

Regression Under Human Assistance

Abir De

MPI-SWS

ade@mpi-sws.org

Niloy Ganguly

IIT Kharagpur

niloy@cse.iitkgp.ac.in

Paramita Koley

IIT Kharagpur

paramita.koley@iitkgp.ac.in

Manuel Gomez-Rodriguez

MPI-SWS

manuelgr@mpi-sws.org

Abstract

Decisions are increasingly taken by both humans and machine learning models. However, machine learning models are currently trained for full automation—they are not aware that some of the decisions may still be taken by humans. In this paper, we take a first step towards the development of machine learning models that are optimized to operate under different automation levels. More specifically, we first introduce the problem of ridge regression under human assistance and show that it is NP-hard. Then, we derive an alternative representation of the corresponding objective function as a difference of nondecreasing submodular functions. Building on this representation, we further show that the objective is nondecreasing and satisfies α -submodularity, a recently introduced notion of approximate submodularity. These properties allow a simple and efficient greedy algorithm to enjoy approximation guarantees at solving the problem. Experiments on synthetic and real-world data from two important applications—medical diagnosis and content moderation—demonstrate that the greedy algorithm beats several competitive baselines.

1 Introduction

In a wide range of critical applications, societies rely on the judgement of human experts to take consequential decisions—decisions which have significant consequences. Unfortunately, the timeliness and quality of the decisions are often compromised due to the large number of decisions to be taken and a shortage of human experts. For example, in certain medical specialties, patients in most countries need to wait for months to be diagnosed by a specialist. In content moderation, online publishers often stop hosting comments sections because their staff is unable to moderate the myriad of comments they receive. In software development, bugs may be sometimes overlooked by software developers who spend long hours on code reviews for large software projects.

In this context, there is a widespread discussion on the possibility of letting machine learning models take decisions in these high-stake tasks, where they have matched,

or even surpassed, the average performance of human experts (Topol 2019; Cheng, Danescu-Niculescu-Mizil, and Leskovec 2015; Pradel and Sen 2018). Currently, these models are mostly trained for full automation—they assume they will take *all* the decisions. However, their decisions are still worse than those by human experts on some instances, where they make far more errors than average (Raghu et al. 2019a). Motivated by this observation, our goal is to develop machine learning models that are optimized to operate under different automation levels. In other words, these models are optimized to take decisions for a given fraction of the instances and leave the remaining ones to humans.

More specifically, we focus on ridge regression and introduce a novel problem formulation that allows for different automation levels. Based on this formulation, we make the following contributions:

- I. We show that the problem is NP-hard. This is due to its combinatorial nature—for each potential meta-decision about which instances the machine will decide upon, there is an optimal set of parameters for the regression model, however, the meta-decision is also something we seek to optimize.
- II. We derive an alternative representation of the objective function as a difference of nondecreasing submodular functions. This representation enables us to use a recent iterative algorithm (Iyer and Bilmes 2012) to solve the problem, however, this algorithm does not enjoy approximation guarantees.
- III. Building on the above representation, we further show that the objective function is nondecreasing and satisfies α -submodularity, a notion of approximate submodularity (Gatmiry and Gomez-Rodriguez 2019). These properties allow a simple and efficient greedy algorithm (refer to Algorithm 1) to enjoy approximation guarantees.

Here, we would like to acknowledge that our contributions are just a first step towards designing machine learning models that are optimized to operate under different automation levels. It would be very interesting to extend our work to more sophisticated machine learning models and other machine learning tasks (*e.g.*, classification).

Finally, we experiment with synthetic and real-world data

from two important applications—medical diagnosis and content moderation. Our results show that the greedy algorithm beats several competitive algorithms, including the iterative algorithm for maximization of a difference of submodular functions mentioned above, and is able to identify and outsource to humans those samples where their expertise is required. To facilitate research in this area, we are releasing an open source implementation of our method¹.

Related work. The work most closely related to ours is by Raghu et al. (2019a), in which a classifier can outsource samples to humans. However, in contrast to our work, their classifier is trained to predict the labels of all samples in the training set, as in full automation, and the proposed algorithm does not enjoy theoretical guarantees. As a result, a natural extension of their algorithm to ridge regression achieves a significantly lower performance than ours, as shown in Figure 4.

There is a rapidly increasing line of work devoted to designing classifiers that are able to defer decisions (Bartlett and Wegkamp 2008; Cortes, DeSalvo, and Mohri 2016; Geifman, Uziel, and El-Yaniv 2018; Geifman and El-Yaniv 2019; Raghu et al. 2019b; Ramaswamy et al. 2018; Thulasidasan et al. 2019; Liu et al. 2019). Here, the classifiers 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, there are two fundamental differences between this work and ours. First, they do not consider there is a human decision maker, with a human error model, who takes a decision whenever the classifiers defer it. Second, the classifiers are trained to predict the labels of all samples in the training set, as in full automation.

Our work is also related to active learning (Cohn, Ghahramani, and Jordan 1995; Hoi et al. 2006; Willett, Nowak, and Castro 2006; Sugiyama 2006; Guo and Schuurmans 2008; Sabato and Munos 2014; Chen and Price 2017; Hashemi et al. 2019), robust linear regression (Bhatia et al. 2017; Suggala et al. 2019; Tsakonas et al. 2014; Wright and Ma 2010) and robust logistic regression (Feng et al. 2014). In active learning, the goal is to determine which subset of training samples one should label so that a supervised machine learning model, trained on these samples, generalizes well across the entire feature space during test. In contrast, our trained model only needs to accurately predict samples which are close to the samples assigned to the machine during training time and rely on humans to predict the remaining samples. In robust linear regression and robust logistic regression, the (implicit) assumption is that a constant fraction of the output variables are corrupted by an unbounded noise. Then, the goal is to find a consistent estimator of the model parameters which ignores the samples whose output variables are noisy. In contrast, in our work, we do not assume any noise model for the output variables but rather a human error per sample and find an estimator of the model parameters that outsources some of the samples to humans.

¹<https://github.com/Networks-Learning/regression-under-assistance>

2 Problem Formulation

In this section, we formally state the problem of ridge regression under human assistance, where some of the predictions can be outsourced to humans.

Given a set of training samples $\{(\mathbf{x}_i, y_i)\}_{i \in \mathcal{V}}$ and a human error per sample $c(\mathbf{x}_i, y_i)$, we can outsource a subset $\mathcal{S} \subseteq \mathcal{V}$ of the training samples to humans, with $|\mathcal{S}| \leq n$. Then, ridge regression under human assistance seeks to minimize the overall training error, including the outsourced samples, *i.e.*,

$$\underset{\mathbf{w}, \mathcal{S}}{\text{minimize}} \quad \ell(\mathbf{w}, \mathcal{S}) \quad \text{subject to} \quad |\mathcal{S}| \leq n, \quad (1)$$

with

$$\ell(\mathbf{w}, \mathcal{S}) = \sum_{i \in \mathcal{S}} c(\mathbf{x}_i, y_i) + \sum_{j \in \mathcal{S}^c} [(y_j - \mathbf{x}_j^\top \mathbf{w})^2 + \lambda \|\mathbf{w}\|_2^2],$$

where the first term accounts for the human error, the second term accounts the machine error, and λ is a given regularization parameter for the machine.

Moreover, if we define $\mathbf{y} = [y_1, y_2, \dots, y_N]^\top$ and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$, we can rewrite the above objective function as

$$\begin{aligned} \ell(\mathbf{w}, \mathcal{S}) = & \sum_{i \in \mathcal{S}} c(\mathbf{x}_i, y_i) \\ & + (\mathbf{y}_{\mathcal{S}^c} - \mathbf{X}_{\mathcal{S}^c}^\top \mathbf{w})^\top (\mathbf{y}_{\mathcal{S}^c} - \mathbf{X}_{\mathcal{S}^c}^\top \mathbf{w}) + \lambda \|\mathbf{w}\|_2^2 \cdot |\mathcal{S}^c|, \end{aligned}$$

where $\mathbf{y}_{\mathcal{S}^c}$ is the subvector of \mathbf{y} indexed by \mathcal{S}^c and $\mathbf{X}_{\mathcal{S}^c}$ is the submatrix formed by columns of \mathbf{X} that are indexed by \mathcal{S}^c . Then, whenever $\mathcal{S} \subset \mathcal{V}$, it readily follows that the optimal parameter $\mathbf{w}^* = \mathbf{w}^*(\mathcal{S})$ is given by

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

If we plug in the above equation into Eq. 1, we can rewrite the ridge regression problem under human assistance as a set function maximization problem, *i.e.*,

$$\underset{\mathcal{S}}{\text{maximize}} \quad -\log \ell(\mathbf{w}^*(\mathcal{S}), \mathcal{S}) \quad \text{subject to} \quad |\mathcal{S}| \leq n, \quad (2)$$

where $\ell(\mathbf{w}^*(\mathcal{S}), \mathcal{S})$ is given by

$$\begin{cases} \sum_{i \in \mathcal{S}} c(\mathbf{x}_i, y_i) + \mathbf{y}_{\mathcal{S}^c}^\top \mathbf{y}_{\mathcal{S}^c} \\ -\mathbf{y}_{\mathcal{S}^c}^\top \mathbf{X}_{\mathcal{S}^c}^\top (\lambda |\mathcal{S}^c| \mathbb{I} + \mathbf{X}_{\mathcal{S}^c} \mathbf{X}_{\mathcal{S}^c}^\top)^{-1} \mathbf{X}_{\mathcal{S}^c} \mathbf{y}_{\mathcal{S}^c} & \text{if } \mathcal{S} \subset \mathcal{V}, \\ \sum_{i \in \mathcal{S}} c(\mathbf{x}_i, y_i) & \text{if } \mathcal{S} = \mathcal{V}. \end{cases}$$

Unfortunately, due to its combinatorial nature, the above problem formulation is difficult to solve, as formalized by the following Theorem:

Theorem 1 *The problem of ridge regression under human assistance defined in Eq. 1 is NP-hard.*

Proof Consider a particular instance of the problem with $c(\mathbf{x}_i, y_i) = 0$ for all $i \in \mathcal{V}$ and $\lambda = 0$. Moreover, assume the response variables \mathbf{y} are generated as follows:

$$\mathbf{y} = \mathbf{X}^\top \mathbf{w}^* + \mathbf{b}^*, \quad (3)$$

where \mathbf{b}^* is a n -sparse vector which takes non-zero values on at most n corrupted samples, and a zero elsewhere. Then,

the problem can be just viewed as a robust least square regression (RLSR) problem (Studer et al. 2011), *i.e.*,

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

which has been shown to be NP-hard (Bhatia et al. 2017). This concludes the proof. \blacksquare

However, in the next section, we will show that, perhaps surprisingly, a simple greedy algorithm enjoys approximation guarantees. In the remainder of the paper, to ease the notation, we will use $\ell(\mathcal{S}) = \ell(\mathbf{w}^*(\mathcal{S}), \mathcal{S})$.

