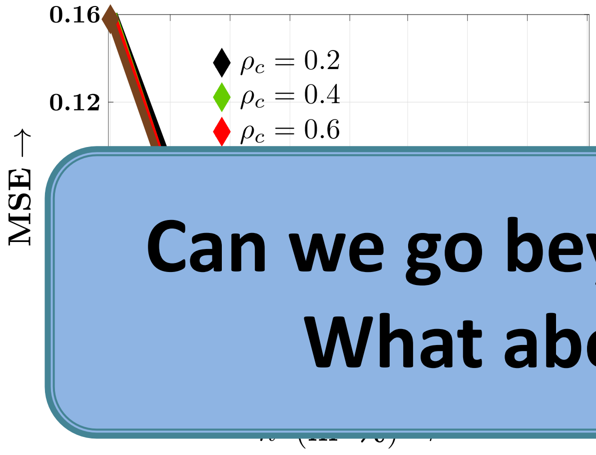


ρ_c: fraction of samples with low human error

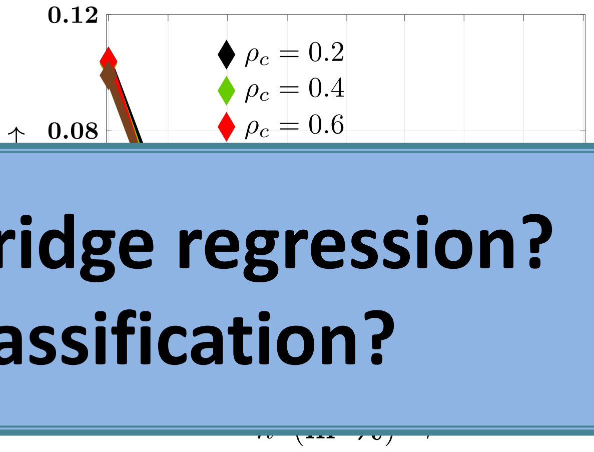
As long as there are **samples that humans can predict with low error**, the **greedy algorithm outsources them to humans** and the **performance improves**

The greedy algorithm spots samples where humans are accurate

Stare-H dataset



Stare-D dataset



Can we go beyond ridge regression?
What about classification?

ρ_c: fraction of samples with low human error

As long as there are **samples that humans can predict with low error**, the **greedy algorithm outsources them to humans** and the **performance improves**

Convex margin-based classifiers, revisited

Regularization parameter

Machine model

Human predictions

$$\underset{\mathcal{S},\boldsymbol{w},b}{\text{minimize}} \sum_{i\in\mathcal{V}\setminus\mathcal{S}} \underbrace{[\lambda\|\boldsymbol{w}\|^2 + [1 - y_i(\boldsymbol{w}^\top\Phi(\boldsymbol{x}_i)+b)]_+]}_{\ell(h_{\boldsymbol{w},b}(\boldsymbol{x}_i),y_i)} + \sum_{i\in\mathcal{S}} \underbrace{[1-y_i h(\boldsymbol{x}_i)]_+}_{c(\boldsymbol{x}_i,y_i)}$$

$$\text{subject to} \quad |\mathcal{S}| \leq n$$

Training samples assigned to machines

Training samples assigned to humans

Max. number of samples that can be assigned to humans

Key idea: measure the error in the human predictions (or, scores) using the same (hinge) loss used in the machine model

Convex optimization + combinatorial optimization

Given a fixed **set** $\mathcal{S}$, we can find the **optimal machine model** using convex programming:

$$\boldsymbol{w}^*(\mathcal{V}\backslash\mathcal{S}), b^*(\mathcal{V}\backslash\mathcal{S}) = \operatorname{argmin}_{\boldsymbol{w},b} \sum_{i\in\mathcal{V}\backslash\mathcal{S}} [\lambda\|\boldsymbol{w}\|^2 + (1 - y_i(\boldsymbol{w}^\top\Phi(\boldsymbol{x}_i) + b))_+]$$

