

**SYNERGISTIC HUMAN-MACHINE PREDICTION: ACTIVE ERROR
ANALYSIS AND MITIGATION WITH GAUSSIAN PROCESS
REGRESSION**

by

Claudio César Claros Olivares

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical & Computer Engineering

Summer 2020

© 2020 Claudio César Claros Olivares
All Rights Reserved

**SYNERGISTIC HUMAN-MACHINE PREDICTION: ACTIVE ERROR
ANALYSIS AND MITIGATION WITH GAUSSIAN PROCESS
REGRESSION**

by

Claudio César Claros Olivares

Approved: _____
Austin J. Brockmeier, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____
Mark S. Mirotznik, Ph.D.
Interim Chair of the Department of Electrical and Computer Engineering

Approved: _____
Levi T. Thompson, Ph.D.
Dean of the College of Engineering

Approved: _____
Douglas J. Doren, Ph.D.
Interim Vice Provost for Graduate and Professional Education and
Dean of the Graduate College

ACKNOWLEDGEMENTS

I would like to acknowledge that this research was partially supported by the University of Delaware's Data Science Institute through a seed grant from the Unidel Foundation.

I would like express my sincere gratitude to my advisor, Dr. Brockmeier, for giving me the opportunity to be involved in research, sharing his insights on a wide variety of topics, engaging in fruitful discussions and guiding me through this process of earning a master's degree. His wisdom, enthusiasm, encouragement and support on my academic and personal development have been instrumental to reach the culmination of this program.

My appreciation also extends to my fellow labmates in the Computational Neural and Information Engineering Laboratory: Hassan, Bilal, Yuksel, and Carlos for helping me improve my work. I am very grateful to the Electrical & Computer Engineering department, all faculty members, and staff at the University of Delaware for encouragement and help.

Nobody has been more important to me in the pursuit of this project than the members of my family. I would like to thank my parents, Isabel and Walter, whose love and guidance are with me in whatever I pursue. To my brother, Abel, my sister, Claudia, and all family members, thank you for encouraging me in all of my pursuits. I also would like to thank my supportive girlfriend, Lorena, for inspiring me to follow my dreams and being with me at each moment throughout this path.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	viii
ABSTRACT	ix
 Chapter	
1 INTRODUCTION	1
2 MOTIVATION	3
3 RELATED WORK	7
3.1 Contributions	8
4 PRELIMINARIES	10
4.1 Learning Setting	10
5 SYNERGISTIC HUMAN-MACHINE PREDICTION	13
5.1 Signaling Function as Loss Interpolation	14
6 EMPIRICAL ESTIMATION AND TESTING	18
6.1 Hypothesis testing	19
6.2 Performance Metrics	20
6.3 Baselines for comparison	21
7 RESULTS	22
7.1 Datasets and Models	22
7.2 Training and evaluation procedure	24
7.3 Classification tasks	25
7.4 Regression tasks	28

8	DISCUSSION	31
8.1	Limitations	33
8.2	Future work	34
9	CONCLUSION	35
	BIBLIOGRAPHY	36
	Appendix	
A	FULL RESULTS	42
A.1	Classification tasks	42
A.2	Regression tasks	43

LIST OF TABLES

7.1	Datasets characteristics	23
7.2	NN architectures	24
A.1	Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \mathbf{x}_n^{(train)}, \boldsymbol{\psi}_n^{(train)} \right\}$ for classification tasks	44
A.2	Loss reduction comparison in test set with $f(x)$ trained on $\Delta = \left\{ \mathbf{x}_n^{(train)}, \boldsymbol{\psi}_n^{(train)} \right\}$ compared against $g(x)$	45
A.3	Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \left(\mathbf{x}_n^{(train)}, \hat{h}(\mathbf{x}_n^{(train)}) \right), \boldsymbol{\psi}_n^{(train)} \right\}$ for classification tasks	46
A.4	Loss reduction comparison in test set with $f(x)$ trained on $\Delta = \left\{ \left(\mathbf{x}_n^{(train)}, \hat{h}(\mathbf{x}_n^{(train)}) \right), \boldsymbol{\psi}_n^{(train)} \right\}$ compared against $g(x)$	47
A.5	Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \mathbf{T} \left(\mathbf{x}_n^{(train)}, \hat{h}(\mathbf{x}_n^{(train)}) \right), \boldsymbol{\psi}_n^{(train)} \right\}$ for classification tasks	48
A.6	Loss reduction comparison in test set with $f(x)$ trained on $\Delta = \left\{ \mathbf{T} \left(\mathbf{x}_n^{(train)}, \hat{h}(\mathbf{x}_n^{(train)}) \right), \boldsymbol{\psi}_n^{(train)} \right\}$ compared against $g(x)$	49
A.7	Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ (\sigma_p \circ \dots \circ \sigma_1) \left(\mathbf{x}_n^{(train)} \right), \boldsymbol{\psi}_n^{(train)} \right\}$ for classification tasks	50

A.8	Loss reduction comparison in test set with $f(x)$ trained on $\Delta = \left\{ (\sigma_p \circ \dots \circ \sigma_1) \left(\mathbf{x}_n^{(train)} \right), \boldsymbol{\psi}_n^{(train)} \right\}$ compared against $g(x)$. . .	51
A.9	Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \left((\sigma_p \circ \dots \circ \sigma_1) \left(\mathbf{x}_n^{(train)} \right), \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \boldsymbol{\psi}_n^{(train)} \right\}$ for classification tasks	52
A.10	Loss reduction comparison in test set with $f(x)$ trained on $\Delta = \left\{ \left((\sigma_p \circ \dots \circ \sigma_1) \left(\mathbf{x}_n^{(train)} \right), \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \boldsymbol{\psi}_n^{(train)} \right\}$ compared against $g(x)$	53
A.11	Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \mathbf{T} \left((\sigma_p \circ \dots \circ \sigma_1) \left(\mathbf{x}_n^{(train)} \right), \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \boldsymbol{\psi}_n^{(train)} \right\}$ for classification tasks	54
A.12	Loss reduction comparison in test set with $f(x)$ trained on $\Delta = \left\{ \mathbf{T} \left((\sigma_p \circ \dots \circ \sigma_1) \left(\mathbf{x}_n^{(train)} \right), \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \boldsymbol{\psi}_n^{(train)} \right\}$ compared against $g(x)$	55
A.13	Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \mathbf{x}_n^{(train)}, \boldsymbol{\psi}_n^{(train)} \right\}$ for regression tasks	56
A.14	Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \left(\mathbf{x}_n^{(train)}, \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \boldsymbol{\psi}_n^{(train)} \right\}$ for regression tasks	57
A.15	Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \mathbf{T} \left(\mathbf{x}_n^{(train)}, \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \boldsymbol{\psi}_n^{(train)} \right\}$ fore regression tasks	58

LIST OF FIGURES

2.1	Examples of systematic sources of error produced by suboptimal decision boundaries.	4
2.2	Depiction of how traditional and modern machine learning regimes can originate sources of systematic mistakes	6
5.1	Decision surface of the best-performing SVM classifier with polynomial kernel trained on Social Network Ads dataset	15
7.1	Summary statistics for Group A with budget $\rho = 0.15$	26
7.2	Summary statistics for Group B with budget $\rho = 0.15$	27
7.3	Comparison of the top 8 selected instances by f and g in Fashion MNIST and CIFAR10 datasets	29
7.4	Summary statistics for regression tasks with budget $\rho = 0.15$	30

ABSTRACT

Before deployment, a machine learning model is evaluated on both training and validation sets. Assuming the latter is a representative sample, the validation performance offers an estimate of how well the model will perform on a test set. Even if the performance meets specifications, there may be cases of systematic errors caused by model underfitting, poor model design or even overfitting. We propose to perform error analysis of the training and validation set during deployment to alert a user when instances similar to previous systematic errors arise. Triggering user vigilance during deployment will improve the synergistic operation of the machine and the user. Our model-agnostic approach interpolates the distribution of errors (taking cues from both the training and validation sets), optimizes the threshold for alerting a user, and requests verification for possibly erroneous predictions that exceed the threshold. Under the assumption that the user would make the correct decision, the approach is evaluated by the reduction of loss, while seeking to maintain a budget of interventions. The framework is tested on illustrative examples and real-world data sets where machine learning models have systematic errors. We conclude with a discussion of the limitations and areas for future work.

Chapter 1

INTRODUCTION

Machine learning, in most cases, can be regarded as a human-labor-saving technology. Naturally, humans are error-prone especially when it comes to repetitive tasks and, even experts, fail to spot particularly unanticipated yet prominent events [12]. Understanding this sort of human limitations is pivotal to consider a machine in decision-making processes such that the burden for humans is eased when a machine can clearly outperform human capabilities [4, 26, 15, 2]. However, that is just a unilateral statement which ignores some limitations associated with the inherent assumptions made throughout the analysis of machine learning algorithms. In that vein, machines also make mistakes that most of the time are traded off in favor of performance, contemplating underrepresented phenomena as noise in the samples provided to train it. While these mistakes can be harmless in various scenarios, high-stake decisions, such as predicting recidivism [19] or cancer risk [14], require frequently more examination. Obviously, in one side of the spectrum, assessing every machine decision annihilates the predictor's main purpose, and, in the other side, trusting every machine prediction confers too much power to an entity whose decision process is often times not fully understood. It is clear that some degree of human intervention is required to tip the balance towards safer AI decision-making.

When machine learning models are deployed in important decision-making, it is crucial to understand their limitations and monitor their performance to identify possible systematic errors they may produce [2]. In practice, a validation set is used for model selection and determining if a model is acceptable for deployment. Even if the performance meets specifications there may be cases of systematic erroneous predictions caused by model underfitting, inadequate model design, or overfitting,

among other phenomena regarding the probability distributions that generated the samples. Errors could be scrutinized by examining the training set, but it is unclear how this retrospection will be useful since the ground truth is already known for these examples. Why not use the information in the training set and validation set, especially the distribution of the empirical loss, in deployment?

In this context, we consider a human-in-the-loop scenario during machine learning deployment (fixed predictor). The human user, who is expected to be a domain expert¹, should be signaled to manually examine an instance when there is undue risk in trusting the machine prediction. Learning when to signal the user is itself a machine learning problem, with the goal of the user boosting the joint decision making process, while still benefiting from the automated decision making a majority of the time.

Therefore, we propose a method which can: (i) detect a ‘confident’ yet erroneous prediction, (ii) identify those samples from the test set that seem to generate confusion for the fixed predictor, and (iii) incorporate desired degrees of human intervention for potential correction. We consider the selection of signaling function via empirical risk minimization and show empirically that our signaling mechanism chooses non-trivial instances.

¹ In this work the expressions human expert, expert or oracle are used interchangeably.

Chapter 2

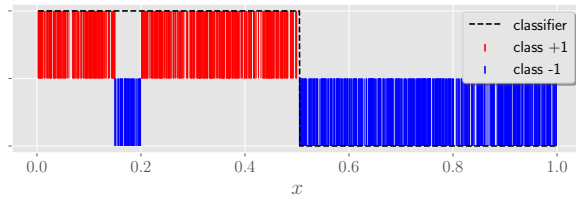
MOTIVATION

AI systems are playing more and more important roles regarding decision-making processes in our societies. It is hard to ignore the benefits of automated systems that could operate in a fraction of the time that a human would; however, some decisions are too important to let the machine take them alone. At the same time, having a human scrutinize all the decisions taken by the machine undermines the whole purpose of an automated system. Deciding whether a human should be placed *in the loop* is a challenging problem that involves multiple considerations such as the system’s main goal as well as its application domain. In this sense, a synergistic human-machine strategy may be beneficial to provide more acceptable decisions.