Remarks. Once the model is trained, given a new unlabeled sample \mathbf{x} , we outsource the sample to a human if the nearest neighbor in the set \mathcal{V} belongs to \mathcal{S}^* , *i.e.*, $\text{argmin}_{i \in \mathcal{V}} \|\mathbf{x}_i - \mathbf{x}\| \in \mathcal{S}^*$, where \mathcal{S}^* is the solution to the above maximization problem, and pass it on to the machine, *i.e.*, $\hat{y}_i = \mathbf{x}_i^\top \mathbf{w}^*(\mathcal{S}^*)$, otherwise. Here, note that, as long as the feature distribution does not change during test, this procedure guarantees that the fraction of samples outsourced to humans during training and test time will be similar. The following proposition formalizes this result:

Proposition 2 *Let $\{\mathbf{x}_i\}_{i \in \mathcal{V}}$ be a set of training samples, $\{\mathbf{x}'_j\}_{j \in \mathcal{V}'}$ a set of test samples, and n and n' the number of training and test samples outsourced to humans, respectively. If $\mathbf{x}_i, \mathbf{x}_j \sim \mathbb{P}(\mathbf{x})$ for all $i \in \mathcal{V}, j \in \mathcal{V}'$, then, it holds that $\mathbb{E}[n']/|\mathcal{V}'| = n/|\mathcal{V}|$.*

3 An Algorithm With Approximation Guarantees

In this section, we first show that the objective function in Eq. 2 can be represented as a difference of nondecreasing submodular functions. Then, we build on this representation to show that the objective function is nondecreasing and satisfies α -submodularity (Gatmiry and Gomez-Rodriguez 2019), a recently introduced notion of approximate submodularity. Finally, we present an efficient greedy algorithm that, due to the α -submodularity of the objective function, enjoys approximation guarantees.

Difference of submodular functions. We first start by rewriting the objective function $\log \ell(\mathcal{S})$ using the following Lemma, which states a well-known property of the Schur complement of a block matrix:

Lemma 3 *Let $\mathbf{Z} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$. If \mathbf{D} is invertible, then $\det(\mathbf{Z}) = \det(\mathbf{D}) \cdot \det(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})$.*

More specifically, consider $\mathbf{A} = \sum_{i \in \mathcal{S}} c(\mathbf{x}_i, y_i) + \mathbf{y}_{\mathcal{S}^c}^\top \mathbf{y}_{\mathcal{S}^c}$, $\mathbf{B} = \mathbf{C}^\top = \mathbf{y}_{\mathcal{S}^c}^\top \mathbf{X}_{\mathcal{S}^c}^\top$ and $\mathbf{D} = \lambda |\mathcal{S}^c| \mathbb{I} + \mathbf{X}_{\mathcal{S}^c} \mathbf{X}_{\mathcal{S}^c}^\top$ in the above lemma. Then, for $\mathcal{S} \subset \mathcal{V}$, it readily follows that:

$$\log \ell(\mathcal{S}) = f(\mathcal{S}) - g(\mathcal{S}) \quad (4)$$

where $f(\mathcal{S})$ is given by

$$\log \det \begin{bmatrix} \sum_{i \in \mathcal{S}} c(\mathbf{x}_i, y_i) + \mathbf{y}_{\mathcal{S}^c}^\top \mathbf{y}_{\mathcal{S}^c} & \mathbf{y}_{\mathcal{S}^c}^\top \mathbf{X}_{\mathcal{S}^c}^\top \\ \mathbf{X}_{\mathcal{S}^c} \mathbf{y}_{\mathcal{S}^c} & \lambda |\mathcal{S}^c| \mathbb{I} + \mathbf{X}_{\mathcal{S}^c} \mathbf{X}_{\mathcal{S}^c}^\top \end{bmatrix}$$

and $g(\mathcal{S})$ is given by,

$$g(\mathcal{S}) = \log \det [\lambda |\mathcal{S}^c| \mathbb{I} + \mathbf{X}_{\mathcal{S}^c} \mathbf{X}_{\mathcal{S}^c}^\top].$$

In the above, note that, for $\mathcal{S} = \mathcal{V}$, the functions f and g are not defined. As it will become clearer later, for $\mathcal{S} = \mathcal{V}$, it will be useful to define their values as follows:

$$\begin{aligned} f(\mathcal{S}) &= \min_{k_1, k_2} \left\{ f(\mathcal{V} \setminus k_1) + f(\mathcal{V} \setminus k_2) - f(\mathcal{V} \setminus \{k_1, k_2\}), \right. \\ &\quad \left. g(\mathcal{V} \setminus k_1) + g(\mathcal{V} \setminus k_2) - g(\mathcal{V} \setminus \{k_1, k_2\}) \right. \\ &\quad \left. + \log \sum_{i \in \mathcal{V}} c(\mathbf{x}_i, y_i) \right\}, \\ g(\mathcal{S}) &= f(\mathcal{S}) - \log \sum_{i \in \mathcal{V}} c(\mathbf{x}_i, y_i), \end{aligned}$$

where note that these values also satisfy Eq. 4. Next, we show that, under mild technical conditions, the above functions are nonincreasing and satisfy a natural diminishing property called submodularity².

Theorem 4 *Assume $c(\mathbf{x}_k, y_k) \leq \gamma y_k^2$ and $\lambda \geq \frac{\gamma}{1-\gamma} \max_{i \in \mathcal{V}} \|\mathbf{x}_i\|_2^2$ with $0 \leq \gamma < 1$, then, f and g are non-increasing and submodular.*

Proof We start by showing that f is submodular, *i.e.*, $f(\mathcal{S} \cup k) - f(\mathcal{S}) \geq f(\mathcal{T} \cup k) - f(\mathcal{T})$ for all $\mathcal{S} \subseteq \mathcal{T} \subset \mathcal{V}$ and $k \in \mathcal{V}$. First, define

$$\mathbf{M}(\mathcal{S}) = \begin{bmatrix} \mathbf{y}_{\mathcal{S}^c}^\top \mathbf{y}_{\mathcal{S}^c} + \sum_{i \in \mathcal{S}} c(\mathbf{x}_i, y_i) & \mathbf{y}_{\mathcal{S}^c}^\top \mathbf{X}_{\mathcal{S}^c}^\top \\ \mathbf{X}_{\mathcal{S}^c} \mathbf{y}_{\mathcal{S}^c} & \lambda |\mathcal{S}^c| \mathbb{I} + \mathbf{X}_{\mathcal{S}^c} \mathbf{X}_{\mathcal{S}^c}^\top \end{bmatrix}.$$

and observe that

$$\mathbf{M}(\mathcal{S} \cup k) = \mathbf{M}(\mathcal{S}) - \begin{bmatrix} y_k^2 - c(\mathbf{x}_k, y_k) & y_k \mathbf{x}_k^\top \\ \mathbf{x}_k y_k & \lambda \mathbb{I} + \mathbf{x}_k \mathbf{x}_k^\top \end{bmatrix}$$

Then, it follows from Proposition 10 (refer to Appendix B) that $\mathbf{M}(\mathcal{S}) - \mathbf{M}(\mathcal{S} \cup k) \succcurlyeq 0$. Hence, we have a Cholesky decomposition $\mathbf{M}(\mathcal{S}) - \mathbf{M}(\mathcal{S} \cup k) = \mathbf{Q}_k \mathbf{Q}_k^\top$. Similarly, we have that $\mathbf{M}(\mathcal{T} \cup k) = \mathbf{M}(\mathcal{T}) - \mathbf{Q}_k \mathbf{Q}_k^\top$, and hence

$$\mathbf{M}(\mathcal{T}) = \mathbf{M}(\mathcal{S}) - \sum_{i \in \mathcal{T} \setminus \mathcal{S}} \mathbf{Q}_i \mathbf{Q}_i^\top \quad (5)$$

Now, for $\mathcal{T} \cup k \subset \mathcal{V}$, a few steps of calculation shows that: $f(\mathcal{S} \cup k) - f(\mathcal{S}) - f(\mathcal{T} \cup k) + f(\mathcal{T})$ equals to

$$\log \frac{\det(\mathbb{I} - \mathbf{Q}_k^\top \mathbf{M}^{-1}(\mathcal{S}) \mathbf{Q}_k)}{\det(\mathbb{I} - \mathbf{Q}_k^\top \mathbf{M}^{-1}(\mathcal{T}) \mathbf{Q}_k)}$$