Then, we can **rewrite** the **optimization problem** as the maximization of the difference of two set functions:

$$\underset{\mathcal{S}}{\text{maximize}} \quad g(\mathcal{S}) - c(\mathcal{S}) \qquad \text{subject to} \quad |\mathcal{S}| \leq n$$

Convex optimization + combinatorial optimization

We can show that the **two set functions** satisfy several **desirable properties**:

$$\underset{\mathcal{S}}{\text{maximize}} \quad \underbrace{g(\mathcal{S})}_{\substack{\textbf{Non-negative}\\ \textbf{monotone}\\ \boldsymbol{\gamma}\textbf{-weakly}\\ \textbf{submodular}}} - \underbrace{c(\mathcal{S})}_{\substack{\textbf{Non-negative}\\ \textbf{modular}}} \quad \text{subject to} \quad |\mathcal{S}| \leq n$$

As a result, a **distorted greedy algorithm** is guaranteed to find a set $\mathcal{S}$ such that:

$$g(\mathcal{S}) - c(\mathcal{S}) \geq (1 - e^{-\gamma})g(\texttt{OPT}) - c(\texttt{OPT})$$

We can upper-bound this constant

Differentiable learning under triage

So far, we "solved" the problem of **learning under triage for ridge regression and SVMs**.

↳ **The solutions are specialized and it is not easy to extend the resulting methodology to, e.g., deep learning.**

Differentiable learning under triage

So far, we "solved" the problem of **learning under triage for ridge regression and SVMs**.

↳ **The solutions are specialized and it is not easy to extend the resulting methodology to, e.g., deep learning.**

Next, we design a **principled & scalable method** that **builds** upon **optimization methods (e.g., SGD)** used **in deep learning**

→ **It does not increase the complexity of vanilla SGD**

→ **It is very easy to implement**

→ **It is guaranteed to converge to a local minimum**

Learning under triage, revisited

We look for the **triage policy** $\pi(\mathbf{x})$ and **predictive model** $m(\mathbf{x})$ that minimize a loss function $\ell(\hat{y}, y)$

$P(\mathbf{x})P(y\,|\,\mathbf{x})\,P(h\,|\,\mathbf{x})$

Model predictions

Human predictions

$$\underset{\pi\in\Pi, m\in\mathcal{M}}{\text{minimize}} \quad \mathbb{E}_{\mathbf{x},y,h}\left[(1-\pi(\mathbf{x}))\,\ell(m(\mathbf{x}),y) + \pi(\mathbf{x})\,\ell(h,y)\right]$$

$$\text{subject to} \quad \underbrace{\mathbb{E}_{\mathbf{x}}[\pi(\mathbf{x})] \leq b}_{\text{Maximum level of triage}}$$