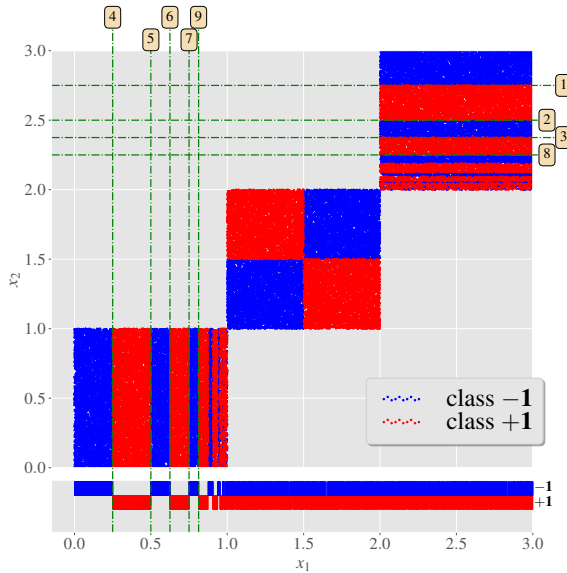
In this chapter, we use synthetic distributions to illustrate some scenarios in which machines generate systematic sources of errors even when the predictor’s accuracy may suggest that there is no need of human intervention.

The examples in Figure 2.1 are two distributions that aim to highlight how poor model design and complex input spaces can lead to systematic sources of error. The problem depicted in Fig 2.1a may look inoffensive considering that the optimal predictor in the hypothesis class $\mathcal{H} = \left\{ x \mapsto b \cdot \text{sgn}(\theta - x) : \theta \in \mathbb{R}, b \in \{-1, 1\} \right\}$ classifies correctly most of the instances. However, the small region that has been left out, which could be relevant in high-stakes decision making, is a source of systematic errors and a byproduct of the optimal decision boundary. The scenario in Figure 2.1b, originally introduced by Biau et al. [6], shows the case where a random forest, a randomized ensemble technique that has the ability to produce complex decision boundaries by breaking the input space in smaller regions, creates systematic errors in the center square. In this case, the predictor seems adequate, nevertheless it yields an expected error rate greater than

$\frac{1}{6} = \frac{1}{3} \cdot \frac{1}{2}$. Clearly, in both cases the expected error rate will not improve with a bigger sample size due to the limited ability of the chosen hypothesis classes.



(a) Poor model design



(b) Complex input space

Figure 2.1: In (a) decision stumps are insufficient to accurately classify this data distribution. The optimal predictor in the hypothesis class \mathcal{H} will achieve an accuracy of 95%. Analyzing the accuracy alone, a user might conclude that the error rate is due to intrinsic noise, but in reality it is due to the insufficient choice of hypothesis class. Plot (b) depicts a degenerate example for random forests [6] with a countably infinite number of class boundaries in the first and last unit squares defined by a geometric progression. The greedy selection of splits in a random forest algorithm will choose splits in the first and last squares (even if the choice of coordinate is selected randomly or greedily), since at each stage there is a split that can create a pure leaf with examples from only one class. In comparison, any axis-aligned split in the center square will not separate the classes. Numbered dashed lines indicate the sequence of splits.

These kind of limitations become even more apparent due to traditional machine learning practices that favor simpler explanations over complicated ones. However, modern machine learning techniques are much better at learning complex input spaces and can achieve impressive results if provided with enough samples, without abiding by the Occam's razor principle. In a thought-provoking explanation, Belkin et al. [5] attributes the success of modern techniques such as deep networks, kernel machines, boosting and random forests to interpolation. These methods manage to obtain (near-)zero training error and at the same time achieve good performance in the test set, defying some traditional concepts in statistical learning that advice against perfect

accuracy during training. However, even in this modern machine learning regime, interpolated predictors can occasionally generate sources of systematic mistakes.

Our analysis stems from the insights and experiments developed by Wyner et al. [42] which help us elaborate on how traditional and modern machine learning regimes are prone to yield regions of systematic failure. Figure 2.2a shows a simulation in which we want to learn a signal represented by a circle with radius $r = 0.4$ centered inside the square $[0, 1]^2$, where the probability of $y = +1$ outside the circle is 0.9 and inside is 0.1. The training points $n = 1000$ are sampled uniformly on the square with the Latin Hypercube design. On the one hand, to elucidate the case for a traditional statistical model, we use a decision tree pruned via cross-validation with the CART algorithm. Figure 2.2b reveals that the predictor found through CART does not achieve enough complexity to capture the circular pattern given that the classifier follows the principle of parsimony by design, which limits its out-of-sample performance. On the other hand, a Random Forest classifier with 500 trees exemplifies the modern regime in which interpolation gives rise to highly competitive predictors. The decision surface produced by this classifier is the product of the interpolating capability of an individual full-grown tree and the self-averaging mechanism of the forest. Figure 2.2c demonstrates that this combination allows a higher complexity with superior out-of-sample performance. In both traditional and modern regimes, we can observe that the trained predictors originate regions in which test points get classified incorrectly. These regions are sources of systematic errors that the machine alone is not able to characterize, yet they are interpretable by human experts.

Naturally, most conflictive regions can be linked to their proximity to the decision boundary, but not all of them. Particularly, Figure 2.2c shows that some small misclassified areas are distant from the decision boundary. Some mechanisms such as entropy and margin distance can be interpreted as confidence level that a predictor has on a given test input; however, they are not totally reliable because test instances inside these areas would be erroneously predicted and yet the classifier would be confident about its prediction. Identifying these regions to alert a human expert about a confident

yet potentially mistaken prediction is what drives this work.

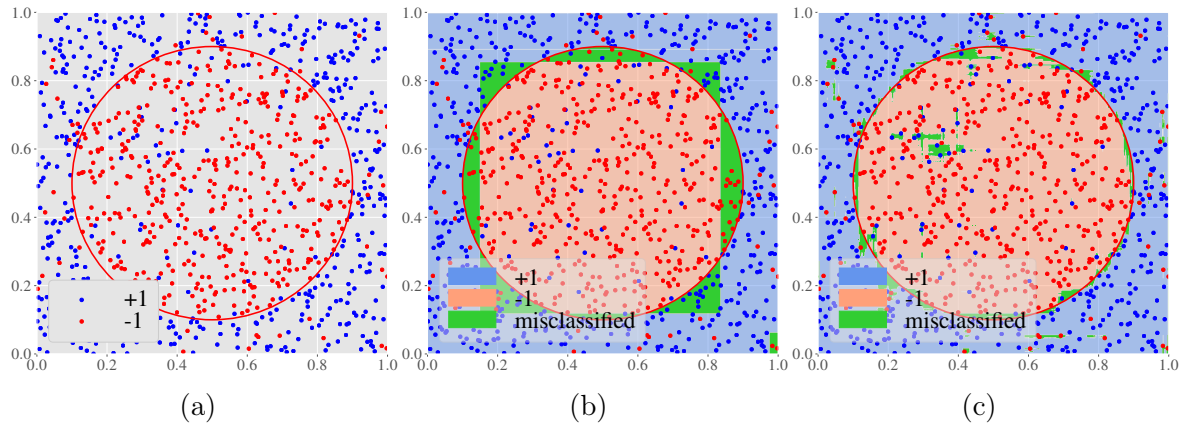


Figure 2.2: Depiction of how traditional and modern machine learning regimes can originate sources of systematic mistakes. (a) Training points sampled uniformly from $[0, 1]^2$, where $\mathbb{P}[y = +1|\mathbf{x}] = 0.9$ outside the circle. (b) Decision surface generated by a decision tree pruned via cross-validation for which the simplest explanation prevails, limiting its complexity. (c) Decision surface produced by a Random Forest model with 500 trees that captures correctly most of the circular pattern due to its interpolating nature.

The examples developed in this chapter may appear trivial considering their low dimensionality. Nevertheless, similar deficiencies are not obvious in higher-dimensional feature spaces. Tasks such as image classification or sentiment analysis operate in highly complex spaces where modern techniques achieve competitive accuracy levels, but do not assess their predictions thoroughly. Entropy and margin distance fail to prioritize regions where there is evidence of systematic erroneous predictions that are not adjacent to the decision boundary. In this cases, identifying such regions and allowing some degree of human intervention can benefit the decision-making process.

Chapter 3

RELATED WORK

Existing machine learning approaches analyze how to best incorporate human knowledge into their algorithms under different scenarios and conjectures. Namely **active learning** [33], surveyed thoroughly by Aggarwal et al. and Ramirez-Loaiza et al. [30], focuses on incorporating human decisions to train a model that maximizes predictive performance posing as few queries as possible. Unlike this scenario, we consider for our analysis a model-agnostic approach with a fixed predictor during deployment, where the signaling function decides if machine predictions are trustworthy or if they need to be shown to a human expert.

Also, some connections can be drawn between our proposed scenario and well-known problems in ML literature. If we consider issues associated with the probability distributions that originated the samples, we can identify similarities with **dataset shift detection**. Specifically, covariate shift $q_{XY}(x, y) = q_X(x)p_{Y|X}(y|x)$ [17, 39, 36] and label shift $q_{XY}(x, y) = q_Y(y)p_{X|Y}(x|y)$ [31, 38, 44] detection, where p and q are source and target distribution, respectively. Particularly, Lipton et al. [23] propose a method called *black box shift detection* (BBSD) which shows that detecting if p and q are different, given a fixed predictor h , only requires detecting that $h(X)|X \sim p \neq h(X)|X \sim q$. Rabanser et al. [28] contrast BBSD against other techniques in an empirical study. Although underrepresentation is a feature that dataset shift and our framework have in common, we consider for this work that the distribution remains the same during training, testing and deployment, and the problem resides in inadequacies of the model rather than with the data distribution.

Moreover, if given only one example from the test data, our proposed paradigm bears resemblance with **anomaly detection**, surveyed by Chalapathy et al. [9] for

deep learning settings and by Chandola et al. [10] for more general settings. Density estimation [7], margin-based [32], and tree-based [24] approaches are amongst the most popular to address this issue. Similarly, **outlier detection** (also known as *out-of-distribution (OOD) sample detection* in modern settings) might share some characteristics with our framework. In particular, Shafaei et al. [35] explore a variety of detection techniques applied to deployed systems and introduces a three-dataset evaluation scheme to address biased (overoptimistic) detection processes. While anomaly and outlier detection methods aim to improve data quality and prevent unpredictable behaviour in deployed systems, they do not include degrees of human intervention for correcting identified instances.

Finally, the notion of **uncertainty** can also be a valid strategy to signal instances. Lakshminarayanan et al. [21] show that an ensemble of neural networks (NN) trained with an adversarial strategy provides predictive uncertainty. Qiu et al. [27] propose to add uncertainty to NN point predictions in regression tasks by fitting prediction residuals with a Gaussian process (\mathcal{GP}) regression that captures task-relevant information. Despite the fact that uncertainty quantification is useful as a detection mechanism and that it can be obtained without adding Bayesian approaches into the training pipeline, it is not naturally associated with how much a human is allowed to intervene for a given level of uncertainty.

3.1 Contributions

In this thesis, we propose a method for signaling instances that incorporates in its calculation the proportion of desired human intervention, and applies to any fixed (pre-trained) predictor. The high-level idea for this method is to exploit the statistical significance of the empirical loss distribution by coupling its task-relevant information in the input space through a Gaussian process regression. First, we show that, if provided with a signaling function f used to alert the oracle when its evaluation at instance x has surpassed certain threshold η , it is always possible to improve performance even when the oracle is noisy (Chapter 5). This analysis gives rise to an intuitive formulation that

can be cast as an optimization problem to find such f and η . Moreover, this formulation allows the user to introduce the level of human intervention ρ as a constraint defined by the user.

Next, we introduce a general formulation for the signaling function f as an interpolator function and propose an algorithm to solve the optimization problem (Chapter 6). Specifically, we assume that systematic sources of erroneous predictions can be found across the training, validation, and test set. Under this assumption, we compute a family of suitable interpolating functions \mathcal{F} on the training set; optimize the selection mechanism (variables η and signaling function $f \in \mathcal{F}$) on the validation set according to the required degree of human inspection ρ ; and evaluate the performance on the test set, following the three-dataset evaluation scheme proposed by Shafaei et al. [35].