Moreover, Eq. 5 indicates that $\mathbf{M}(\mathcal{S}) \succcurlyeq \mathbf{M}(\mathcal{T}) \succcurlyeq 0$. Therefore $\mathbf{M}^{-1}(\mathcal{T}) \succcurlyeq \mathbf{M}^{-1}(\mathcal{S})$ and hence $\mathbb{I} - \mathbf{Q}_k^\top \mathbf{M}^{-1}(\mathcal{S}) \mathbf{Q}_k \succcurlyeq \mathbb{I} - \mathbf{Q}_k^\top \mathbf{M}^{-1}(\mathcal{T}) \mathbf{Q}_k$. In addition, we also note that $\mathbf{M}(\mathcal{T}) - \mathbf{Q}_k \mathbf{Q}_k^\top \succcurlyeq 0$. This, together with Lemma 3, we have that $\mathbb{I} - \mathbf{Q}_k^\top \mathbf{M}^{-1}(\mathcal{T}) \mathbf{Q}_k \succcurlyeq 0$. Hence, due to Proposition 11 (refer to Appendix B), we have $\det(\mathbb{I} - \mathbf{Q}_k^\top \mathbf{M}^{-1}(\mathcal{S}) \mathbf{Q}_k) \geq \det(\mathbb{I} - \mathbf{Q}_k^\top \mathbf{M}^{-1}(\mathcal{T}) \mathbf{Q}_k)$. Finally, for $\mathcal{T} \cup k = \mathcal{V}$, we have that

$$f(\mathcal{S} \cup k_1) - f(\mathcal{S}) \geq f(\mathcal{V} \setminus k_2) - f(\mathcal{V} \setminus \{k_1, k_2\}) \quad (6)$$

$$\geq f(\mathcal{V}) - f(\mathcal{V} \setminus \{k_1\}), \quad (7)$$

²A set function $f(\cdot)$ is submodular iff it satisfies that $f(\mathcal{S} \cup \{k\}) - f(\mathcal{S}) \geq f(\mathcal{T} \cup \{k\}) - f(\mathcal{T})$ for all $\mathcal{S} \subseteq \mathcal{T} \subset \mathcal{V}$ and $k \in \mathcal{V}$, where \mathcal{V} is the ground set.

where the first inequality follows from the proof of submodularity for $\mathcal{T} \cup k \subset \mathcal{V}$ and the second inequality comes from the definition of $f(\mathcal{S})$ for $\mathcal{S} = \mathcal{V}$. This concludes the proof of submodularity of f .

Next, we show that f is nonincreasing. First, recall that, for $|\mathcal{S}| < |\mathcal{V}| - 1$, we have that

$$f(\mathcal{S} \cup k) - f(\mathcal{S}) = \log \frac{\det(\mathbf{M}(\mathcal{S}) - \mathbf{Q}_k \mathbf{Q}_k^\top)}{\det(\mathbf{M}(\mathcal{S}))} \quad (8)$$

Then, note that $\mathbf{M}(\mathcal{S}) - \mathbf{Q}_k \mathbf{Q}_k^\top \preceq \mathbf{M}(\mathcal{S})$ and $\mathbf{M}(\mathcal{S}) - \mathbf{Q}_k \mathbf{Q}_k^\top \succeq \mathbf{0}$. Hence, using Proposition 11 (refer to Appendix B), it follows that $\det(\mathbf{M}(\mathcal{S}) - \mathbf{Q}_k \mathbf{Q}_k^\top) \leq \det(\mathbf{M}(\mathcal{S}))$, which proves f is nonincreasing for $|\mathcal{S}| < |\mathcal{V}| - 1$. Finally, for $|\mathcal{S}| = |\mathcal{V}| - 1$, it readily follows from Eq. 7 that

$$f(\mathcal{V}) - f(\mathcal{V} \setminus \{k_1\}) \leq f(\mathcal{V} \setminus \{k_2\}) - f(\mathcal{V} \setminus \{k_1, k_2\}) \quad (9)$$

Now $f(\mathcal{V} \setminus \{k_2\}) - f(\mathcal{V} \setminus \{k_1, k_2\}) \leq 0$ since we have proved that $f(\mathcal{S})$ is nonincreasing for $|\mathcal{S}| < |\mathcal{V}| - 1$. This concludes the proof of monotonicity of f .

Proceeding similarly, it can be proven that g is also non-decreasing and submodular. ■

We would like to highlight that, in the above, the technical conditions have a natural interpretation—the first condition is satisfied if the human error is not greater than a fraction $\sqrt{\gamma}$ of the true response variable and the second condition is satisfied if the regularization parameter is not *too small*.

In our experiments, the above result will enable us to use a series of recent heuristic iterative algorithms for maximizing the difference of submodular functions (Iyer and Bilmes 2012) as baselines. However, these algorithms do not enjoy approximation guarantees—they only guarantee to monotonically reduce the objective function at every step.

Monotonicity. We first start by analyzing the monotonicity of $\log \ell(\mathcal{S})$ whenever $\mathcal{S} = \mathcal{V} \setminus k$, for any $k \in \mathcal{V}$ in the following Lemma (proven in Appendix A):

Lemma 5 Assume $c(\mathbf{x}_k, y_k) < \gamma y_k^2$ and $\lambda > \frac{\gamma}{1-\gamma} \max_{i \in \mathcal{V}} \|\mathbf{x}_i\|_2^2$ with $0 \leq \gamma < 1$. Then, it holds that $\log \ell(\mathcal{V}) - \log \ell(\mathcal{V} \setminus k) < 0$ for all $k \in \mathcal{V}$.

Then, building on the above lemma, we have the following Theorem, which shows that $\log \ell(\mathcal{S})$ is a strictly nonincreasing function (proven in Appendix A):

Theorem 6 Assume $c(\mathbf{x}_k, y_k) < \gamma y_k^2$ and $\lambda > \frac{\gamma}{1-\gamma} \max_{i \in \mathcal{V}} \|\mathbf{x}_i\|_2^2$ with $0 \leq \gamma < 1$, then, the function $\log \ell(\mathcal{S})$ is strictly nonincreasing, i.e.,

$$\log \ell(\mathcal{S} \cup k) - \log \ell(\mathcal{S}) < 0$$

for all $\mathcal{S} \in \mathcal{V}$ and $k \in \mathcal{V}$.

Finally, note that the above result does *not* imply that the human error $c(\mathbf{x}_k, y_k)$ is always smaller than the machine error $(y_k - \mathbf{x}_k^\top \mathbf{w}^*(k))^2$, where $\mathbf{w}^*(k)$ is optimal parameter for $\mathcal{S} = \{k\}$, as formalized by the following Proposition (proven in Appendix A):

Proposition 7 Assume $\rho^2 y_k^2 < c(\mathbf{x}_k, y_k) < \gamma y_k^2$ and $\frac{\gamma}{1-\gamma} \max_{i \in \mathcal{V}} \|\mathbf{x}_i\|_2^2 < \lambda < \frac{\rho}{1-\rho} \max_{i \in \mathcal{V}} \|\mathbf{x}_i\|_2^2$ with $\gamma < \rho < \sqrt{\gamma}$ and $0 \leq \gamma < 1$, then, it holds that

$$c(\mathbf{x}_k, y_k) > (y_k - \mathbf{x}_k^\top \mathbf{w}^*(k))^2.$$

Algorithm 1 Greedy algorithm

Input: Ground set \mathcal{V} , set of training samples $\{(\mathbf{x}_i, y_i)\}_{i \in \mathcal{V}}$, parameters n and λ .

Output: Set of items \mathcal{S}

```

1:  $\mathcal{S} \leftarrow \emptyset$ 
2: while  $|\mathcal{S}| < n$  do
3:   % Find best sample
4:    $k^* \leftarrow \operatorname{argmax}_{k \in \mathcal{V} \setminus \mathcal{S}} -\log \ell(\mathcal{S} \cup k) + \log \ell(\mathcal{S})$ 
5:   % Sample is outsourced to humans
6:    $\mathcal{S} \leftarrow \mathcal{S} \cup \{k^*\}$ 
7: end while
8: return  $\mathcal{S}$ 

```

α -submodularity. Given the above results, we are now ready to present and prove our main result, which characterizes the objective function of the optimization problem defined in Eq. 2:

Theorem 8 Assume $c(\mathbf{x}_k, y_k) < \gamma y_k^2$, $\lambda > \frac{\gamma}{1-\gamma} \max_{i \in \mathcal{V}} \|\mathbf{x}_i\|_2^2$ with $0 \leq \gamma < 1$, and $\sum_{i \in \mathcal{V}} c(\mathbf{x}_i, y_i) \geq 1^3$. Then, the function $-\log \ell(\mathcal{S})$ is a nondecreasing α -submodular function⁴ and the parameter α satisfies that

$$\alpha \leq \alpha^* = 1 - \min \left\{ \frac{(1 - \kappa_\ell) \log \ell(\mathcal{V})}{\max_{k_1, k_2} f(\mathcal{V} \setminus \{k_1, k_2\}) - f(\mathcal{V} \setminus \{k_1\})}, \frac{(1 - \kappa_\ell) \log \ell(\mathcal{V})}{\max_k \log \ell(\mathcal{V} \setminus k) - \log \ell(\mathcal{V})} \right\} \quad (10)$$

$$\text{with, } \kappa_\ell = \frac{\log [\ell(\emptyset) - \min_k (\ell(\mathcal{V} \setminus k) - \ell(\mathcal{V}))]}{\log \ell(\emptyset)}$$

Proof Using that $\sum_{i \in \mathcal{V}} c(\mathbf{x}_i, y_i) > 1$ and the function ℓ is nonincreasing, we can conclude that $1 < \ell(\mathcal{V}) < \ell(\mathcal{S})$. Then, it readily follows from the proof of Theorem 6 that

$$1 < \ell(\mathcal{S} \cup k) < \ell(\mathcal{S}) - (\ell(\mathcal{V} \setminus k) - \ell(\mathcal{V})) \quad (11)$$

Hence we have,

$$\begin{aligned} \frac{\log \ell(\mathcal{S} \cup k)}{\log \ell(\mathcal{S})} &\leq \frac{\log (\ell(\mathcal{S}) - (\ell(\mathcal{V} \setminus k) - \ell(\mathcal{V})))}{\log \ell(\mathcal{S})} \\ &\stackrel{(a)}{=} \frac{\log (\ell(\emptyset) - (\ell(\mathcal{V} \setminus k) - \ell(\mathcal{V})))}{\log \ell(\emptyset)} \\ &\leq \kappa_\ell \end{aligned} \quad (12)$$

where equality (a) follows from Theorem 6, which implies

³Note that we can always rescale the data to satisfy this last condition.