Set of all triage policies

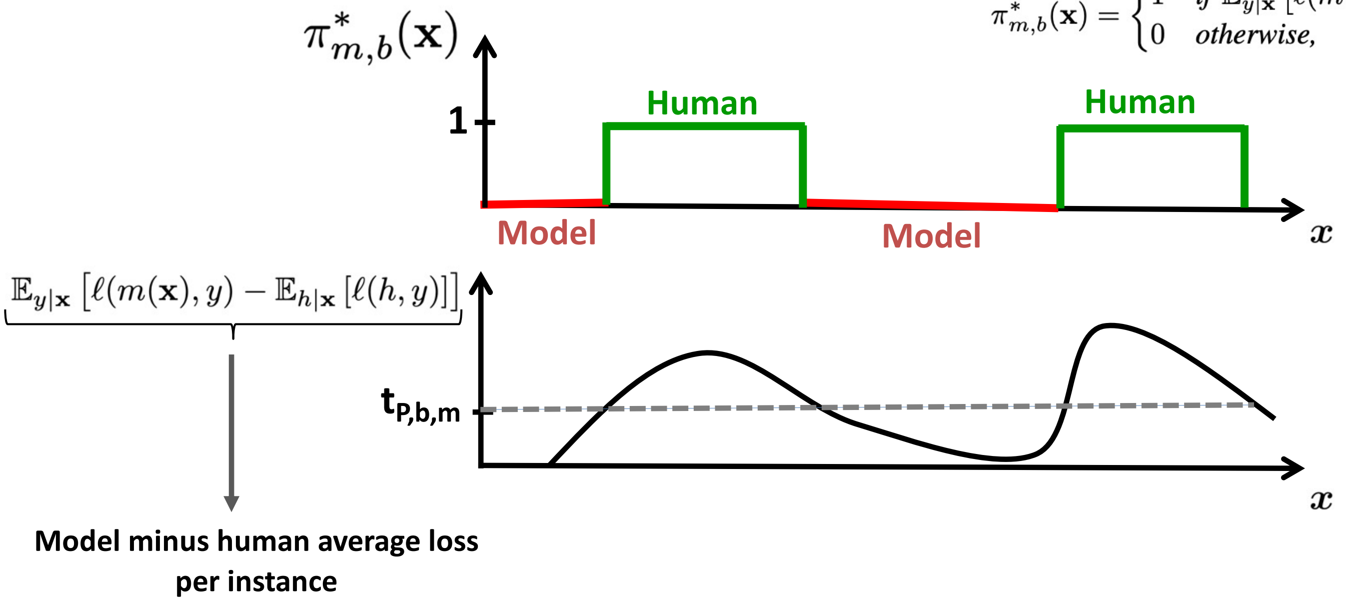
Hypothesis class of predictive models

Triage policy
- $\pi(\mathbf{x}) = 0$ **Model predicts sample x**
- $\pi(\mathbf{x}) = 1$ **Human predicts sample x**

The optimal triage policy is a threshold rule

Let $m \in \mathcal{M}$ be any fixed predictive model. Then, the optimal triage policy $\pi^*_{m,b}(\mathbf{x})$ is a **deterministic threshold rule**:

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



When is a predictive model suboptimal under triage?

Let $m_{\theta_0^*}$ the **optimal predictive model under full automation** ⇨ $\underbrace{\mathbb{E}_{\mathbf{x},y}\left[\nabla_\theta\ \ell(m_\theta(\mathbf{x}), y)\big|_{\theta=\theta_0^*}\right]}_{\textbf{Average gradient on all samples}} = \mathbf{0}$

When is a predictive model suboptimal under triage?

Let $m_{\theta_0^*}$ the **optimal predictive model under full automation** ➡ $\overbrace{\mathbb{E}_{\mathbf{x},y}\left[\nabla_\theta\ \ell(m_\theta(\mathbf{x}),y)|_{\theta=\theta_0^*}\right]}^{\textbf{Average gradient on all samples}} = \mathbf{0}$

Let $\pi^*_{m_{\theta_0^*},b}$ **its optimal triage policy** under max. triage level b. Then, if

$$\underbrace{\int_{\mathbf{x}\in\mathcal{V}} \mathbb{E}_{y|\mathbf{x}}\left[\nabla_\theta \ell(m_\theta(\mathbf{x}),y)|_{\theta=\theta_0^*}\right] dP}_{\text{Average gradient on the set of samples } \mathcal{V} \text{ that } \pi^*_{m_{\theta_0^*},b} \text{ lets the model predicts}} \neq \mathbf{0}$$

When is a predictive model suboptimal under triage?

Let $m_{\theta_0^*}$ the **optimal predictive model under full automation** $\Rightarrow$ $\overbrace{\mathbb{E}_{\mathbf{x},y}\left[\nabla_\theta \, \ell(m_\theta(\mathbf{x}), y)|_{\theta=\theta_0^*}\right]}^{\textbf{Average gradient on all samples}} = \mathbf{0}$

Let $\pi^*_{m_{\theta_0^*},b}$ **its optimal triage policy** under max. triage level b. Then, if