Our results suggest that the selection mechanism that we propose ($f(X) \geq \eta$) presents an edge over *model-aware* selection approaches (Chapter 7). Particularly, we consider probability-based and margin-based predictions as baseline measurements that can inform a user about the presence of a diffident decision. Interestingly, our approach selects instances that are not only problematic decisions in terms of the model output, but also poorly learned instances that the fixed predictor would predict confidently. Along with these results, we demonstrate empirically that our signaling function picks instances for correction at a better rate than a random selection.

Finally, we discuss some effects of kernel selection on relation to the dimensionality of the problem as well as limitations that were identified. Also, we draw some connections with adaptation domain developments and present future work in this area (Chapter 8).

Extended explanations of the experimental setup and results are presented in the Appendix A.

Chapter 4

PRELIMINARIES

In this chapter, we provide an overview of fundamental concepts about statistical learning.

4.1 Learning Setting

A prototypical learning task characterization considers a probability space $(\mathcal{Z}, \mathcal{D})$ where $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ in a supervised learning setting, with \mathcal{X} as the instance domain (a measurable space) and \mathcal{Y} as the target domain (a closed subset of \mathbb{R}), and where \mathcal{D} is a fixed and unknown probability distribution that can be seen only through a training set $\mathbf{z}_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$, with $n \in \mathbb{N}$ as the sample size. Strictly speaking, in this probabilistic framework, \mathbf{z}_n is a realization of the random sample $\mathbf{Z}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\} \in \mathcal{Z}^n$, where the pairs $Z_i = (X_i, Y_i)$ are random variables independent and identically distributed (i.i.d.) according to \mathcal{D} . Given these elements, the goal of a learning algorithm \mathbf{A} is to generate a mapping that takes the random sample \mathbf{Z}_n as input and outputs a function $\hat{h} \in \mathcal{H}$, where \mathcal{H} is known as the hypothesis set. More formally,

$$\mathbf{A} : \bigcup_{n \in \mathbb{N}} \mathcal{Z}^n \rightarrow \mathcal{H}, \quad \mathbf{Z}_n \mapsto \hat{h} = \mathbf{A}(\mathbf{Z}_n), \quad (4.1)$$

where the function \hat{h} can, roughly speaking, be devised to perform two different tasks: prediction or estimation. For prediction tasks, we are interested in minimizing the expected risk

$$\mathcal{L}(\hat{h}) = \mathbb{E}_{Z \sim \mathcal{D}} \ell(\hat{h}, Z) = \mathbb{E}_{(X, Y) \sim \mathcal{D}} \ell(\hat{h}, (X, Y)), \quad (4.2)$$

where $\ell : \mathcal{H} \times \mathcal{Z} \rightarrow \mathbb{R}_{\geq 0}$ is the loss function. This last expression measures the performance of the predictor \hat{h} , derived from \mathbf{z}_n , by evaluating a fresh sample Z (drawn from the same distribution \mathcal{D} that generated \mathbf{Z}_n) with the loss function ℓ hoping to get a small value in expectation. In the other hand, estimation tasks seek to minimize the risks of the form

$$\mathcal{L}(\hat{h}, h^*) = \left\| \hat{h} - h^* \right\|_{L^p(\mathcal{D})}, \quad (4.3)$$

where $h^*(x) = \mathbb{E}_{Y|X \sim \mathcal{D}_{Y|X}}[Y|X = x]$, considering a regression problem as example. In this case, h^* is a parameter that depends on \mathcal{D} , and the measure of performance is defined in terms of the normed difference of the estimator \hat{h} with respect to h^* hoping to be small. In this work, we focus on prediction tasks mainly, but estimation and prediction can be equivalent under some conditions.

For both prediction and estimation problems, we could try to understand the probability distribution \mathcal{D} and the quantities associated to it in order to infer the desired result. However, our connection with \mathcal{D} is only through \mathbf{z}_n which can be thought as a window restricted by the sample size n . Therefore, this is in general a harder problem to tackle and that can be unnecessary if the problem at hand requires a simpler solution such as differentiating classes within a probability distribution.

It is clear that following a path towards comprehending \mathcal{D} leads to a more complex problem than required. Nevertheless, the measure of performance \mathcal{L} for a proposed predictor \hat{h} is defined assuming that we have access to \mathcal{D} . Instead, we can use an approximation, called empirical risk, which takes into account the information \mathcal{S}_n that is actually available to us. Since this analysis must be general enough to be performed for any data set, we will define the empirical risk in terms of \mathbf{Z}_n ,

$$\hat{\mathcal{L}}(h) = \frac{1}{n} \sum_{i=1}^n \ell(h, Z_i) = \frac{1}{n} \sum_{i=1}^n \ell\left(h, (X_i, Y_i)\right). \quad (4.4)$$

Analyzing the performance of the predictor \hat{h} is typically straightforward if we can derive a closed-form solution. However, most learning procedures cannot be

defined as analytical expressions in terms of a finite number of certain functions, but as optimization problems,

$$\min_{x \in \mathcal{K}} g(x), \tag{4.5}$$

where g is the objective function we want to minimize and x is the optimization variable constrained to be in some set \mathcal{K} .

Chapter 5

SYNERGISTIC HUMAN-MACHINE PREDICTION

Let $f : \mathcal{X} \rightarrow \mathbb{R}$ denote the signaling function. When the evaluation of f at X is greater than a threshold value η , the expert is alerted and asked to provide a prediction $h_*(X)$. If the expert intervenes, the synergistic system incurs some loss $L(h_*, Z)$; if he does not, but the deployed predictor \hat{h} outputs an incorrect prediction, the synergistic system incurs a loss $L(\hat{h}, Z)$; otherwise, it suffers no loss. Therefore, we can express the synergistic loss as $\ell_{f,\eta}(\hat{h}, h_*, Z) = L(h_*, Z)\mathbb{I}_{\{f(X) > \eta\}} + L(\hat{h}, Z)\mathbb{I}_{\{f(X) \leq \eta\}}$, where $L : \mathcal{H} \times \mathcal{Z} \rightarrow \mathbb{R}_{\geq 0}$ denotes a non-negative, bounded testing loss function (possibly distinct from the loss used to train the deployed predictor). Assuming the expert's predictions $h_*(X)$ are perfect, then the associated loss is exactly zero and the synergistic risk is

$$\begin{aligned} \mathbb{E} \left[\ell_{f,\eta}(\hat{h}, h_*, Z) \right] &= \mathbb{E} \left[L(h_*, Z)\mathbb{I}_{\{f(X) > \eta\}} + L(\hat{h}, Z)\mathbb{I}_{\{f(X) \leq \eta\}} \right] = \mathbb{E} \left[L(\hat{h}, Z)\mathbb{I}_{\{f(X) \leq \eta\}} \right] \\ &\leq \mathbb{E} \left[L(\hat{h}, Z) \right], \end{aligned} \quad (5.1)$$

where \mathbb{I}_A denotes the indicator random variable with $\mathbb{I}_A = 1$, if A occurs, and $\mathbb{I}_A = 0$, otherwise. Synergistic performance will be better than strictly machine prediction if the expert is ever signaled. Even if the expert makes mistakes, we assume the expert's expected loss is equivariant across the domain, and independent of f such that $\mathbb{E} \left[L(h_*, Z)\mathbb{I}_{\{f(X) > \eta\}} \right] = \mathbb{E} \left[L(h_*, Z) \right] \cdot \mathbb{E} \left[\mathbb{I}_{\{f(X) > \eta\}} \right]$, then

$$\begin{aligned} \mathbb{E} \left[\ell_{f,\eta}(\hat{h}, h_*, Z) \right] &= \mathbb{E} \left[L(h_*, Z)\mathbb{I}_{\{f(X) > \eta\}} + L(\hat{h}, Z)\mathbb{I}_{\{f(X) \leq \eta\}} \right] \\ &= L_* \mathbb{P}[f(X) > \eta] + \mathbb{E} \left[L(\hat{h}, Z)\mathbb{I}_{\{f(X) \leq \eta\}} \right], \end{aligned} \quad (5.2)$$

where $L_\star \geq 0$ is the expected loss from the expert. Synergistic performance is expected to be better than strictly machine prediction if the expert is signaled for instances when the expected loss is greater than L_\star . Furthermore, if there are regions where the machine’s expected loss is less than L_\star , then optimal synergistic performance will improve upon expert performance alone. The objective is to choose an f and η that minimize Eq. (5.2). We assume L_\star is unknown, and instead propose the following equivalent constrained optimization: firstly, we wish to bound the loss for instances that are not signaled; and secondly, the proportion of human interventions should not exceed a budget ρ . The resulting optimization problem is

$$\begin{aligned} \arg \min_{\eta, f \in \mathcal{F}} \quad & \mathbb{E} \left[L(\hat{h}, Z) \mathbb{I}_{\{f(X) \leq \eta\}} \right] \\ \text{s.t.} \quad & \mathbb{P}[f(X) > \eta] \leq \rho, \end{aligned} \tag{5.3}$$

where \mathcal{F} is a suitable family of functions. For $0 \leq \rho \leq 1$ there is a corresponding L_\star such that Eq. (5.2) and Eq. (5.3) have the same solution. It should be noted that Eq. (5.2) can be seen as the Lagrangian of the optimization problem in Eq. (5.3) without a constant factor $-L_\star\rho$; however, this last term does not include any of the optimization variables so the optimization objective is in fact Eq. (5.2).

5.1 Signaling Function as Loss Interpolation

In the context of binary classification, a principled way to signal instances for human inspection could take into account the notion of margin [11, 3]. Based on this interpretation, $f(X) = -|\hat{h}(X)|$ and $\eta = -\gamma$, where $|\hat{h}(X)|$ is a magnitude understood as a confidence level, and γ is the margin distance. Hence, assuming that we have access to them, the synergistic loss can be expressed as $\ell_\gamma(\hat{h}, h_\star, Z) = L(h_\star, Z) \mathbb{I}_{\{|\hat{h}(X)| < \gamma\}} + L(\hat{h}, Z) \mathbb{I}_{\{|\hat{h}(X)| \geq \gamma\}}$. Instances lying inside the margin region $|\hat{h}(X)| \leq \gamma$ can turn more and more questionable as they get closer to the decision boundary. However, mistaken predictions also occur in regions where the classifier is confident, i.e. $|\hat{h}(X)| > \gamma$, which might have been produced by poorly learned instances that require further inspection even when the predictor is certain about its decision.

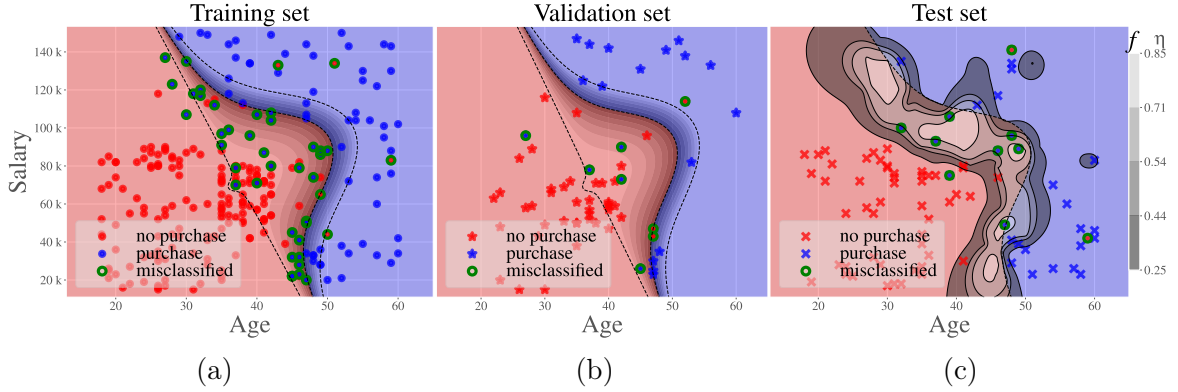


Figure 5.1: Decision surface of the best-performing SVM classifier with polynomial kernel trained on Social Network Ads dataset. (a) Regions of systematic erroneous predictions can be spotted in $\mathbf{Z}_n^{(train)}$, regardless of their position in relation to the margin region (gray-colored gradient area). (b) Misclassifications in $\mathbf{Z}_n^{(val)}$ occur even when the classifier is confident (solid-color area) about its predictions. (c) A signaling function f based on the error set Δ could help to alert the expert when new instances $X \in \mathbf{Z}_n^{(test)}$ lie inside potentially problematic regions.