⁴A function $f(\cdot)$ is α -submodular (Gatmiry and Gomez-Rodriguez 2019) iff it satisfies that $f(\mathcal{S} \cup \{k\}) - f(\mathcal{S}) \geq (1 - \alpha)[f(\mathcal{T} \cup \{k\}) - f(\mathcal{T})]$ for all $\mathcal{S} \subseteq \mathcal{T} \subset \mathcal{V}$ and $k \in \mathcal{V}$, where \mathcal{V} is the ground set and α is the generalized curvature (Lehmann, Lehmann, and Nisan 2006; Bogunovic, Zhao, and Cevher 2018; Hashemi et al. 2019).

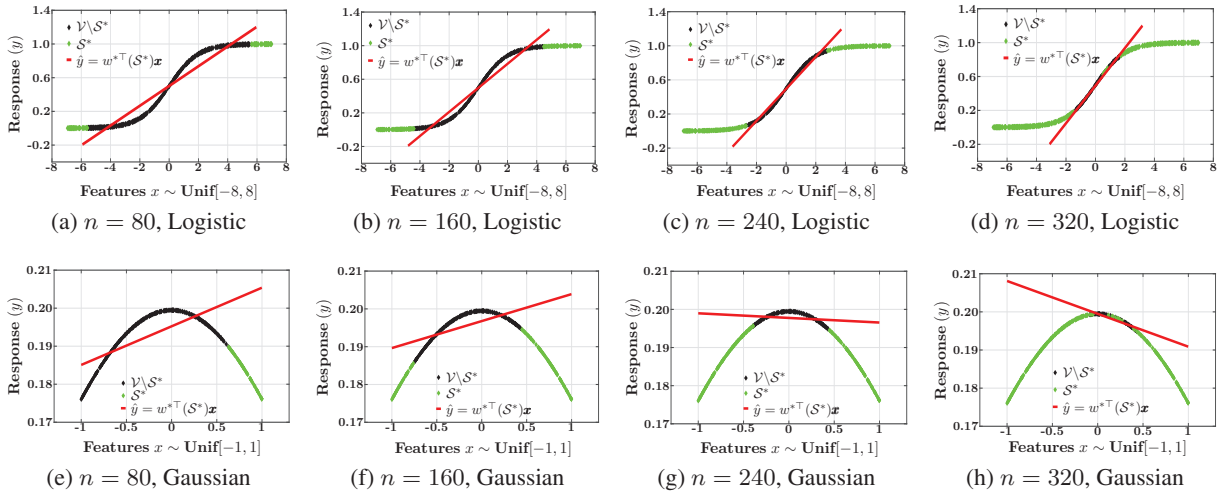


Figure 1: Solution $(w^*(\mathcal{S}^*), \mathcal{S}^*)$ provided by our greedy algorithm for a gaussian and logistic response variable distribution and different number of outsourced samples n . In all cases, we used $d = 1$ and $\sigma_2 = 0.01$. For the logistic distribution, as n increases, the greedy algorithm let the machine to focus on the samples where the relationship between features and the response variables is more linear and outsource the remaining points to humans. For the gaussian distribution, as n increases, the greedy algorithm outsources samples on the tails of the distribution to humans.

that $\ell_{\max} = \ell(\emptyset)$. Then, we have that

$$\begin{aligned}
1 - \alpha &= \min_{k, \mathcal{S} \subseteq \mathcal{T} \subseteq \mathcal{V}} \frac{\log \ell(\mathcal{S}) - \log \ell(\mathcal{S} \cup k)}{\log \ell(\mathcal{T}) - \log \ell(\mathcal{T} \cup k)} \\
&\geq \min \left\{ \min_{k, \mathcal{S} \subseteq \mathcal{T}: |\mathcal{T}| \leq |\mathcal{V}| - 2} \frac{\log \ell(\mathcal{S}) - \log \ell(\mathcal{S} \cup k)}{\log \ell(\mathcal{T}) - \log \ell(\mathcal{T} \cup k)}, \right. \\
&\quad \left. \min_{\mathcal{S}, k} \frac{\log \ell(\mathcal{S}) - \log \ell(\mathcal{S} \cup k)}{\log \ell(\mathcal{V} \setminus k) - \log \ell(\mathcal{V})} \right\} \quad (13)
\end{aligned}$$

Next, we bound the first term as follows:

$$\begin{aligned}
&\min_{k, \mathcal{S} \subseteq \mathcal{T}: |\mathcal{T}| \leq |\mathcal{V}| - 2} \frac{\log \ell(\mathcal{S}) - \log \ell(\mathcal{S} \cup k)}{\log \ell(\mathcal{T}) - \log \ell(\mathcal{T} \cup k)} \\
&\stackrel{(a)}{\geq} \min_{k, \mathcal{S} \subseteq \mathcal{T}: |\mathcal{T}| \leq |\mathcal{V}| - 2} \frac{(1 - \kappa_\ell) \log \ell(\mathcal{S})}{\log \ell(\mathcal{T}) - \log \ell(\mathcal{T} \cup k)} \\
&\stackrel{(b)}{\geq} \min_{k, |\mathcal{T}| \leq |\mathcal{V}| - 2} \frac{(1 - \kappa_\ell) \log \ell(\mathcal{V})}{\log \ell(\mathcal{T}) - \log \ell(\mathcal{T} \cup k)} \\
&= \min_{k, |\mathcal{T}| \leq |\mathcal{V}| - 2} \frac{(1 - \kappa_\ell) \log \ell(\mathcal{V})}{f(\mathcal{T}) - f(\mathcal{T} \cup k) - (g(\mathcal{T}) - g(\mathcal{T} \cup k))} \\
&\stackrel{(c)}{\geq} \min_{k, |\mathcal{T}| \leq |\mathcal{V}| - 2} \frac{(1 - \kappa_\ell) \log \ell(\mathcal{V})}{f(\mathcal{T}) - f(\mathcal{T} \cup k)} \\
&\stackrel{(d)}{\geq} \frac{(1 - \kappa_\ell) \log \ell(\mathcal{V})}{\max_{k_1, k_2} (f(\mathcal{V} \setminus \{k_1, k_2\}) - f(\mathcal{V} \setminus k_1))},
\end{aligned}$$

where inequality (a) follows from Eq. 12, inequality (b) follows from the monotonicity of $\log \ell(\mathcal{S})$, and inequalities (c) and (d) follows from Theorem 4. Finally, we use the monotonicity of $\log \ell(\mathcal{S})$ and Eq. 12 to bound the second term in Eq. 13 is always greater than $\frac{(1 - \kappa_\ell) \log \ell(\mathcal{V})}{\max_k \log \ell(\mathcal{V} \setminus k) - \log \ell(\mathcal{V})}$, which concludes the proof. ■

A greedy algorithm. The greedy algorithm proceeds iteratively and, at each step, it assigns to the humans the sample

(x_k, y_k) that provides the highest marginal gain among the set of samples which are currently assigned to the machine. Algorithm 1 summarizes the greedy algorithm.

Since the objective function in Eq. 2 is α -submodular, it readily follows from Theorem 9 in Khashayar and Gomez-Rodriguez (Gatmiry and Gomez-Rodriguez 2019) that the above greedy algorithm enjoys an approximation guarantee. More specifically, we have the following Theorem:

Theorem 9 *The greedy algorithm returns a set \mathcal{S} such that $-\log \ell(\mathcal{S}) \geq (1 + 1/(1 - \alpha))^{-1} OPT$, where OPT is the optimal value and $\alpha \leq \alpha^*$ with α^* defined in Eq. 10.*

In the above, note that, due to Theorem 6, the actual (regularized) loss function is strictly nonincreasing and thus the greedy algorithm always goes until $|\mathcal{S}| = n$, however, the overall accuracy may be higher for some values of $|\mathcal{S}| < n$ as shown in Figure 6. In the next section, we will demonstrate that, in addition to enjoying the above approximation guarantees, the above greedy algorithm performs better in practice than several competitive baselines.

4 Experiments on Synthetic Data

In this section, we experiment with a variety of synthetic examples. First, we look into the solution $(w^*(\mathcal{S}^*), \mathcal{S}^*)$ provided by the greedy algorithm. Then, we compare the performance of the greedy algorithm with several competitive baselines. Finally, we investigate how the performance of the greedy algorithm varies with respect to the amount of human error.