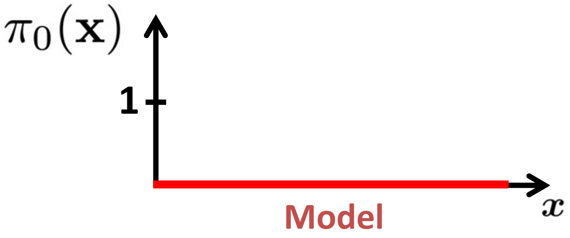
$$\underbrace{\int_{\mathbf{x}\in\mathcal{V}} \mathbb{E}_{y|\mathbf{x}}\left[\nabla_\theta \ell(m_\theta(\mathbf{x}), y)|_{\theta=\theta_0^*}\right] dP}_{\textbf{Average gradient on the set of samples } \mathcal{V} \textbf{ that } \pi^*_{m_{\theta_0^*},b} \textbf{ lets the model predicts}} \neq \mathbf{0}$$

Then, **there exists a better predictive model** under $\pi^*_{m_{\theta_0^*},b}$:

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

Augmenting SGD to learn under triage

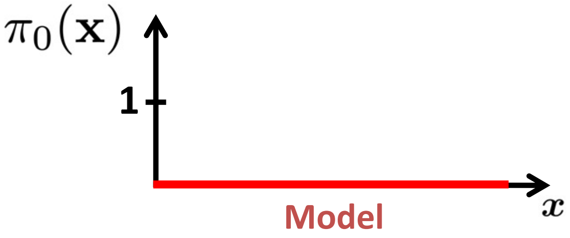
Step t = 0



Train predictive model $m_{\theta_0^*}$ **using SGD with samples for which** $\pi_0(\mathbf{x}) = 0$

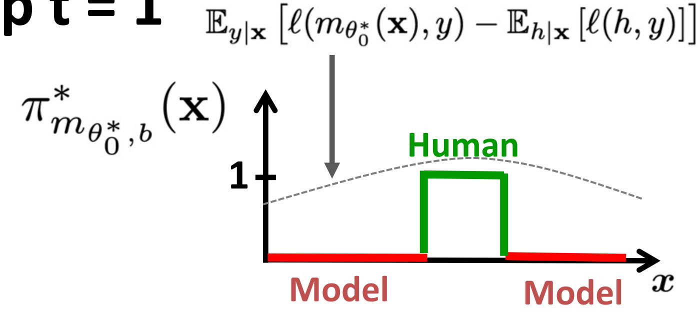
Augmenting SGD to learn under triage

Step t = 0



Train predictive model $m_{\theta_0^*}$ using SGD with samples for which $\pi_0(\mathbf{x}) = 0$

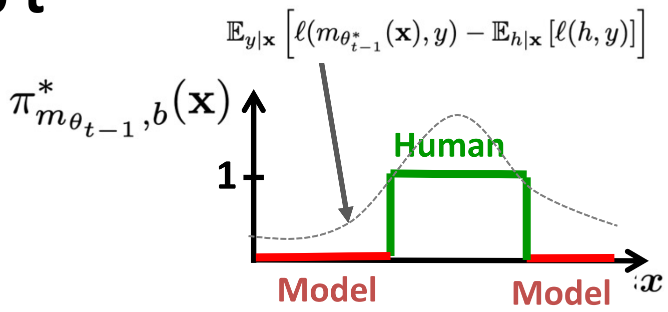
Step t = 1



Train predictive model $m_{\theta_1^*}$ using SGD with samples for which $\pi^*_{m_{\theta_0^*},b}(\mathbf{x}) = 0$

Augmenting SGD to learn under triage

Step t



Train predictive model m_{θ_t}
using SGD with samples for which
$\pi^*_{m_{\theta_{t-1}},b}(\mathbf{x}) = 0$

Local convergence guarantees of SGD under triage

Under **mild conditions,** it holds that:

$$\underbrace{L(\pi^*_{m_{\theta_{t-1}},b}, m_{\theta_{t-1}}) < L(\pi^*_{m_{\theta_t},b}, m_{\theta_t})}$$

The performance of the triage policies and predictive models improves in each step

An **implementation of the algorithm** with **Monte-Carlo estimates converges to a local minimum of the empirical loss**