To pinpoint those regions in the input space \mathcal{X} that could be a source of systematic errors, we can couple the information related to the predictor’s performance with its spatial distribution. This arrangement lays out a broader picture about conflictive regions in the input space that are overlooked by the margin criterion alone. Consider the scenario in Figure 5.1. We assume that the dataset is divided into training, validation and test sets such that $\mathbf{Z}_n = \{\mathbf{Z}_n^{(train)}, \mathbf{Z}_n^{(val)}, \mathbf{Z}_n^{(test)}\}$. The predictor that achieves the best performance for this particular arrangement of the dataset misclassifies instances in regions that can be recognized in both the training $\mathbf{Z}_n^{(train)} = \{\mathbf{X}_n^{(train)}, \mathbf{Y}_n^{(train)}\}$ (Figure 5.1a) and validation $\mathbf{Z}_n^{(val)} = \{\mathbf{X}_n^{(val)}, \mathbf{Y}_n^{(val)}\}$ (Figure 5.1b) sets. For this low-dimensional example, we can define $f(X) = \Psi \in \Delta$, where $\Delta = \{\mathbf{X}_n^{(train)}, \Psi_n^{(train)}\} = \left\{ (X, \Psi) : \Psi = L(\hat{h}, (X, Y)), \forall (X, Y) = Z \in \mathbf{Z}_n^{(train)} \right\}$ is the error set that the signaling function f interpolates, and $L(\hat{h}, (X, Y)) = \mathbb{I}_{\{Y \neq \text{sgn} |\hat{h}(X)|\}}$ is the testing loss function for a binary classification setting. Thus, the main goal of the signaling function f is to *envelope* those zones where systematic errors arise (i.e., clusters where $\Psi > 0$) through a local interpolator $f(X)$ that searches for problematic regions across the input space

uniformly. Then, when a new instance X_* arrives, this function f can be used to alert the expert if $f(X_*) > \eta$ (Figure 5.1c).

While in principle it is always possible to associate raw instances X from the input space \mathcal{X} to their loss distribution Ψ , other arrangements of Δ may be more beneficial for the interpolation. For example, the system’s performance is expected to rely on its representations, then the interpolator could use a representation induced by the deployed learner \hat{h} . Consequently, the signaling function f can alternatively utilize $\Delta = \left\{ \left(\mathbf{X}_n^{(train)}, \hat{h}(\mathbf{X}_n^{(train)}) \right), \Psi_n^{(train)} \right\}$ that also includes the predictions produced by \hat{h} . For classification tasks, these can be the underlying decision values rather than the hard decisions, making the interpolation more sensitive to task-relevant information.

Another consideration regarding the error set is its dimensionality. When dealing with high-dimensional input spaces, using the raw input instances $\mathbf{X}_n^{(train)}$ could harm the interpolation since these spaces tend to be sparsely populated. In this situation, the interpolator can profit from the predictor’s internal representations. For example, if this predictor is a neural network model with P layers, then it can be expressed as a composite function $\hat{h}(X) = (\sigma_P \circ \dots \circ \sigma_1)(X)$, where the functions σ_p represent the computations of the p -th layer with $p \in [P]$ and $[P] = \{1, \dots, P\}$. Given that each layer constitutes some kind of task-relevant feature, the signaling function f can work on the spaces where these task-relevant features exist. In particular, the signaling function can be formulated as $f(X) = s \circ (\sigma_p \circ \dots \circ \sigma_1)(X)$, where $\sigma_p \circ \dots \circ \sigma_1 : \mathcal{X} \rightarrow \Omega$ is a truncated composition that maps the input space to some space Ω and $s : \Omega \rightarrow \mathbb{R}$ is the signaling function that operates on the space Ω .

In general, we propose a framework that can correlate any representation of the input space in the computation of the signaling function f . Specifically, we define $f(X) = s \circ \phi(X) = \Psi$, where $\phi : \mathcal{X} \rightarrow \Omega$ is any mapping from the input space. Under these circumstances, the error set Δ employed to compute f must reflect the associations of the mapped instances with their loss distributions. Hence, $\Delta = \left\{ \phi(\mathbf{X}_n^{(train)}), \Psi_n^{(train)} \right\} = \left\{ (\phi(X), \Psi) : \Psi = L(\hat{h}, (X, Y)), \forall (X, Y) = Z \in \mathbf{Z}_n^{(train)} \right\}$ is the error set for the interpolation, and the testing loss function L has to be decided

by the expert considering the type of the problem at hand.

Finally, the nature of this procedure relies on learning a function f under weak assumptions. Therefore, the family of suitable functions \mathcal{F} that contains f requires no specific form, i.e., it is a nonparametric class of functions on the input space Ω . In particular, we suggest that $f \sim \mathcal{GP}(\mu, \kappa)$, where \mathcal{GP} is Gaussian Process regression characterized entirely by the mean μ and covariance κ functions. This choice is well-suited in this scenario for two reasons: firstly, its ability to approximate arbitrary functions, and secondly, its inherent uncertainty quantification, which yields confidence bounds. This last characteristic plays an important role in our formulation because undersampled regions produce high levels of uncertainty that can be used to alert the expert as well.

Chapter 6

EMPIRICAL ESTIMATION AND TESTING

Working with a \mathcal{GP} involves inferring a distribution over functions given some data, and then using this distribution to make predictions for new instances. Therefore, in this section, we fix the random variables $X = x$, $Y = y$ and $\Psi = \psi$ to denote the observed data. In that sense, $\mathbf{z}_n^{(train)} = \{\mathbf{x}_n^{(train)}, \mathbf{y}_n^{(train)}\} = \{(x_i, y_i), i = 1, \dots, n_{train}\}$ indicates the observed training set used to find and deploy the predictor \hat{h} , where $n_{train} = |\mathbf{z}_n^{(train)}|$ represents the cardinality of the training set; and $\mathbf{\Delta} = \left\{ \phi(\mathbf{x}_n^{(train)}), \boldsymbol{\psi}_n^{(train)} \right\} = \left\{ (\phi(x), \psi) : \psi = L(\hat{h}, (x, y)), \forall (x, y) = z \in \mathbf{z}_n^{(train)} \right\}$ symbolizes the observed error set that we use to compute the posterior predictive distribution $p(f_* | x_*, \mathbf{\Delta})$, where $f_* = s \circ \phi(x_*)$ is the prediction at evaluation point x_* . For a finite sample, this posterior is defined by $p(f_* | x_*, \mathbf{\Delta}) \sim \mathcal{N}(f_* | \mu_*, \Sigma_*)$ which is a joint Gaussian with

$$\mu_* = \mu(\phi(x_*)) + \mathbf{K}_*^T \mathbf{K}^{-1} \left(\boldsymbol{\psi}_n^{(train)} - \mu(\phi(\mathbf{x}_n^{(train)})) \right), \quad (6.1)$$

$$\Sigma_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_*, \quad (6.2)$$

where

$$[\mathbf{K}]_{i,j} = \kappa(\phi(x_i), \phi(x_j)), \quad x_i, x_j \in \mathbf{x}_n^{(train)}, \quad i, j = 1 \dots, n_{train} \quad (6.3)$$

$$[\mathbf{K}_*]_{i,*} = \kappa(\phi(x_i), \phi(x_*)), \quad x_i \in \mathbf{x}_n^{(train)}, \quad i = 1 \dots, n_{train} \quad (6.4)$$

$$\mathbf{K}_{**} = \kappa(\phi(x_*), \phi(x_*)), \quad (6.5)$$

and the covariance function κ is a user's choice. Given these elements, the upper confidence bound (UCB) at x_* can be expressed as

$$f_*^\gamma = \mu_* + \gamma \sqrt{\Sigma_*}, \quad (6.6)$$

where $\gamma > 0$ sets the confidence level and parametrizes multiple signaling functions that can be used to minimize Eq. (5.3).

Lacking knowledge of the true distribution, the validation set $\mathbf{z}_n^{(val)} = \{\mathbf{x}_n^{(val)}, \mathbf{y}_n^{(val)}\} = \{(x_i, y_i), i = 1, \dots, n_{val}\}$ is used as a surrogate optimization for Eq. (5.3), where $n_{val} = |\mathbf{z}_n^{(val)}|$ indicates the cardinality of the validation set. Optimal solution pairs consist of a signaling function f^γ that is above the threshold η for the $B = \lfloor n_{val} \cdot \rho \rfloor$ largest testing losses, where ρ symbolizes the budget required by the user. Let n_{val} define the vectors $\mathbf{u} \in \{0, 1\}^{n_{val}}$, $\mathbf{v} \in \mathbb{R}_{\geq 0}^{n_{val}}$ such that $u_i = \mathbb{I}_{\{f^\gamma(x_i) > \eta\}}$ and $v_i = L(\hat{h}, (x_i, y_i))$ for all $(x_i, y_i) = z_i \in \mathbf{z}_n^{(val)}$. The pair (f^γ, η) optimizes Eq. (5.3) if \mathbf{u} is 1 at exactly B locations that correspond to the B largest losses, such that $\sum_{i=1}^{n_{val}} u_i = B$ and $\tau = \sum_{i=1}^{n_{val}} u_i v_i = \sum_{i=1}^B v_{(i)}$, where $v_{(1)} \geq \dots \geq v_{(n_{val})}$ denote the order statistics of the empirical losses. The optimality proof follows from the rearrangement inequality by Hardy et al. [18]. Algorithm 1 summarizes the entire process.

6.1 Hypothesis testing

Given that our proposed signaling function f^γ relies on the loss distribution, it is likely that those losses are just aleatoric without any meaningful pattern associated with the input space. If that is the case, then the selection based on f^γ makes no difference against a random selection. In that context, the validation set $\mathbf{z}_n^{(val)}$ is also useful for deciding if the signaling function is performing better than random. Let $P = \{x : f^\gamma(\phi(x)) \geq \eta, \forall x \in \mathbf{x}_n^{(val)}\}$ denote the set of signaled instances by f^γ in the validation set, $Q = \{x : L(\hat{h}, (x, y)) > 0, \forall (x, y) = z \in \mathbf{z}_n^{(val)}\}$ define the set of positive losses in the validation set, and $Q_{|P|} \subset Q$ characterizes any subset of Q with cardinality $|P|$. If the null hypothesis $H_0 : P = Q_{|P|}$ can be rejected then our proposed selection mechanism is statistically significant. This evaluation is carried out through a Mann-Whitney U test.

Algorithm 1 Signaling function algorithm

- 1: **Input:** dataset $\mathbf{z}_n^{(train, val)}$, predictor \hat{h} , testing loss L , budget ρ , mean function μ , covariance function κ , confidence-level set γ
 - 2: **Output:** threshold η , signaling function f
 - 3: Compute $\boldsymbol{\psi}_n^{(train)} = L(\hat{h}, \mathbf{z}_n^{(train)})$
 - 4: Compose $\Delta = \left\{ \phi(\mathbf{x}_n^{(train)}), \boldsymbol{\psi}_n^{(train)} \right\}$
 - 5: Fit $f \sim \mathcal{GP}(\mu, \kappa)$ with Δ
 - 6: Evaluate $\mathbf{v} = L(\hat{h}, \mathbf{z}_n^{(val)})$
 - 7: Calculate $n_{val} = \left\lceil \mathbf{z}_n^{(val)} \right\rceil$, $B = \lfloor n_{val} \cdot \rho \rfloor$, $N = |\gamma|$
 - 8: Set $\tau = \infty$
 - 9: **for all** γ in γ **do**
 - 10: Predict UCB at $\mathbf{x}_n^{(val)}$: $f_{val}^\gamma = \mu_{val} + \gamma \sqrt{\Sigma_{val}}$
 - 11: Sort $f_{val(1)}^\gamma \geq \dots \geq f_{val(n_{val})}^\gamma$
 - 12: Set $\eta = f_{val(B)}^\gamma$
 - 13: Evaluate $\mathbf{u} = \mathbb{I}_{\{f_{val}^\gamma > \eta\}}$
 - 14: **if** $\mathbf{u}^T \mathbf{v} < \tau$ **then**
 - 15: $\tau = \mathbf{u}^T \mathbf{v}$
 - 16: $\eta^* = \eta$
 - 17: $\gamma^* = \gamma$
 - 18: **end if**
 - 19: **end for**
 - 20: **RETURN** η^*, f^{γ^*}
-

6.2 Performance Metrics

After solving the optimization problem in (5.3), we evaluate the performance of the best pair (f^γ, η) by comparing two indicators: how much our selection mechanism is able to improve the overall testing loss given a certain budget and how much intervention is required for new instances in comparison with the required budget ρ . The first indicator is quantified by the loss reduction $r_* = \left(\sum_i^{n_*} v_i - \mathbf{u}^T \mathbf{v} \right) / \sum_i^{n_*} v_i$, which is evaluated for validation $\mathbf{z}_n^{(val)}$ and test $\mathbf{z}_n^{(test)}$ sets. The second indicator verifies if the required budget ρ by the user is met for a batch of unseen instances. Therefore, this indicator is calculated as $\hat{\rho} = \left(\sum_i^{n_{test}} u_i \right) / n_{test}$ only for the test set which represents novel instances. It is worth mentioning that for these two metrics we assume that once an instance has been signaled and passed to the expert, his prediction is always correct.