Experimental setup. For each sample (x, y) , we first generate each dimension of the feature vector $x \in \mathbb{R}^d$ uniformly at random, i.e., $x_i \sim U(-1, 1)$ and then sample the response variable y from either (i) a Gaussian distribution $\mathcal{N}(\mathbf{1}^\top x/d, \sigma_2^2)$ or (ii) a logistic distribution $1/(1 +$

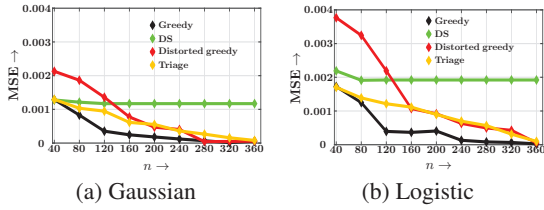


Figure 2: Mean squared error (MSE) against number of outsourced samples n for the proposed greedy algorithm, DS (Iyer and Bilmes 2012), distorted greedy (Harshaw et al. 2019) and Triage (Raghu et al. 2019a) on synthetic data. In all cases, we used $d = 5$ and $\sigma_2 = 10^{-3}$. In panel (a), we set $\lambda = 5 \cdot 10^{-3}$ and, in panel (b), we set $\lambda = 10^{-3}$. The greedy algorithm consistently outperforms the baselines across the entire range of automation levels.

$\exp(-\mathbf{1}^\top \mathbf{x}/d)$). Moreover, we sample the associated human error from a Gaussian distribution, *i.e.*, $c(\mathbf{x}, y) \sim \mathcal{N}(0, \sigma_2^2)$. In each experiment, we use $|\mathcal{V}| = 500$ training samples and we compare the performance of the greedy algorithm with three competitive baselines:

- An iterative heuristic algorithm (DS) for maximizing the difference of submodular functions by Iyer and Bilmes (2012).
- A greedy algorithm (Distorted greedy) for maximizing γ -weakly submodular functions by Harshaw et al. (2019)⁵.
- A natural extension of the algorithm (Triage) by Raghu et al. (2019a), originally developed for classification under human assistance, where we first solve the standard ridge regression problem for the entire training set, then we map each test sample to the nearest neighbor training sample and finally outsource to humans the top n samples sorted in decreasing order of the difference between machine and human error of the assigned training sample.

Results. We first look into the solution $(\mathbf{w}^*(\mathcal{S}^*), \mathcal{S}^*)$ provided by the greedy algorithm both for the Gaussian and logistic distributions and a different number of outsourced samples n . Figure 1 summarizes the results, which reveal several interesting insights. For the logistic distribution, as n increases, the greedy algorithm let the machine to focus on the samples where the relationship between features and the response variables is more linear and outsource the remaining points to humans. For the Gaussian distribution, as n increases, the greedy algorithm outsources samples on the tails of the distribution to humans.

Then, we compare the performance of the greedy algorithm in terms of mean squared error (MSE) on a held-out set against the three competitive baselines. Figure 2 summarizes the results, which show that the greedy algorithm consistently outperforms the baselines across the entire range of automation levels. Finally, we investigate how the performance of our greedy algorithm varies with respect to the amount of human error. Figure 3 summarizes the results, which shows that, for low levels of human error, the overall mean squared error decreases monotonically with respect to

⁵Note that any α -submodular function is γ -weakly submodular (Gatmiry and Gomez-Rodriguez 2019).

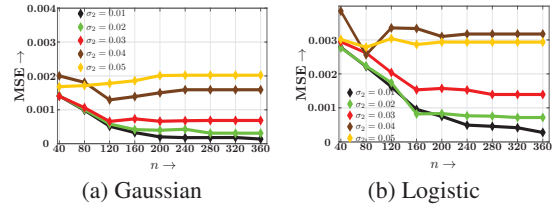


Figure 3: Mean squared error (MSE) achieved by the proposed greedy algorithm against the number of outsourced samples n for different levels of human error (σ_2) on synthetic data. In all cases, we used $d = 5$. In panel (a), we set $\lambda = 5 \cdot 10^{-3}$ and, in panel (b), we set $\lambda = 10^{-3}$. For low levels of human error, the overall mean squared error decreases monotonically with respect to the number of outsourced samples. In contrast, for high levels of human error, it is not beneficial to outsource samples to humans.

the number of outsourced samples to humans. In contrast, for high levels of human error, it is not beneficial to outsource samples.

5 Experiments on Real Data

In this section, we experiment with four real-world datasets from two important applications, medical diagnosis and content moderation, and show that the greedy algorithm beats several competitive baselines. Moreover, we also look at the samples that the greedy algorithm outsources to humans and show that, for different distributions of human error, the outsourced samples are those that humans are able to predict more accurately.

Experimental setup. We experiment with one dataset for content moderation and three datasets for medical diagnosis, which are publicly available (Davidson et al. ; Decencière et al. 2014; Hoover, Kouznetsova, and Goldbaum 2000). More specifically:

- (i) **Hatespeech:** It consists of ~ 1000 tweets⁶ containing words, phrases and lexicons used in hate speech. Each tweet is given several scores by three to five annotators from Crowdfunder, which measure the severity of hate speech.
- (ii) **Stare-H:** It consists of ~ 400 retinal images. Each image is given a score by one single expert, on a five point scale, which measures the severity of a retinal hemorrhage.
- (iii) **Stare-D:** It contains the same set of images from Stare-H. However, in this dataset, each image is given a score by a single expert, on a six point scale, which measures the severity of the Drusen disease.
- (iv) **Messidor:** It contains 500 eye images. Each image is given score by one single expert, on a three point scale, which measures the severity of an edema.

⁶The original Hatespeech dataset consists of ~ 25000 tweets, however, we report results on a randomly chosen subset of 1000 tweets because the distorted greedy and DS algorithms did not scale in the original dataset. We found similar results in other random subsets.

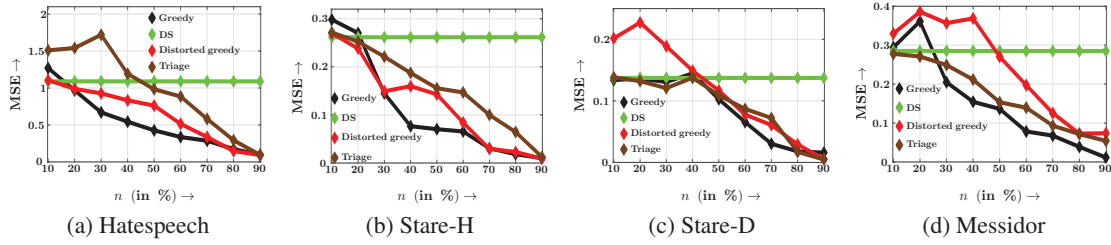


Figure 4: Mean squared error (MSE) against number of outsourced samples n for the proposed greedy algorithm, DS (Iyer and Bilmes 2012), distorted greedy (Harshaw et al. 2019) and triage (Raghu et al. 2019a) on four real-world datasets. In Panels (b-d), we set the parameter $p = 0.90$. The greedy algorithm outperforms the baselines across a majority of automation levels. The only exceptions are high automation levels, where the distorted greedy and the triage algorithms sometimes achieve slightly better performance.

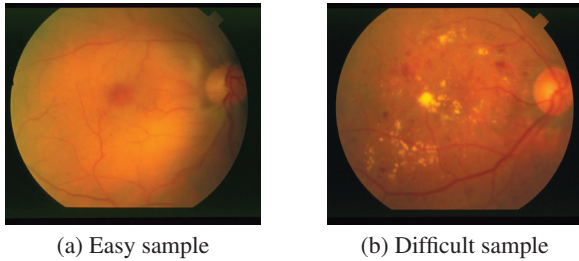


Figure 5: An easy and a difficult sample image from the Stare-D dataset. Both images are given a score of severity zero for the Drusen disease, which is characterized by pathological yellow spots. The easy sample does not contain yellow spots and thus it is easy to predict its score. In contrast, the difficult sample contains yellow spots, which are manifested not from Drusen, but diabetic retinopathy, and thus it is challenging to accurately predict its score. As a result, the greedy algorithm decides to outsource the difficult sample to humans, whereas it lets the machine decide about the easy one.

We first generate a 100 dimensional feature vector using fasttext (Joulin et al. 2016) for each sample in the Hatespeech dataset and a 1000 dimensional feature vector using Resnet (He et al. 2016) for each sample in the Stare-H, Stare-D, and Messidor datasets. Then, we use the top 50 features, as identified by PCA, as \mathbf{x} in our experiments. For the image datasets, the response variable y is just the available score by a single expert and the human predictions are sampled from a categorical distribution $s \sim \text{Cat}(\mathbf{p}_{\mathbf{x}})$, where $\mathbf{p}_{\mathbf{x}}$ are the probabilities of each potential score value s for a sample with features \mathbf{x} . More specifically, if the response variable takes values on a t point scale, we consider:

$$p_{\mathbf{x}}(k) = \begin{cases} p & \text{if } k = y \\ \frac{1-p}{2} & \text{if } k \in \{y-1, y+1\} \text{ and } 1 < y < t \\ 1-p & \text{if } k = y-1 \text{ and } y = t \\ 1-p & \text{if } k = y+1 \text{ and } y = 1, \end{cases}$$

where p is a parameter that controls the human accuracy. For the Hatespeech dataset, the response variable y is the mean of the scores provided by the annotators and the human pre-

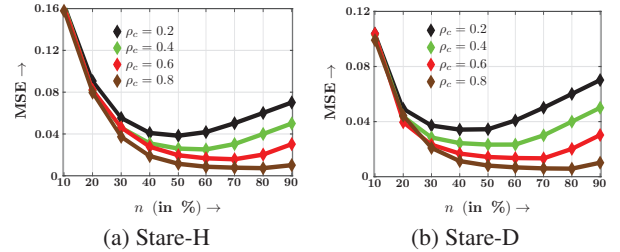


Figure 6: Mean squared error (MSE) achieved by the proposed greedy algorithm against the number of outsourced samples n under different distributions of human errors on two real-world datasets. Under each distribution of human error, human error is low ($c(\mathbf{x}, y) = 10^{-4}$) for a fractions ρ_c of the samples and high ($c(\mathbf{x}, y) = 0.5$) for the remaining fraction $1 - \rho_c$. As long as there are samples that humans can predict with low error, the greedy algorithm does outsource them to humans and thus the overall performance improves. However, whenever the fraction of outsourced samples is higher than the fraction of samples with low human error, the performance degrades. This results in a characteristic U-shaped curve.

dictions are picked uniformly at random from the available individual scores given by each annotator. In each dataset, we compute the human error as $c(\mathbf{x}, y) = \mathbb{E}(y - s)^2$ for each sample (\mathbf{x}, y) and set the same value of λ across all competitive methods. Finally, in each experiment, we use 80% samples for training and 20% samples for testing.