Global guarantees for convex losses

Under **convex losses,** it holds that:

$$\lim_{t \to \infty} L(\pi^*_{m_{\theta_t},b}, m_{\theta_t}) - L(\pi^*_{m_{\theta^*},b}, m_{\theta^*}) \leq \frac{4H^2\Lambda_{\max}}{\Lambda^2_{\min}(1-b)^2}$$

Maximum level of triage

where

$$\ell(m_\theta(\mathbf{x}), y) - \ell(m_{\theta'}(\mathbf{x}), y) \leq H \cdot \|\theta - \theta'\|$$

$$\Lambda_{\min}\mathbb{I} \preccurlyeq \nabla^2_\theta \, \ell(m_\theta(\mathbf{x}), y) \preccurlyeq \Lambda_{\max}\mathbb{I}$$

SGD under triage at test time

Let m_{θ_T} and $\pi^*_{m_{\theta_T},b}(\mathbf{x})$ the last predictive model and its optimal triage policy.

At test time, we have a problem:
we cannot compute the triage value $\pi^*_{m_{\theta_T},b}(\mathbf{x})$ for unseen samples. We do not know the value of the model/human loss!

SGD under triage at test time

Let m_{θ_T} and $\pi^*_{m_{\theta_T},b}(\mathbf{x})$ the last predictive model and its optimal triage policy.

At test time, we have a problem:
we cannot compute the triage value $\pi^*_{m_{\theta_T},b}(\mathbf{x})$ for unseen samples. We do not know the value of the model/human loss!

Solution:
we train a parameterized triage policy $\hat{\pi}_\gamma(\mathbf{x})$ to approximate the optimal triage policy $\pi^*_{m_{\theta_T},b}(\mathbf{x})$

Learning under triage on real datasets

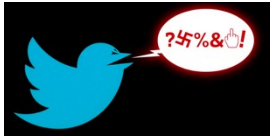
Few public datasets with **several human predictions per instance,** necessary to estimate the **human loss per instance**.

Learning under triage on real datasets

Few public datasets with **several human predictions per instance**, necessary to estimate the **human loss per instance**.

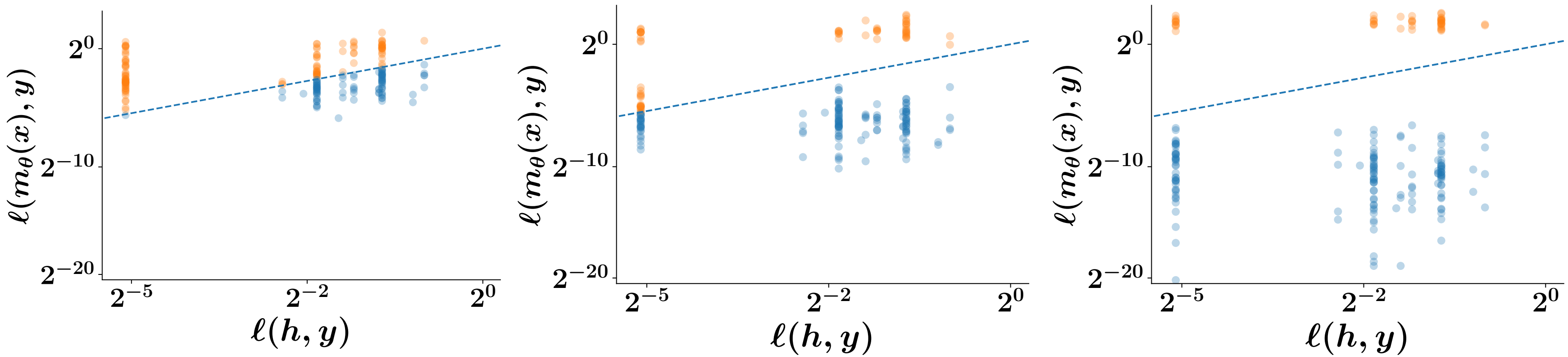
Experiments with **two datasets** from **applications in content moderation** and **scientific discovery:**