6.3 Baselines for comparison

As described in Section 5.1, a model-aware mechanism can also be used to signal an oracle. In that sense, we contrast our proposed method with either the margin distance $g(x) = |\hat{h}(x)|$ or entropy $g(x) = H[p(y|x)]$ for which we compute a threshold θ following a similar procedure to the one described in Algorithm 1. This model-aware mechanism ($g(x) > \theta$) can be thought of as the model's self-assessment about its predictions. Using our signaling function and a model-aware mechanism, we compare how much greater is the loss reduction and how similar is the selection of instances. In that sense, the loss reductions achieved by $f^\gamma(\phi(x)) > \eta$ and $g(x) > \theta$ are contrasted for each user's budget ρ . Additionally, the Jaccard index $J = |A \cap B| / |A \cup B|$, where $A = \{x_i : f^\gamma(\phi(x_i)) > \eta\}$ and $B = \{x_j : g(x_j) > \theta\}$, is computed in order to quantify the overlap of selected instances by both criteria.

Chapter 7

RESULTS

In this chapter, we show a summary of the results obtained for 9 datasets. Results for classification settings are obtained with the testing loss $L(\hat{h}, (x_i, y_i)) = \mathbb{I}_{\{y_i \neq \hat{h}(x_i)\}}$ and the exponential covariance function $\kappa(x, x') = \exp(-\|x - x'\|/2\sigma^2)$. For regression settings, $L(\hat{h}, (x_i, y_i)) = |\hat{h}(x_i) - y_i|$ and $\kappa(x, x') = \exp(-\|x - x'\|^2/2\sigma^2)$ are utilized. The mean function $\mu(\phi(x)) = 0$ is chosen for both tasks. Additionally, Algorithm 1 is executed, for all tasks, considering multiple budgets $\rho \in \{1\%, 5\%, 10\%, 15\%, 20\%\}$ and confidence set γ composed of 30 confidence-level values $\gamma \in [0, 3]$.

7.1 Datasets and Models

In order to get a comprehensive evaluation of our proposed technique, we explore its performance with a variety of data sets for both classification and regression tasks (Table 7.1). Social Network Ads, Pima Indians Diabetes, Physionet Challenge 2012, German Credit, and IMDB are binary classification tasks. Fashion MNIST and CIFAR10 are multi-class classification tasks. Finally, Red Wine Quality and Boston Housing are single-target output regression tasks.

Given that we want to evaluate how our proposed method performs compared against margin distance and entropy for classification tasks, we require margin-based learners $\hat{h} : \mathcal{X} \rightarrow \mathbb{R}$ and probabilistic learners $\hat{h} : \mathcal{X} \rightarrow [0, 1]$. On the one hand, Support Vector Machine (SVM) and Gaussian Process Classifier (GPC) provide margin distance and probability, respectively, as outputs by definition. In this case, we train SVM with a polynomial kernel and GPC with a Radial Basis Function (RBF) kernel. On the other hand, Neural Networks (NN) models are much more flexible, which allow us to choose different types of activations for the output layer such that the learning procedure

Table 7.1: Datasets characteristics. Description of the number of attributes (without the label attribute), number of instances in each dataset, number of classes for classification tasks, and the models applied to each dataset. Regression tasks present single-target output.

Dataset	# Attributes	# Instances	# Classes	Models
Classification				
Social Network Ads	2	400	2	SVM & GPC
Pima Indians Diabetes [13]	7	768	2	NN
Physionet Challenge 2012 [37]	181	4000	2	NN
German Credit Data [13]	19	1000	2	NN
IMDB [25]	-	25000	2	RNN
Fashion MNIST [43]	28×28	60000	10	CNN
CIFAR 10 [20]	32×32×3	50000	10	CNN
Regression				
Red Wine Quality [13]	12	4898	-	LR, Lasso, SVR
Boston Housing	14	506	-	LR, Lasso, SVR

relies on different measures of success and provides the required outputs. When using NN, the setting to obtain a probabilistic output involves choosing a sigmoid activation evaluated with log loss. Alternatively, to get a margin distance output, we have to set a SVM-type objective evaluated with hinge loss [40]. We can use this same approach to get probability and margin distance for Recurrent Neural Networks (RNN) architectures. It should be noted that tweaking the output layer to get margin distance works well for binary classification tasks, but it does not scale well for multi-class scenarios. Therefore, for multi-class classification problems, we only use probabilistic learners $\hat{h} : \mathcal{X} \rightarrow [0, 1]^m$, where m is the number of classes, with a softmax activation and cross-entropy loss. Table 7.2 describes the NN architectures adopted for the classification tasks.

Regarding regression tasks, the models employed are Linear Regression (LR), Lasso, and Support Vector Regression (SVR) are all single-target output $\hat{h} : \mathcal{X} \rightarrow \mathbb{R}$. We only evaluate, in this setting, the performance of the signaling function f in the validation set $\mathbf{z}_n^{(val)}$ to contrast its performance in the test set $\mathbf{z}_n^{(test)}$.

Table 7.2: NN architectures. Sequential characterization of the components that define the neural network architectures utilized to train and deploy predictors \hat{h} . The symbol $\sigma(\cdot)$ denotes the internal representations and it is positioned at the layer where the activations are taken.

Model	Architecture
NN	$\mathbf{x}_n \xrightarrow{\text{input}} \text{FCL}(128) \xrightarrow{\text{ReLU}} \text{FCL}(64) \xrightarrow{\text{ReLU}} \sigma(\cdot) \rightarrow \text{FCL}(1) \xrightarrow[\text{Linear}]{\text{Sigmoid}}$
RNN	$\mathbf{x}_n \xrightarrow{\text{input}} \text{EL}(5000, 32, 500) \rightarrow \text{LSTM}(100) \xrightarrow{\text{Tanh}} \sigma(\cdot) \rightarrow \text{FCL}(1) \xrightarrow[\text{Linear}]{\text{Sigmoid}}$
CNN	$\mathbf{x}_n \xrightarrow{\text{input}} \text{CL}(32, [3, 3]) \xrightarrow{\text{ReLU}} \text{CL}(64, [3, 3]) \xrightarrow{\text{ReLU}} \text{PL}([2, 2]) \rightarrow \text{Dropout}(0.5) \rightarrow \text{Flatten}() \rightarrow \text{FCL}(128) \xrightarrow{\text{ReLU}} \sigma(\cdot) \rightarrow \text{FCL}(10) \xrightarrow{\text{Softmax}}$
FCL(output dimension) = Fully Connected Layer EL(input dimension, output dimension, input length) = Embedding Layer LSTM(output dimension) = Long Short-Term Memory Layer CL(number of filters, kernel size) = Convolution Layer PL(pool size) = Pooling Layer <i>ReLU, Sigmoid, Linear, Tanh</i> = Activations	

7.2 Training and evaluation procedure

Since the loss distribution guides where the signaling function f operates, it is expected that some losses are just random erroneous predictions that bear no meaningful correlation with the input space. In that case, there is no benefit in computing the signaling function. To test variability of the predictions, we carry out a 5-fold experiment for each dataset. Each fold is divided into training, validation, and test sets so that we can train a predictor \hat{h} , compute the signaling function f and evaluate its performance per fold. For NN models, the training set in each fold is used to minimize the loss with a 128 batch size for 10 epochs and the validation set is utilized for early stopping. For the rest of the models, the training set in each fold is employed to find the best-performing predictor through a 5-fold cross-validation procedure.

Once a predictor \hat{h} is found in each fold, we compute our proposed signaling function f according to Algorithm 1. Then, we test the selection provided by f to ensure that its instance selection is statistically significant, which is indicated through a p-value (See Section 6.1). Since each dataset is a 5-fold experiment, we obtain 5 different

p-values and 5 different signaling function performance reports. If the median of the these p-values is less than 0.05, then we reject the null hypothesis H_0 . If this is the case, the performance results for signaling functions whose folds deliver p-values greater than 0.05 are discarded and the statistics for the performance results are computed with the results of the remaining folds. If the null hypothesis cannot be rejected, then the whole experiment is discarded.

7.3 Classification tasks

The results in this section are a summary that showcase the performance of our proposed signaling function f in relation to a model-aware signaling function that could be either margin distance $g(x) = |\hat{h}(x)|$ or entropy $g(x) = H[p(y|x)]$, depending on the nature of the deployed model. Results for classification tasks are divided into two groups: A and B , which are characterized by the dimensionality of their datasets.

On the one hand, group A considers datasets whose input space dimensionalities can be fed to a \mathcal{GP} directly without running into processing problems. In that sense, we propose three variants of the error set that include raw input instances in the error set. The first variant of the error set is $\Delta^* = \{\mathbf{x}_n^{(train)}, \psi_n^{(train)}\}$, which associates the raw input instances with its corresponding loss distribution. The second variant considers $\Delta^{**} = \left\{ \left(\mathbf{x}_n^{(train)}, \hat{h}(\mathbf{x}_n^{(train)}) \right), \psi_n^{(train)} \right\}$ that correlates the concatenated raw instances and underlying decision values from the predictor \hat{h} with their loss distribution. The third variant is characterized by $\Delta^{***} = \left\{ \mathbf{T} \left(\mathbf{x}_n^{(train)}, \hat{h}(\mathbf{x}_n^{(train)}) \right), \psi_n^{(train)} \right\}$, where \mathbf{T} is an operator that performs a dimensionality reduction technique—for this work, we use PCA (see full results for group A in Appendix A, Tables A.1 to A.6). Figure 7.1 display the results for datasets in group A .