Results. We first compare the performance of the greedy algorithm in terms of mean squared error (MSE) on a held-out set against the same competitive baselines used in the experiments on synthetic data, *i.e.*, DS (Iyer and Bilmes 2012), distorted greedy (Harshaw et al. 2019), and triage (Raghu et al. 2019a). Figure 4 summarizes the results, which show that the greedy algorithm outperforms the baselines across a majority of automation levels. The only exceptions are high automation levels, where the distorted greedy and the triage algorithms sometimes achieve slightly better performance.

Next, we look at the samples that the greedy algorithm outsources to humans and those that leaves to machines. Intuitively, human assistance should be required for those sam-

ples which are difficult (easy) for a machine (a human) to decide about. Figure 5 provides an illustrative example of an easy and a difficult sample image. While both sample images are given a score of severity zero for the Drusen disease, one of them contains yellow spots, which are often a sign of Drusen disease⁷, and is therefore difficult to predict. In this particular case, the greedy algorithm outsourced the difficult sample to humans and let the machine decide about the easy one. Does this intuitive assignment happen consistently? To answer this question, we run our greedy algorithm on the Stare-H and Stare-D datasets under different distributions of human error and assess to which extent the greedy algorithm outsources to humans those samples they can predict more accurately.

More specifically, we sample the human predictions from a non-uniform categorical distribution under which human error is low ($c(\mathbf{x}, y) = 10^{-4}$) for a fraction ρ_c of the samples and high ($c(\mathbf{x}, y) = 0.5$) for the remaining fraction $1 - \rho_c$. Figure 6 shows the performance of the greedy algorithm for different ρ_c values. We observe that, as long as there are samples that humans can predict with low error, the greedy algorithm outsources them to humans and thus the overall performance improves. However, whenever the fraction of outsourced samples is higher than the fraction of samples with low human error, the performance degrades. This results in a characteristic U-shaped curve.

6 Conclusions

In this paper, we have initiated the development of machine learning models that are optimized to operate under different automation levels. We have focused on ridge regression under human assistance and shown that a simple greedy algorithm is able to find a solution with nontrivial approximation guarantees. Moreover, using both synthetic and real-world data, we have shown that this greedy algorithm beats several competitive baselines and is able to identify and outsource to humans those samples they can predict more accurately.

Our work also opens many interesting venues for future work. For example, it would be very interesting to advance the development of other more sophisticated machine learning models, both for regression and classification, under different automation levels. It would be valuable to find tighter lower bounds on the parameter α , which better characterize the good empirical performance. It would be very interesting to study sequential decision making scenarios under human assistance, *e.g.*, autonomous driving under different automation levels. Finally, we have assumed that we can measure the human error for every training sample. It would be interesting to tackle the problem under uncertainty.

A Proofs

Proof of Proposition 2. Let the feature space be \mathcal{F} . Moreover we denote that $\mathcal{X} = \{\mathbf{x}_i\}_{i \in \mathcal{V}}$ and $\mathcal{X}' = \{\mathbf{x}'_j\}_{j \in \mathcal{V}'}$. Then we denote that

$$H_{\mathbf{x}_i} = \cap_{k \in \mathcal{V}} \{\mathbf{x} \in \mathcal{F} \mid \|\mathbf{x}_i - \mathbf{x}\| \leq \|\mathbf{x}_k - \mathbf{x}\|\}. \quad (14)$$

⁷In this particular case, the patient suffered diabetic retinopathy, which is also characterized by yellow spots.

Hence, the set of test samples, which are nearest to \mathbf{x}_i , is denoted as $\mathcal{X}' \cap H_{\mathbf{x}_i}$. Since the features in \mathcal{X} and \mathcal{X}' are i.i.d random variables, $|\mathcal{X}' \cap H_{\mathbf{x}_i}|$ are also i.i.d random variables for different realizations of \mathcal{X} and \mathcal{X}' . Let us define $\vartheta = \mathbb{E}[|\mathcal{X}' \cap H_{\mathbf{x}_i}|]$. Hence we have, $\mathbb{E}[n'] = \sum_{i \in \mathcal{S}} \mathbb{E}[|\mathcal{X}' \cap H_{\mathbf{x}_i}|] = n\vartheta$ and $\mathbb{E}[|\mathcal{V}'| - n'] = (|\mathcal{V}'| - n)\vartheta$, which leads to the required result.

Proof of Lemma 5. By definition, we have that

$$\begin{aligned} \ell(\mathbf{w}^*(\mathcal{V}), \mathcal{V}) &= \sum_{i \in \mathcal{S}} c(\mathbf{x}_i, y_i) + c(\mathbf{x}_k, y_k) \\ \ell(\mathbf{w}^*(\mathcal{V} \setminus k), \mathcal{V} \setminus k) &= y_k^2 - y_k^2 \mathbf{x}_k^\top (\lambda \mathbb{I} + \mathbf{x}_k \mathbf{x}_k^\top)^{-1} \mathbf{x}_k \\ &\quad + \sum_{i \in \mathcal{S}} c(\mathbf{x}_i, y_i) \end{aligned}$$

Moreover, note that it is enough to prove that $\ell(\mathbf{w}^*(\mathcal{V}), \mathcal{V}) - \ell(\mathbf{w}^*(\mathcal{V} \setminus k), \mathcal{V} \setminus k) < 0$, without the logarithms, to prove the result. Then, we have that

$$\begin{aligned} &\ell(\mathbf{w}^*(\mathcal{V}), \mathcal{V}) - \ell(\mathbf{w}^*(\mathcal{V} \setminus k), \mathcal{V} \setminus k) \\ &= c(\mathbf{x}_k, y_k) - y_k^2 + y_k^2 \mathbf{x}_k^\top (\lambda \mathbb{I} + \mathbf{x}_k \mathbf{x}_k^\top)^{-1} \mathbf{x}_k \\ &\stackrel{(a)}{=} c(\mathbf{x}_k, y_k) - y_k^2 + y_k^2 \mathbf{x}_k^\top \left(\frac{1}{\lambda} \mathbb{I} - \frac{1}{\lambda^2} \frac{\mathbf{x}_k \mathbf{x}_k^\top}{1 + \frac{\mathbf{x}_k^\top \mathbf{x}_k}{\lambda}} \right) \mathbf{x}_k \\ &= c(\mathbf{x}_k, y_k) - y_k^2 + y_k^2 \frac{\mathbf{x}_k^\top \mathbf{x}_k}{\lambda} \left(1 - \frac{\mathbf{x}_k^\top \mathbf{x}_k}{\lambda + \mathbf{x}_k^\top \mathbf{x}_k} \right) \\ &< y_k^2 \left(\frac{\mathbf{x}_k^\top \mathbf{x}_k}{\lambda + \mathbf{x}_k^\top \mathbf{x}_k} - (1 - \gamma) \right) \\ &\stackrel{(b)}{<} y_k^2 \left(\frac{\mathbf{x}_k^\top \mathbf{x}_k}{\frac{\gamma \mathbf{x}_k^\top \mathbf{x}_k}{1 - \gamma} + \mathbf{x}_k^\top \mathbf{x}_k} - (1 - \gamma) \right) = 0, \end{aligned}$$

where equality (a) follows from Lemma 12 and inequality (b) follows from the lower bound on λ .