→ **Hatespeech:** 25k tweets containing lexicons used in hate speech
3-5 human predictions per instance
labels = {hate-speech, offensive, neither}



→ **Galaxy zoo:** 10k galaxy images
30+ human predictions per instance
labels = {early-type, spiral}

Model vs human loss on the Hatespeech dataset

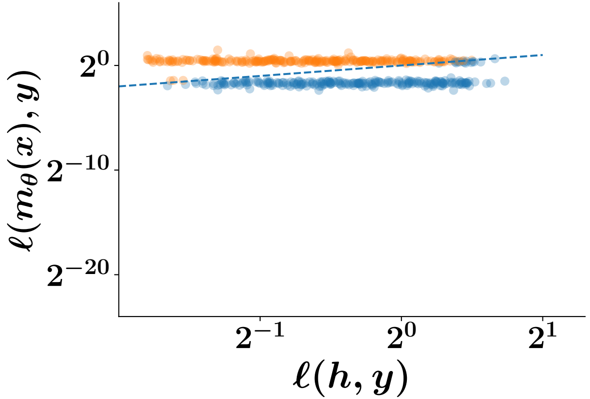


t = 1 **t = 2** **t = 3**

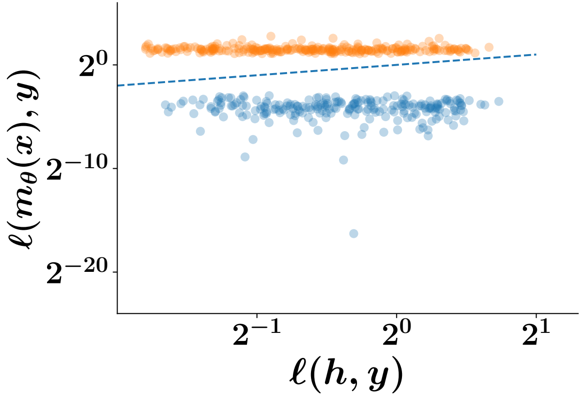
triage level ≤ 40%

The **model learns to specialize** in a **region** of the **feature space**, **where it achieves low loss, and gives up on the rest, where the loss is very high**

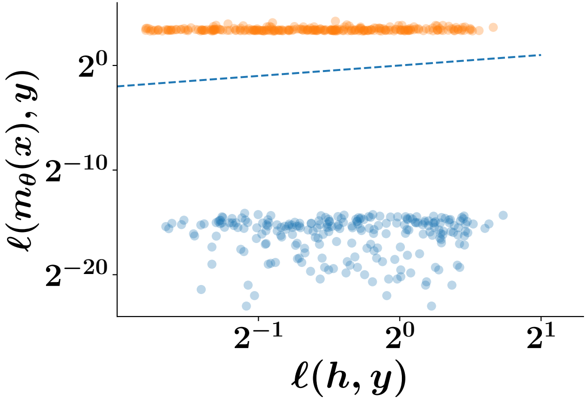
Model vs human loss on the Galaxy zoo dataset



t = 1



t = 2



t = 3

triage level ≤ 100%

The **model learns to specialize** in a **region** of the **feature space**, **where it achieves** **low loss**, **and gives up on** **the rest**, **where the** **loss is very high**

Where do we go from here?

- We have focused on **independent samples. Algorithmic triage in sequential decision making (e.g. semi-autonomous driving).**
- We have assumed **each instance** is **either predicted by a model or by a human**. **All instances predicted by humans, who are informed by a model.**
- **Validate algorithmic triage** using **interventional experiments.**

Thanks!

Regression under human assistance, AAAI 2020

https://arxiv.org/abs/1909.02963

https://github.com/Networks-Learning/regression-under-assistance

Classification under human assistance, AAAI 2021

https://arxiv.org/abs/2006.11845

https://github.com/Networks-Learning/classification-under-assistance

Differentiable learning under triage, NeurIPS 2021

https://arxiv.org/abs/2103.08902

https://github.com/Networks-Learning/differentiable-learning-under-triage