Group B , on the other hand, takes into account datasets whose input space dimensionalities are either too large or impractical (e.g., categorical variables) to be fed directly to a \mathcal{GP} . In consequence, the error sets include the deployed predictor’s internal representations of the input space in $\Delta = \left\{ \phi \left(\mathbf{x}_n^{(train)} \right), \psi_n^{(train)} \right\}$. Therefore, we

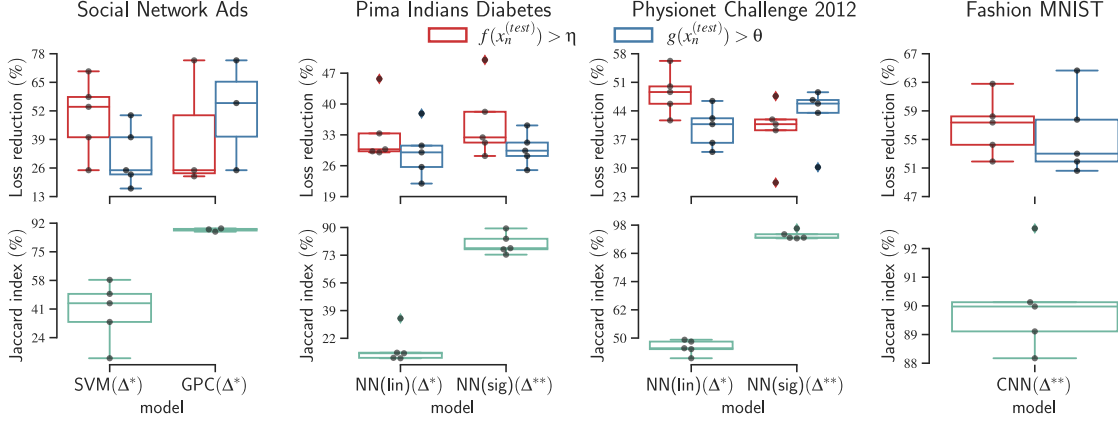


Figure 7.1: Summary statistics for Group A with budget $\rho = 0.15$. Summary statistics for classification tasks whose variants of the error sets consider raw instances from the input space. Each dataset shows the model that is used to deploy a predictor \hat{h} and the variant of the error set that achieves the largest loss reduction in the validation set. These boxplots are derived from a 5-fold experiment for each dataset where the gray bullets and diamonds represent the accepted folds (p-value less than 0.05) in each experiment.

propose the following three error sets: $\Delta^* = \left\{ (\sigma_p \circ \dots \circ \sigma_1) (\mathbf{x}_n^{(train)}), \boldsymbol{\psi}_n^{(train)} \right\}$, which pairs task-relevant features induced by the deployed predictor with its loss distribution; $\Delta^{**} = \left\{ \left((\sigma_p \circ \dots \circ \sigma_1) (\mathbf{x}_n^{(train)}), \hat{h} (\mathbf{x}_n^{(train)}) \right), \boldsymbol{\psi}_n^{(train)} \right\}$, which correlates induced task-relevant features and underlying decision values with their loss distribution; and, finally, $\Delta^{***} = \left\{ \mathbf{T} \left((\sigma_p \circ \dots \circ \sigma_1) (\mathbf{x}_n^{(train)}), \hat{h} (\mathbf{x}_n^{(train)}) \right), \boldsymbol{\psi}_n^{(train)} \right\}$, which associates a transformation \mathbf{T} of the induced task-relevant features and underlying decision values with their loss distribution; in this case \mathbf{T} is also PCA (see full results for group A in Appendix A, Tables A.7 to A.12). Figure 7.2 show the results for datasets in group B .

The results reported in Figure 7.1 and Figure 7.2 correspond to the loss reduction attained for both our proposed signaling function $f(x)$ (red) and the loss reduction achieved by the model-aware signaling function $g(x)$ (blue) in the test set for a user-defined budget $\rho = 0.15$. The variant of the error set is chosen considering the one that achieves the largest loss reduction in the validation set with our signaling function f as reference. Notably, in both groups A and B , the variant that gets chosen most of the time is Δ^{**} . Also, this arrangement of the error set outperforms

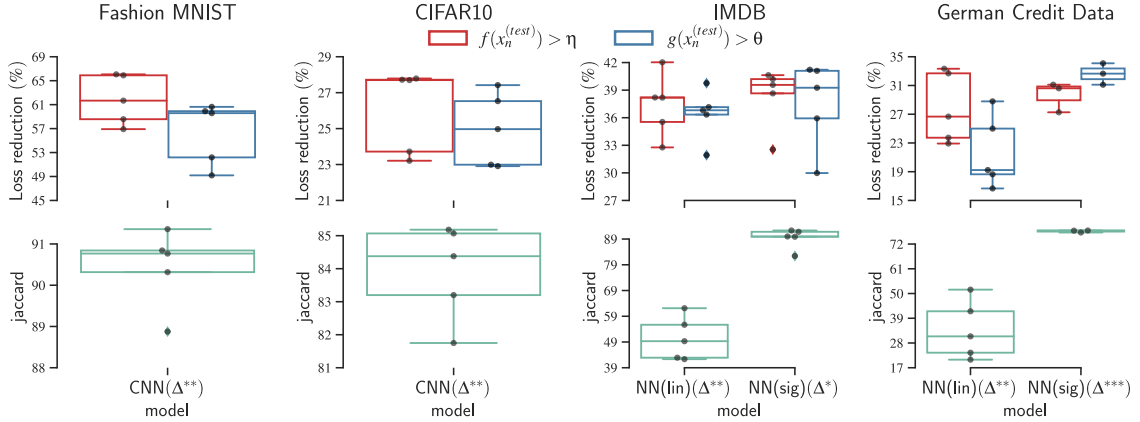


Figure 7.2: Summary statistics for Group B with budget $\rho = 0.15$. Summary statistics for classification tasks whose variants of the error sets rely on internal representations induced from the input space by the deployed predictor. Each dataset shows the model that is used to deploy a predictor \hat{h} and the variant of the error set that achieves the largest loss reduction in the validation set. These boxplots are derived from a 5-fold experiment for each dataset where the gray where the gray bullets and diamonds represent the accepted folds (p-value less than 0.05) in each experiment.

model-aware signaling functions g evaluated in the test set in most cases. This means that the combination of either the raw input instances with their corresponding underlying decision values $\phi(\mathbf{x}_n^{(train)}) = (\mathbf{x}_n^{(train)}, \hat{h}(\mathbf{x}_n^{(train)}))$ or the induced internal representations of the input instances along with their corresponding underlying decision values $\phi(\mathbf{x}_n^{(train)}) = ((\sigma_p \circ \dots \circ \sigma_1)(\mathbf{x}_n^{(train)}), \hat{h}(\mathbf{x}_n^{(train)}))$ provides the most informative feature description for the signaling function f to identify regions where systematic erroneous predictions occur.

The boxplots in Figure 7.1 and Figure 7.2 also elicit some interesting facts. In one respect, the difference between the Jaccard indices computed for margin-based and probabilistic predictors is striking. According to these results, our proposed signaling function f picks instances that are more similar to those selected by the entropy-based measure than those selected by the margin-based predictor. While this similarity index reflects that our signaling function pinpoints instances in a similar way as an entropy-based principle, the selection is not exactly the same. This difference suggests that f not only explores regions with high entropy, but also regions with low entropy, where the

deployed predictor \hat{h} is confident about its decisions. In another respect, we must point out that each gray circle in the boxplots in Figure 7.1 and Figure 7.2 represent a fold where the signaling function f is statistically significant. Given that each experiment is 5-fold, we expect to obtain at most 5 circles per boxplot. In most experiments, we get 5 accepted folds, which demonstrates that our proposed signaling function f chooses instances that is different from a random selection.

Finally, Figure 7.3 compares what instances are selected in the test set by our signaling function f and by entropy as model-aware signaling function g for both Fashion MNIST and CIFAR10 datasets. Figure 7.3a shows that, for the Fashion MNIST dataset, f picks instances for which the deployed predictor \hat{h} is highly confident about its decisions. The entropy level for these selected instances is low, yet some of them are mistakenly classified. Therefore, high entropy is not able to identify the regions where systematic erroneous predictions take place. Figure 7.3b displays that, for the CIFAR10 dataset, the selection of the top instances is more correlated with the entropy level. These entropy quantities in the predictions for CIFAR10 are expected to be higher than the entropies for Fashion MNIST because it is a much more complex problem. However, we must highlight that the prioritization of instances is not the same. For both datasets, we can observe that the selection that f performs is focused around regions of misclassified instances, while the choice that g executes is driven by the *proximity* to the deployed predictor’s decision surface.

7.4 Regression tasks

The results in this section are a summary of regression tasks that illustrate the performance of our proposed signaling function f evaluated in the test set. Similar to classification tasks, we consider three variants of the error set to compute f . The first variant of the error set is $\Delta^* = \left\{ \mathbf{x}_n^{(train)}, \psi_n^{(train)} \right\}$, which associates the raw input instances with its corresponding loss distribution. The second variant considers $\Delta^{**} = \left\{ \left(\mathbf{x}_n^{(train)}, \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \psi_n^{(train)} \right\}$ that correlates the concatenated raw instances and their corresponding predictions from \hat{h} with their loss distribution. The third variant

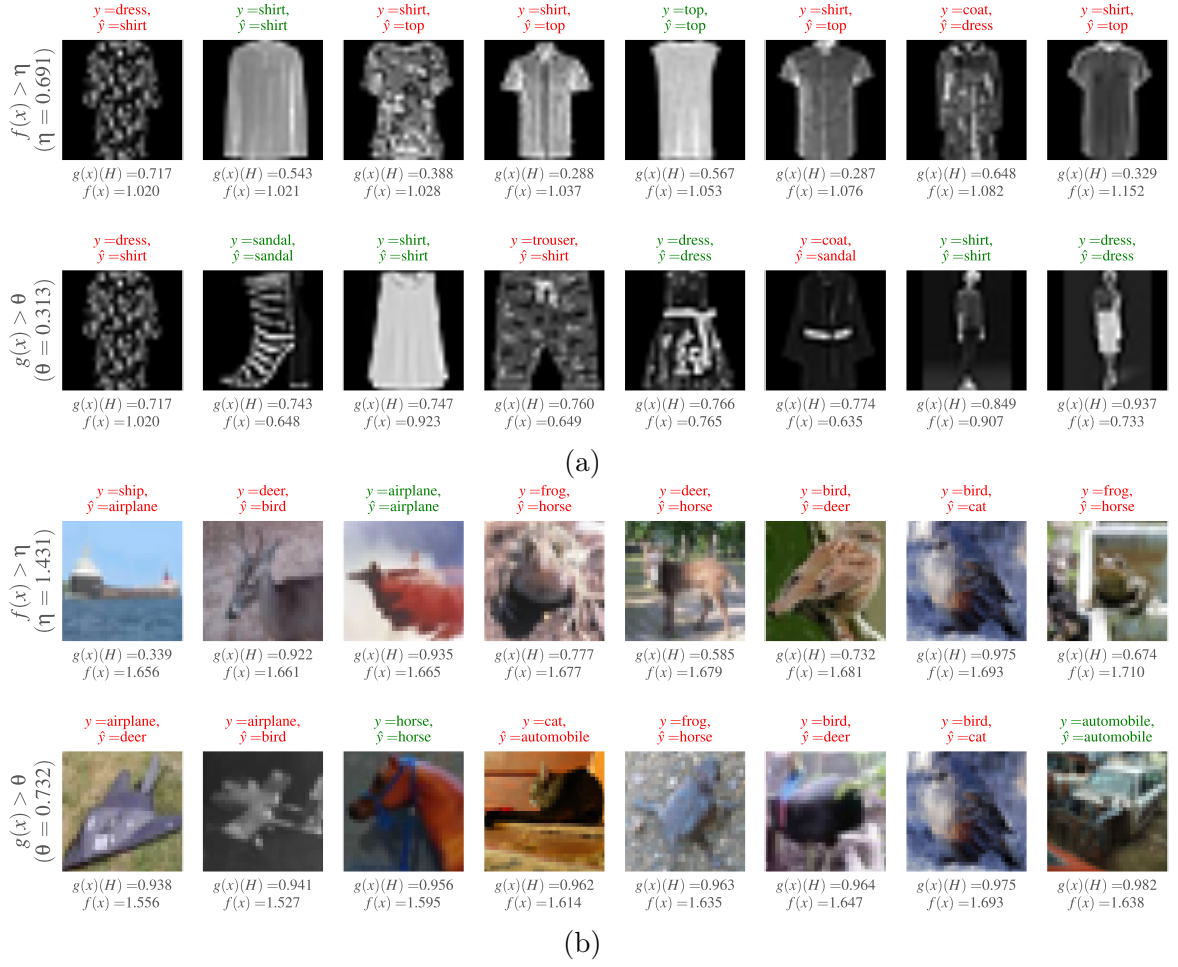


Figure 7.3: Comparison of the top 8 selected instances by our proposed signaling function f and by entropy as model-aware signaling function g . Top 8 selected instances in Fashion MNIST dataset (a) and in CIFAR10 (b) for each signaling function where the samples that are picked first by f (images in the upper row of each group) and g (images in the lower row of each group) are organized in descending order from right to left. Some instances picked by f present low entropy, which indicates a confident prediction \hat{y} by the deployed predictor that is erroneous in some cases.

is characterized by $\Delta^{***} = \left\{ \mathbf{T} \left(\mathbf{x}_n^{(train)}, \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \boldsymbol{\psi}_n^{(train)} \right\}$, where \mathbf{T} is an operator that performs a dimensionality reduction technique—for this work, we use PCA (see full results in Appendix A, Tables A.13 to A.15). Figure 7.4 display the results for regression tasks.