Proof of Theorem 6. Define $\Lambda_0 = \lambda |\mathcal{S}^c| \mathbb{I} + \mathbf{X}_{\mathcal{S}^c} \mathbf{X}_{\mathcal{S}^c}^\top$, $\Lambda_1 = \lambda |\mathcal{S}^c| \mathbb{I} + \mathbf{X}_{\mathcal{S}^c} \mathbf{X}_{\mathcal{S}^c}^\top - \lambda \mathbb{I} - \mathbf{x}_k \mathbf{x}_k^\top$ and $\Theta = \lambda \mathbb{I} + \mathbf{x}_k \mathbf{x}_k^\top$. Moreover, note that

$$\Lambda_1 = \Lambda_0 - \Theta \quad \text{and} \quad \Lambda_1^{-1} \stackrel{(a)}{=} \Lambda_0^{-1} + \underbrace{(\Lambda_0 \Theta^{-1} \Lambda_0 - \Lambda_0)^{-1}}_{\text{Define as } \Omega}$$

where equality (a) follows from Proposition 13. Then, it fol-

lows that

$$\begin{aligned}
\ell(\mathcal{S} \cup k) &= \sum_{i \in \mathcal{S}} c(\mathbf{x}_i, y_i) + c(\mathbf{x}_k, y_k) + \mathbf{y}_{\mathcal{S}^c}^\top \mathbf{y}_{\mathcal{S}^c} - y_k^2 \\
&\quad - (\mathbf{y}_{\mathcal{S}^c}^\top \mathbf{X}_{\mathcal{S}^c}^\top - y_k \mathbf{x}_k^\top) \mathbf{\Lambda}_1^{-1} (\mathbf{X}_{\mathcal{S}^c} \mathbf{y}_{\mathcal{S}^c} - y_k \mathbf{x}_k) \\
&\stackrel{(a)}{=} \sum_{i \in \mathcal{S}} c(\mathbf{x}_i, y_i) + c(\mathbf{x}_k, y_k) + \mathbf{y}_{\mathcal{S}^c}^\top \mathbf{y}_{\mathcal{S}^c} - y_k^2 \\
&\quad - \mathbf{y}_{\mathcal{S}^c}^\top \mathbf{X}_{\mathcal{S}^c}^\top \mathbf{\Lambda}_0^{-1} \mathbf{X}_{\mathcal{S}^c} \mathbf{y}_{\mathcal{S}^c} - \mathbf{y}_{\mathcal{S}^c}^\top \mathbf{X}_{\mathcal{S}^c}^\top \mathbf{\Omega}^{-1} \mathbf{X}_{\mathcal{S}^c} \mathbf{y}_{\mathcal{S}^c} \\
&\quad + 2y_k \mathbf{y}_{\mathcal{S}^c}^\top \mathbf{X}_{\mathcal{S}^c}^\top \mathbf{\Lambda}_1^{-1} \mathbf{x}_k^\top - y_k^2 \mathbf{x}_k^\top \mathbf{\Lambda}_1^{-1} \mathbf{x}_k \\
&= \ell(\mathcal{S}) + c(\mathbf{x}_k, y_k) - y_k^2 - \begin{bmatrix} \mathbf{y}_{\mathcal{S}^c}^\top \mathbf{X}_{\mathcal{S}^c}^\top & y_k \mathbf{x}_k^\top \end{bmatrix} \\
&\quad \begin{bmatrix} \mathbf{\Omega}^{-1} & -\mathbf{\Lambda}_1^{-1} \\ -\mathbf{\Lambda}_1^{-1} & \mathbf{\Lambda}_1^{-1} \mathbf{\Omega} \mathbf{\Lambda}_1^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{\mathcal{S}^c} \mathbf{y}_{\mathcal{S}^c} \\ y_k \mathbf{x}_k \end{bmatrix} \\
&\quad - y_k^2 \mathbf{x}_k^\top (\mathbf{\Lambda}_1^{-1} - \mathbf{\Lambda}_1^{-1} \mathbf{\Omega} \mathbf{\Lambda}_1^{-1}) \mathbf{x}_k \\
&\stackrel{(b)}{\leq} \ell(\mathcal{S}) + c(\mathbf{x}_k, y_k) - y_k^2 - y_k^2 \mathbf{x}_k^\top (\mathbf{\Lambda}_1^{-1} - \mathbf{\Lambda}_1^{-1} \mathbf{\Omega} \mathbf{\Lambda}_1^{-1}) \mathbf{x}_k \\
&\stackrel{(c)}{=} \ell(\mathcal{S}) + c(\mathbf{x}_k, y_k) - y_k^2 + y_k^2 \mathbf{x}_k^\top (\lambda \mathbb{I} + \mathbf{x}_k \mathbf{x}_k^\top)^{-1} \mathbf{x}_k \\
&\stackrel{(d)}{=} \ell(\mathcal{S}) + \ell(\mathcal{V}) - \ell(\mathcal{V} \setminus k),
\end{aligned}$$

where equality (a) follows from Proposition 13, inequality

(b) uses that $\begin{bmatrix} \mathbf{\Omega}^{-1} & -\mathbf{\Lambda}_1^{-1} \\ -\mathbf{\Lambda}_1^{-1} & \mathbf{\Lambda}_1^{-1} \mathbf{\Omega} \mathbf{\Lambda}_1^{-1} \end{bmatrix} \succcurlyeq 0$, equality (c) follows from the following observation:

$$\begin{aligned}
&(\mathbf{\Lambda}_1^{-1} - \mathbf{\Lambda}_1^{-1} \mathbf{\Omega} \mathbf{\Lambda}_1^{-1}) \\
&= (\mathbf{\Lambda}_0^{-1} + \mathbf{\Omega}^{-1}) - (\mathbf{\Lambda}_0^{-1} + \mathbf{\Omega}^{-1}) \mathbf{\Omega} (\mathbf{\Lambda}_0^{-1} + \mathbf{\Omega}^{-1}) \\
&= -\mathbf{\Lambda}_0^{-1} \mathbf{\Omega} \mathbf{\Lambda}_0^{-1} - \mathbf{\Lambda}_0^{-1} \\
&= -\mathbf{\Lambda}_0^{-1} (\mathbf{\Lambda}_0 \mathbf{\Theta}^{-1} \mathbf{\Lambda}_0 - \mathbf{\Lambda}_0) \mathbf{\Lambda}_0^{-1} - \mathbf{\Lambda}_0^{-1} = -\mathbf{\Theta}^{-1},
\end{aligned}$$

and inequality (d) follows from Lemma 5.

Proof of Proposition 7.

$$\begin{aligned}
&(y_k - \mathbf{x}_k^\top \mathbf{w}^*(k))^2 \\
&= (y_k - \mathbf{x}_k^\top \mathbf{w}^*(k))^2 + \lambda \|\mathbf{w}^*(k)\|^2 - \lambda \|\mathbf{w}^*(k)\|^2 \\
&= y_k^2 - y_k^2 \mathbf{x}_k^\top (\lambda \mathbb{I} + \mathbf{x}_k \mathbf{x}_k^\top)^{-1} \mathbf{x}_k - \lambda y_k^2 \mathbf{x}_k^\top (\lambda \mathbb{I} + \mathbf{x}_k \mathbf{x}_k^\top)^{-2} \mathbf{x}_k \\
&= y_k^2 - y_k^2 \frac{\mathbf{x}_k^\top \mathbf{x}_k}{\lambda + \mathbf{x}_k^\top \mathbf{x}_k} - \lambda y_k^2 \mathbf{x}_k^\top (\lambda \mathbb{I} + \mathbf{x}_k \mathbf{x}_k^\top)^{-2} \mathbf{x}_k \\
&\stackrel{(a)}{=} \frac{\lambda y_k^2}{\lambda + \mathbf{x}_k^\top \mathbf{x}_k} - \lambda y_k^2 \mathbf{x}_k^\top \left(\frac{1}{\lambda} \mathbb{I} - \frac{1}{\lambda^2} \frac{\mathbf{x}_k \mathbf{x}_k^\top}{1 + \frac{\mathbf{x}_k^\top \mathbf{x}_k}{\lambda}} \right)^2 \mathbf{x}_k \\
&= \frac{\lambda y_k^2}{\lambda + \mathbf{x}_k^\top \mathbf{x}_k} - \frac{y_k^2}{\lambda} \mathbf{x}_k^\top \left(\mathbb{I} - \frac{\mathbf{x}_k \mathbf{x}_k^\top}{\lambda + \mathbf{x}_k^\top \mathbf{x}_k} \right)^2 \mathbf{x}_k \\
&= y_k^2 \left(\frac{\lambda}{\lambda + \mathbf{x}_k^\top \mathbf{x}_k} \right)^2 \stackrel{(b)}{\leq} \rho^2 y_k^2,
\end{aligned}$$

where equality (a) follows from Lemma 12 and inequality (b) follows from the assumption $\lambda \leq \frac{\rho}{1-\rho} \max_i \|\mathbf{x}_i\|_2^2$. Finally, since $c(\mathbf{x}_k, y_k) > \rho^2 y_k^2$, we can conclude that $c(\mathbf{x}_k, y_k) > (y_k - \mathbf{x}_k^\top \mathbf{w}^*(k))^2$.

B Auxiliary lemmas and propositions

Proposition 10 Assume $c(\mathbf{x}_k, y_k) \leq \gamma y_k^2$ and $\lambda \geq \frac{\gamma}{1-\gamma} \|\mathbf{x}_k\|_2^2$. Then, $\begin{bmatrix} y_k^2 - c(\mathbf{x}_k, y_k) & y_k \mathbf{x}_k^\top \\ \mathbf{x}_k y_k & \lambda \mathbb{I} + \mathbf{x}_k \mathbf{x}_k^\top \end{bmatrix} \succcurlyeq 0$

Proof We use Schur complement property for positive-definiteness (Boyd et al. 1994, Page 8)

$$\begin{aligned}
&\text{on the matrix } \begin{bmatrix} y_k^2 - c(\mathbf{x}_k, y_k) & y_k \mathbf{x}_k^\top \\ \mathbf{x}_k y_k & \lambda \mathbb{I} + \mathbf{x}_k \mathbf{x}_k^\top \end{bmatrix} \\
&\lambda \mathbb{I} + \mathbf{x}_k \mathbf{x}_k^\top - \mathbf{x}_k \mathbf{x}_k^\top (y_k^2 / (y_k^2 - c(\mathbf{x}_k, y_k))) \\
&\succcurlyeq \lambda \mathbb{I} - \frac{\gamma}{1-\gamma} \mathbf{x}_k \mathbf{x}_k^\top.
\end{aligned} \tag{15}$$

Given that $\mathbf{x}_k \mathbf{x}_k^\top$ is a rank one matrix, it has only one non-zero eigenvalue. Hence it is same as $\text{tr}(\mathbf{x}_k \mathbf{x}_k^\top) = \|\mathbf{x}_k\|_2^2$, which along with the assumed bound on λ proves that $\lambda \mathbb{I} + \mathbf{x}_k \mathbf{x}_k^\top - \mathbf{x}_k \mathbf{x}_k^\top (y_k^2 / (y_k^2 - c(\mathbf{x}_k, y_k))) \succcurlyeq 0$. Then from Schur complement method, we have

$$\begin{bmatrix} y_k^2 - c(\mathbf{x}_k, y_k) & y_k \mathbf{x}_k^\top \\ \mathbf{x}_k y_k & \lambda \mathbb{I} + \mathbf{x}_k \mathbf{x}_k^\top \end{bmatrix} \succcurlyeq 0. \quad \blacksquare$$

Proposition 11 If $\mathbf{A} \succcurlyeq \mathbf{B}$ and \mathbf{B} and \mathbf{A} are both positive definite matrices, then $\det(\mathbf{A}) \geq \det(\mathbf{B})$.