The boxplots in Figure 7.4 evoke a couple of interpretations. First, the fact that the loss reduction in the test set is similar across models for a common dataset

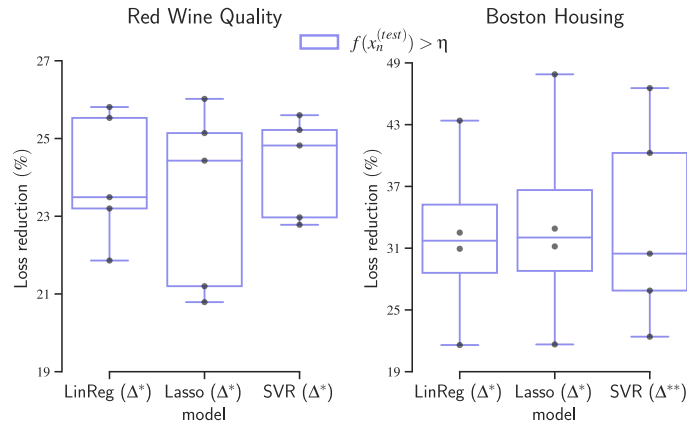


Figure 7.4: Summary statistics for regression tasks with budget $\rho = 0.15$. Summary statistics for regression tasks whose variants of the error sets rely on raw instances from the input space. Each dataset shows the model that is used to deploy a predictor \hat{h} and the variant of the error set that achieves the largest loss reduction in the validation set. These boxplots are derived from a 5-fold experiment for each dataset where the gray bullets represent the accepted folds (p-value less than 0.05) in each experiment.

indicates that probably the configuration chosen to compute f does not capture the difference among the different model predictions. Second, for a small user-defined budget, obtaining a large loss reduction could be a sign of an underperforming predictor that might need to be reconsidered by the original modeler.

Chapter 8

DISCUSSION

The key contribution of our work is a methodology to prevent the propagation of a model’s training errors into deployment through the intelligent use of a human in the loop. While distribution shift, outliers and label noise may also produce systematic errors, we assume that the probability distribution remains the same during training, testing and deployment, and the problem resides in inadequacies of the model rather than with the data distribution. Certain cases, such as model underfitting due to lack of instances, can be cast in terms of distribution shift if similar instances are prevalent in the test case. Under some assumptions, such shifts can be identified and solved by importance-weighting empirical risk minimization during the training stage, without considering the model’s training set performance. Methods for outlier detection and methods for handling label noise aim to improve the quality of the data in order to improve the performance of the predictor in deployment. In terms of the underlying mechanics, these methods employ similar interpolation schemes to understand the representativeness of data instances (both features and target variables). Commonly, outlier detection techniques use clustering algorithms to assess what the conditional probability $p(y|x)$ is in order to improve training data labels with expert’s advice. Notably this does not involve a discrepancy in the model. Another difference with our work is that in our case an expert’s attention is conditioned on a user-defined threshold, which is translated as a constrained level of human intervention.

In this work, we consider a human-in-the-loop scenario during machine learning deployment. The human user, who is expected to be a domain expert, should be signaled to manually examine an instance when there is undue risk in trusting the machine prediction. Learning when to signal the user is itself a machine learning

problem, with the goal of the user boosting the joint decision making process, while still benefiting from the automated decision making a majority of the time. This is similar to online active learning, but the goal is to improve the test set performance through the combination of the human and machine decisions rather than updating the model. By deciding when to query the user, our proposed framework bears similarity to active learning. However, it differs in motivation since the machine model remains fixed and the level of human intervention can be adjusted. Furthermore, some active learning procedures require knowledge of the model’s uncertainty [22], either in terms of the conflicting decisions of a committee [16, 34] or distance to margin [8, 41] to decide on what to query while our approach relies on the loss distribution to guide the queries to the expert. Finally, with our approach, we seek regions of systematic erroneous predictions that the deployed model is not able to recognize by its own means.

The results in Section 7 provide some useful insights. A signaling function based on the association of the input space, or a representation of the input space, designed to find regions where the predictor is performing poorly, often outperforms a model-aware criterion such as entropy. This criterion derived from the deployed predictor can be thought of as a *self-assessment* that could easily ignore regions distant from to the decision surface. Our signaling function f not only identifies erroneous predictions in disputable regions, but also in zones where the predictor is fairly confident about its decisions. Figures 7.1 and 7.2 show that f selects instances similar to an entropy-based criterion, but not identical. Figures 7.3a and 7.3b demonstrates that this selection prioritizes instances in a different way, where not only uncertain (high entropy) predictions, but also confident (low entropy) predictions are singled out based on the regions that present inconsistent decisions with the underlying true distribution. Thanks to the spatially-driven nature of our proposed signaling function f , we can direct our attention to those problematic spaces and study how relevant the features of a signaled instance are in relation to the problem. Furthermore, the selection that f provides can help understand why some samples do not contribute towards a more accurate decision surface. Finally, the results suggest that the best setting to compute f is the combination

of task-relevant features (induced representation from the deployed predictor) and underlying decision values with their loss distribution. This setting achieves a higher loss reduction in the Fashion MNIST dataset, which is tested considering both raw input instances (Figure 7.1) and induced representations (Figure 7.2) from the input space.

8.1 Limitations

A limitation of the current study is related to data sets with categorical features. In order to deal with them, we chose to analyze induced representations from the deployed predictor to feed the \mathcal{GP} . However, our current setting does not process raw categorical variables adequately. The choice of a \mathcal{GP} for interpolation may be ill-suited to categorical or count data. The chosen kernel functions for classification and regression tasks are unable to capture abrupt changes produced by the presence of categorical variables. Further analysis is required to determine an appropriate kernel capable to handle those conditions.

Another limitation could derive from the use of signaled and corrected instances. In deployment, all observations contribute to an unbiased sample of $\mathbb{P}[f(x) > \eta]$. These can be used to update the threshold in order to stay close to the budget. However, the new observations of the loss are only observed at times the user is signaled—a biased sample. These loss observations can be used to improve the interpolation, but not the empirical risk. New losses will only be observed in areas that the machine is predicted to perform poorly. Thus, the interpolation (and the signal function) should be updated when there is evidence that the interpolation is underestimating the machine’s ability. Unlike online learning, and similar to correctable learning [29], our proposal requires us to store additional examples (in addition to the original training and validation set). To update the threshold we only need to maintain the signaling function evaluation. However, to update the interpolation we need to store all examples (since even examples that are not signaled need to be stored to set an appropriate threshold).

8.2 Future work

In deployment, the signaling function is expected to flag instances for user inspection. This will create more training data, and along with the original training data, a new model or models can be trained to boost the original model in cases for which the latter performs poorly. These new models function as *patches*. Given sufficient evidence that the patches' predictions are within an acceptable level of risk, these patches can be applied to similar instances automatically. In fact, the signaling function may identify cases that are correctable with a patch trained purely on existing data.

Since our approach aims at finding regions of systematic erroneous predictions, we could argue that a different hypothesis set can be used for each region to *patch* them. Specifically, we assume that the input space \mathcal{X} can be divided into p disjoint partitions: $\mathcal{X} = \bigcup_{k=1}^p \mathcal{X}_k$ with $\mathcal{X}_k \cap \mathcal{X}_{k'} = \emptyset$ for $k \neq k'$. For instance, this partitioning can be guided by the threshold η of the signaling function f . We then denote by \mathcal{H}_k the hypothesis set used for region \mathcal{X}_k and by \mathcal{L}_k the expected loss of h on region \mathcal{X}_k : $\mathcal{L}_k(h) = \mathbb{E}_{(X,Y) \sim \mathcal{D}} \left[\ell(h, (X, Y)) \mid X \in \mathcal{X}_k \right]$ for each $k \in [m] = \{1, \dots, m\}$. Hence, for any hypothesis h , we have the generalization error defined by $\mathcal{L}(h) = \sum_{k=1}^m q_k \mathcal{L}_k(h)$, where q_k can be computed based on an importance-weighted mechanism. Considering this framework, many challenges arise: How to define the hypothesis sets for each region? How to best combine predictions from the deployed predictor with the predictions from the patches? Can the original partitioning and the signaling function be modified adaptively? Can multiple oracles be combined? Answers to these questions require further investigation.

Chapter 9

CONCLUSION

Deploying machine learning models can accelerate and improve decision making in various areas. Yet, trained models are unlikely to have performed perfectly in training and validation sets. These errors can be attributed to multiple causes including intrinsic variability, but localized regions or repeated cases of relatively high loss could indicate systematic problems with the model. We argue that the information on the errors should be used during deployment. Specifically, novel test cases that are somehow similar to previous errors should be flagged for user inspection. We formalize this by defining a signaling function that interpolates the distribution of empirical loss and alerts the user when this function exceeds a threshold. We demonstrate the effectiveness of Gaussian process regression for this task on both synthetic and real-world data sets and show reduced loss for synergistic performance (assuming an expert user who is perfect when queried).

BIBLIOGRAPHY

- [1] Charu C. Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and Philip S. Yu. “Active learning: A survey”. In: *Data Classification: Algorithms and Applications* (2014), pp. 571–606.
- [2] Bibb Allen Jr., Steven E Seltzer, Curtis P Langlotz, Keith P Dreyer, Ronald M Summers, Nicholas Petrick, Danica Marinac-Dabic, Marisa Cruz, Tarik K Alkasab, Robert J Hanisch, et al. “A Road Map for Translational Research on Artificial Intelligence in Medical Imaging: From the 2018 National Institutes of Health/RSNA/ACR/The Academy Workshop”. In: *Journal of the American College of Radiology* (2019).
- [3] Peter L. Bartlett and Marten H. Wegkamp. “Classification with a Reject Option Using a Hinge Loss”. In: *Journal of Machine Learning Research* 9 (2008), pp. 1823–1840.
- [4] Andrew H Beck, Ankur R Sangoi, Samuel Leung, Robert J Marinelli, Torsten O Nielsen, Marc J Van De Vijver, Robert B West, Matt Van De Rijn, and Daphne Koller. “Systematic analysis of breast cancer morphology uncovers stromal features associated with survival”. In: *Science Translational Medicine* 3 (2011).
- [5] Mikhail Belkin, Daniel J. Hsu, and Partha Mitra. “Overfitting or perfect fitting? Risk bounds for classification and regression rules that interpolate”. In: *NeurIPS* (2018).
- [6] Gérard Biau, Luc Devroye, and Gabor Lugosi. “Consistency of Random Forests and Other Averaging Classifiers”. In: *Journal of Machine Learning Research* 9 (2008), pp. 2015–2033.