Proof If $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ is the Cholesky factorization and since \mathbf{A} is strictly positive definite, \mathbf{L} has an inverse.

$$\begin{aligned}
\mathbf{A} \succcurlyeq \mathbf{B} &\implies \mathbb{I} \succcurlyeq \mathbf{L}^{-1} \mathbf{B} \mathbf{L}^{-\top} \succcurlyeq 0 \\
&\implies 1 > \text{eig}_i(\mathbf{L}^{-1} \mathbf{B} \mathbf{L}^{-\top}) > 0 \forall \text{ eigenvalues } \text{eig}_i \\
&\implies 1 > \prod_i \text{eig}_i(\mathbf{L}^{-1} \mathbf{B} \mathbf{L}^{-\top}) \\
&\implies 1 > \det(\mathbf{L}^{-1} \mathbf{B} \mathbf{L}^{-\top}) = (1/\det(\mathbf{A}))(\det(\mathbf{B}))
\end{aligned}$$

which immediately gives the required result. \blacksquare

Lemma 12 ((Sherman and Morrison 1950)) Assume \mathbf{A} is an invertible matrix. Then, the following equality holds:

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^\top)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1} \mathbf{u}\mathbf{v}^\top \mathbf{A}^{-1}}{1 + \mathbf{v}^\top \mathbf{A}^{-1} \mathbf{u}} \tag{17}$$

Proposition 13 Assume \mathbf{A} and \mathbf{B} are invertible matrices. Then, the following equality holds:

$$(\mathbf{A} - \mathbf{B})^{-1} = \mathbf{A}^{-1} + (\mathbf{A}\mathbf{B}^{-1}\mathbf{A} - \mathbf{A})^{-1} \tag{18}$$

Proof We observe that, $(\mathbf{A}\mathbf{B}^{-1}\mathbf{A} - \mathbf{A}) = (\mathbf{A}\mathbf{B}^{-1}\mathbf{A} - \mathbf{A})\mathbf{A}^{-1}(\mathbf{A} - \mathbf{B}) + (\mathbf{A} - \mathbf{B})$. Pre-multiply by $(\mathbf{A}\mathbf{B}^{-1}\mathbf{A} - \mathbf{A})^{-1}$ and post-multiply by $(\mathbf{A} - \mathbf{B})^{-1}$ on both sides to get the result. \blacksquare

Proposition 14 The function $t(x) = \frac{\log(x-a)}{\log x}$ is increasing for $x > a + 1$.

Proof

$$dt(x)/dx = \frac{x \log x - (x-a) \log(x-a)}{x(x-a)(\log x)^2} > 0 \tag{19}$$

References

- Bartlett, P. L., and Wegkamp, M. H. 2008. Classification with a reject option using a hinge loss. *JMLR* 9(Aug):1823–1840.
- Bhatia, K.; Jain, P.; Kamalaruban, P.; and Kar, P. 2017. Consistent robust regression. In *NeurIPS*, 2110–2119.
- Bogunovic, I.; Zhao, J.; and Cevher, V. 2018. Robust maximization of non-submodular objectives. *arXiv preprint arXiv:1802.07073*.
- Boyd, S.; El Ghaoui, L.; Feron, E.; and Balakrishnan, V. 1994. *Linear matrix inequalities in system and control theory*, volume 15. Siam.
- Chen, X., and Price, E. 2017. Active regression via linear-sample sparsification. *arXiv preprint arXiv:1711.10051*.
- Cheng, J.; Danescu-Niculescu-Mizil, C.; and Leskovec, J. 2015. Antisocial behavior in online discussion communities. In *ICWSM*.
- Cohn, D. A.; Ghahramani, Z.; and Jordan, M. I. 1995. Active learning with statistical models. In *NeurIPS*, 705–712.
- Cortes, C.; DeSalvo, G.; and Mohri, M. 2016. Learning with rejection. In *ALT*, 67–82. Springer.
- Davidson, T.; Warmlesley, D.; Macy, M.; and Weber, I. Automated hate speech detection and the problem of offensive language. *ICWSM*, 512–515.
- Decencière, E.; Zhang, X.; Cazuguel, G.; Lay, B.; Cochener, B.; Trone, C.; Gain, P.; Ordonez, R.; Massin, P.; Erginay, A.; Char-ton, B.; and Klein, J.-C. 2014. Feedback on a publicly distributed database: the messidor database. *Image Analysis & Stereology* 33(3):231–234.
- Feng, J.; Xu, H.; Mannor, S.; and Yan, S. 2014. Robust logistic regression and classification. In *NeurIPS*, 253–261.
- Gatmiry, K., and Gomez-Rodriguez, M. 2019. Non-submodular function maximization subject to a matroid constraint, with applications. *arXiv preprint arXiv:1811.07863*.
- Geifman, Y., and El-Yaniv, R. 2019. Selectivenet: A deep neural network with an integrated reject option. *arXiv preprint arXiv:1901.09192*.
- Geifman, Y.; Uziel, G.; and El-Yaniv, R. 2018. Bias-reduced uncertainty estimation for deep neural classifiers.
- Guo, Y., and Schuurmans, D. 2008. Discriminative batch mode active learning. In *NeurIPS*, 593–600.
- Harshaw, C.; Feldman, M.; Ward, J.; and Karbasi, A. 2019. Sub-modular maximization beyond non-negativity: Guarantees, fast algorithms, and applications. *arXiv preprint arXiv:1904.09354*.
- Hashemi, A.; Ghasemi, M.; Vikalo, H.; and Topcu, U. 2019. Submodular observation selection and information gathering for quadratic models. *arXiv preprint arXiv:1905.09919*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.
- Hoi, S. C.; Jin, R.; Zhu, J.; and Lyu, M. R. 2006. Batch mode active learning and its application to medical image classification. In *ICML*.
- Hoover, A.; Kouznetsova, V.; and Goldbaum, M. 2000. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical imaging* 19(3):203–210.
- Iyer, R., and Bilmes, J. 2012. Algorithms for approximate minimization of the difference between submodular functions, with applications. *arXiv preprint arXiv:1207.0560*.
- Joulin, A.; Grave, E.; Bojanowski, P.; Douze, M.; Jégou, H.; and Mikolov, T. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.
- Lehmann, B.; Lehmann, D.; and Nisan, N. 2006. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior* 55(2):270–296.
- Liu, Z.; Wang, Z.; Liang, P. P.; Salakhutdinov, R. R.; Morency, L.-P.; and Ueda, M. 2019. Deep gamblers: Learning to abstain with portfolio theory. In *NeurIPS*.
- Pradel, M., and Sen, K. 2018. Deepbugs: A learning approach to name-based bug detection. *Proceedings of the ACM on Programming Languages* 2(OOPSLA):147.
- Raghu, M.; Blumer, K.; Corrado, G.; Kleinberg, J.; Obermeyer, Z.; and Mullainathan, S. 2019a. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220*.
- Raghu, M.; Blumer, K.; Sayres, R.; Obermeyer, Z.; Kleinberg, B.; Mullainathan, S.; and Kleinberg, J. 2019b. Direct uncertainty prediction for medical second opinions. In *ICML*.
- Ramaswamy, H. G.; Tewari, A.; Agarwal, S.; et al. 2018. Consistent algorithms for multiclass classification with an abstain option. *Electronic J. of Statistics* 12(1):530–554.
- Sabato, S., and Munos, R. 2014. Active regression by stratification. In *NeurIPS*, 469–477.
- Sherman, J., and Morrison, W. J. 1950. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics* 21(1):124–127.
- Studer, C.; Kuppinger, P.; Pope, G.; and Bolcskei, H. 2011. Recovery of sparsely corrupted signals. *IEEE Transactions on Information Theory* 58(5):3115–3130.
- Suggala, A. S.; Bhatia, K.; Ravikumar, P.; and Jain, P. 2019. Adaptive hard thresholding for near-optimal consistent robust regression. *arXiv preprint arXiv:1903.08192*.
- Sugiyama, M. 2006. Active learning in approximately linear regression based on conditional expectation of generalization error. *JMLR* 7(Jan):141–166.
- Thulasidasan, S.; Bhattacharya, T.; Bilmes, J.; Chennupati, G.; and Mohd-Yusof, J. 2019. Combating label noise in deep learning using abstention. *arXiv preprint arXiv:1905.10964*.
- Topol, E. J. 2019. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine* 25(1):44.
- Tsakonas, E.; Jaldén, J.; Sidiropoulos, N. D.; and Ottersten, B. 2014. Convergence of the huber regression m-estimate in the presence of dense outliers. *IEEE Signal Processing Letters* 21(10):1211–1214.
- Willett, R.; Nowak, R.; and Castro, R. M. 2006. Faster rates in regression via active learning. In *NeurIPS*.
- Wright, J., and Ma, Y. 2010. Dense error correction via ℓ^1 -minimization. *IEEE Transactions on Information Theory* 56(7):3540–3560.