- [7] Markus Breunig, Hans-Peter Kriegel, Raymond Ng, and Joerg Sander. “LOF: Identifying Density-Based Local Outliers”. In: *ACM Sigmod Record* 29 (2000), pp. 93–104. DOI: [10.1145/342009.335388](https://doi.org/10.1145/342009.335388).
- [8] Colin Campbell, Nello Cristianini, and Alex J Smola. “Query Learning with Large Margin Classifiers”. In: *Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc. 2000, pp. 111–118.
- [9] Raghavendra Chalapathy and Sanjay Chawla. *Deep Learning for Anomaly Detection: A Survey*. 2019. arXiv: [cs.LG/1901.03407v2](https://arxiv.org/abs/cs.LG/1901.03407v2) [[cs.LG](https://arxiv.org/abs/cs.LG/1901.03407v2)].
- [10] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly Detection: A Survey”. In: *ACM Computing Surveys* 41 (2009). DOI: [10.1145/1541880.1541882](https://doi.org/10.1145/1541880.1541882).
- [11] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. “Learning with Rejection”. In: *Algorithmic Learning Theory*. Ed. by Ronald Ortner, Hans Ulrich Simon, and Sandra Zilles. 2016, pp. 67–82.
- [12] Trafton Drew, Melissa L-H Võ, and Jeremy M Wolfe. “The invisible gorilla strikes again: sustained inattention blindness in expert observers”. In: *Psychological Science* 24 (2013), pp. 1848–53.
- [13] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: <http://archive.ics.uci.edu/ml>.
- [14] Joann G. Elmore and Suzanne W. Fletcher. “The Risk of Cancer Risk Prediction: “What Is My Risk of Getting Breast Cancer?””. In: *JNCI: Journal of the National Cancer Institute* 98 (2006), pp. 1673–1675. DOI: [10.1093/jnci/djj501](https://doi.org/10.1093/jnci/djj501).
- [15] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. “Dermatologist-level classification of skin cancer with deep neural networks”. In: *Nature* 542 (2017), pp. 115–118.

- [16] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. “Selective sampling using the query by committee algorithm”. In: *Machine Learning* 28 (1997), pp. 133–168.
- [17] Arthur Gretton, Alexander Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. “Covariate Shift by Kernel Mean Matching”. In: *Dataset Shift in Machine Learning*. Vol. 3. 2008, pp. 131–160. DOI: [10.7551/mitpress/9780262170055.003.0008](https://doi.org/10.7551/mitpress/9780262170055.003.0008).
- [18] G. H. Hardy, Littlewood John Edensor, and Pólya George. *Inequalities*. Vol. 1. Cambridge University Press, 1934.
- [19] Danielle Leah Kehl and Samuel Ari Kessler. *Algorithms in the Criminal Justice System: Assessing the Use of Risk Assessments in Sentencing*. 2017.
- [20] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. *CIFAR-10 (Canadian Institute for Advanced Research)*. URL: <http://www.cs.toronto.edu/~kriz/cifar.html>.
- [21] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. “Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles”. In: *NeurIPS* (2016).
- [22] David D Lewis and William A Gale. “A sequential algorithm for training text classifiers”. In: *ACM SIGIR Conference on Research and Development in Information Retrieval*. 1994, pp. 3–12.
- [23] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. “Detecting and Correcting for Label Shift with Black Box Predictors”. In: *ICML* (2018).
- [24] F. T. Liu, K. M. Ting, and Z. Zhou. “Isolation Forest”. In: *IEEE International Conference on Data Mining* (2008).
- [25] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. “Learning Word Vectors for Sentiment Analysis”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011, pp. 142–150.

- [26] Nicholas Petrick, Berkman Sahiner, Samuel G Armato III, Alberto Bert, Loredana Correale, Silvia Delsanto, Matthew T Freedman, David Fryd, David Gur, Lubomir Hadjiiski, et al. “Evaluation of computer-aided detection and diagnosis systems”. In: *Medical Physics* 40 (2013).
- [27] Xin Qiu, Elliot Meyerson, and Risto Miikkulainen. *Quantifying Point-Prediction Uncertainty in Neural Networks via Residual Estimation with an I/O Kernel*. 2019. arXiv: [cs.LG/1906.00588v5](https://arxiv.org/abs/cs.LG/1906.00588v5) [[cs.LG](https://arxiv.org/abs/cs.LG/1906.00588v5)].
- [28] Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. “Failing Loudly: An Empirical Study of Methods for Detecting Dataset Shift”. In: *NeurIPS* (2019).
- [29] Karthik Raman, Krysta M Svore, Ran Gilad-Bachrach, and Chris JC Burges. “Learning from mistakes: towards a correctable learning algorithm”. In: *ACM International Conference on Information and Knowledge Management* (2012).
- [30] Maria Eugenia Ramirez-Loaiza, Manali Sharma, Geet Kumar, and Mustafa Bilgic. “Active learning: an empirical study of common baselines”. In: *Data Mining and Knowledge Discovery* 31 (2016).
- [31] Marco Saerens, Patrice Latinne, and Christine Decaestecker. “Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure”. In: *Neural Computation* 14 (2002). DOI: [10.1162/089976602753284446](https://doi.org/10.1162/089976602753284446).
- [32] Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. “Support Vector Method for Novelty Detection”. In: *NIPS* (1999).
- [33] Burr Settles. “Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning”. In: *Active Learning*. 2012.
- [34] H. S. Seung, M. Opper, and H. Sompolinsky. “Query by Committee”. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. 1992, pp. 287–294. DOI: [10.1145/130385.130417](https://doi.org/10.1145/130385.130417).

- [35] Alireza Shafaei, Mark Schmidt, and James J. Little. *Does Your Model Know the Digit 6 Is Not a Cat? A Less Biased Evaluation of “Outlier” Detectors*. 2018. arXiv: [cs.LG/1809.04729v2](https://arxiv.org/abs/cs.LG/1809.04729v2) [[cs.LG](#)].
- [36] Hidetoshi Shimodaira. “Improving predictive inference under covariate shift by weighting the log-likelihood function”. In: *Journal of Statistical Planning and Inference* 90 (2000), pp. 227–244. DOI: [10.1016/S0378-3758\(00\)00115-4](https://doi.org/10.1016/S0378-3758(00)00115-4).
- [37] Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. “Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012”. In: *Computing in Cardiology* 39 (2012), p. 245.
- [38] Amos Storkey. “When Training and Test Sets Are Different: Characterizing Learning Transfer”. In: *Dataset Shift in Machine Learning* (2009), pp. 3–28. DOI: [10.7551/mitpress/9780262170055.003.0001](https://doi.org/10.7551/mitpress/9780262170055.003.0001).
- [39] M. Sugiyama, Shinichi Nakajima, H. Kashima, P.V. Buenau, and Motoaki Kawanabe. “Direct importance estimation with model selection and its application to covariate shift adaptation”. In: *Proceeding of the 21st Annual Conference on Neural Information Processing Systems* (2007), pp. 1433–1440.
- [40] Yichuan Tang. “Deep learning using linear support vector machines”. In: *ICML* (2013).
- [41] Simon Tong and Daphne Koller. “Support vector machine active learning with applications to text classification”. In: *Journal of Machine Learning Research* 2 (2001), pp. 45–66.
- [42] Abraham J. Wyner, Matthew Olson, Justin Bleich, and David Mease. “Explaining the Success of Adaboost and Random Forests as Interpolating Classifiers”. In: *Journal of Machine Learning Research* 18 (2017), pp. 1558–1590.
- [43] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. Aug. 28, 2017. arXiv: [cs.LG/1708.07747](https://arxiv.org/abs/cs.LG/1708.07747) [[cs.LG](#)].

- [44] K. Zhang, B. Schölkopf, Krikamol Muandet, and Z. Wang. “Domain adaptation under target and conditional shift”. In: *ICML (2013)*.

Appendix A

FULL RESULTS

This appendix shows the full numeric results for all experiments that are carried out in this work. These outcomes are meant to quantify the performance of our proposed signaling function f , first, in relation to its performance in the validation set $\mathbf{z}_n^{(val)}$, and, second, in relation to a model-aware signaling function that could be either margin distance $g(x) = |\hat{h}(x)|$ or entropy $g(x) = H[p(y|x)]$, depending on the nature of the learner. In order to understand under what circumstances our signaling function f is most effective, we train it using multiple variants of $\Delta = \{\phi(\mathbf{x}_n^{(train)}), \psi_n^{(train)}\}$. The first variant considers $\phi(\mathbf{x}_n^{(train)}) = \mathbf{x}_n^{(train)}$, which is the most trivial setting and only accounts for the raw inputs. The second variant contemplates $\phi(\mathbf{x}_n^{(train)}) = \left(\mathbf{x}_n^{(train)}, \hat{h}(\mathbf{x}_n^{(train)})\right)$ that concatenates the raw input instances to the raw predictions. The third variant regards $\phi(\mathbf{x}_n^{(train)}) = \mathbf{T}\left(\mathbf{x}_n^{(train)}, \hat{h}(\mathbf{x}_n^{(train)})\right)$, where \mathbf{T} is an operator that performs a dimensionality reduction technique—for this work, we used PCA. These three variants are applied to low and mid-dimensional data sets involved in classification and regressions tasks. For high-dimensional and mixed-type data sets, we also use those three variants described previously, but with a slight modification. Instead of using the raw input instances $\mathbf{x}_n^{(train)}$, we use a representation $(\sigma_p \circ \dots \circ \sigma_1)(\mathbf{x}_n^{(train)})$ that is induced by the deployed model.

A.1 Classification tasks

Classification experiments are arranged in Tables [A.1](#) to [A.12](#). All experiments in this group follow this pattern: odd number tables ([A.1](#), [A.3](#), [A.5](#), [A.7](#), [A.9](#), [A.11](#)) display the performance metrics of our proposed signaling function f evaluated for validation and test sets, as described in Section [6.2](#), and even number tables

(A.2, A.4, A.6, A.8, A.10, A.12) exhibit the corresponding baseline comparisons of f against g that could be either margin distance or entropy, as detailed in Section 6.3. The performance metrics and baseline comparison are paired by a common signaling function f ; for instance, Table A.1 and A.2 expose results for the same f trained on $\Delta = \left\{ \mathbf{x}_n^{(train)}, \boldsymbol{\psi}_n^{(train)} \right\}$.

In one respect, the performance metrics in Tables A.1, A.3, A.5, A.7, A.9, A.11 display the synergistic performance in validation set r_{val} and test set r_{test} assuming perfect predictions from the human. Additionally, they show how often the human assistance is required in the test set $\hat{\rho}$ to contrast if our solution meets the user-defined budget ρ . In another respect, the loss reduction comparison in Tables A.2, A.4, A.6, A.8, A.10, A.12 showcases the loss that could be corrected in test set evaluated for signaling functions $f(x)$ and $g(x)$. Moreover, they exhibit the Jaccard index J that quantifies how similar is the selection between those functions. The results reported in these tables correspond to cases where the evaluation loss for signaled instances is statistically significant (reject H_0).

Finally, classification tasks are divided into groups A and B . Group A is characterized by datasets whose input spaces that can be fed directly to a \mathcal{GP} , without running into computation problems, and consist of Tables A.1 to A.6. Group B is characterized by datasets whose input spaces are too large or impractical to be fed directly to a \mathcal{GP} , and consist of Tables A.7 to A.12

A.2 Regression tasks

Regression experiments are arranged in Tables A.13 to A.15. Similar to the performance metrics for classification tasks, these tables unveil the synergistic performance in validation set r_{val} and test set r_{test} assuming perfect predictions from the human. Also, they display how often the human assistance is required in the test set $\hat{\rho}$ to compare if our proposed methodology meets the user-defined budget ρ . The results reported correspond to cases where the evaluation loss for signaled instances is statistically significant (reject H_0).

Table A.1: Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \{\mathbf{x}_n^{(train)}, \boldsymbol{\psi}_n^{(train)}\}$ for classification tasks. The loss reduction symbolized by $r_{val}(\%)$ and $r_{test}(\%)$ are the percentages of the loss that could be corrected in validation and test set, respectively. The signal rate $\hat{\rho}$ shows how often the human assistance is required in the test set. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$r_{val}(\%)$	$\hat{\rho}$	$r_{test}(\%)$	H_0
Social Network Ads	SVM	0.05	20.0(20.0-25.0)	0.02(0.02-0.06)	20.0(12.5-23.1)	✓
		0.10	33.3(33.3-41.7)	0.08(0.08-0.12)	50.0(30.0-53.9)	✓
		0.15	50.0(50.0-53.3)	0.14(0.09-0.14)	53.9(40.0-58.3)	✓
		0.20	58.3(55.6-60.0)	0.16(0.10-0.20)	58.3(50.0-70.0)	✓
	GPC	0.05	27.5(25.0-30.8)	0.03(0.02-0.05)	11.1(0.0-26.0)	✓
		0.10	44.4(34.7-52.2)	0.08(0.07-0.09)	22.2(11.1-23.6)	✓
		0.15	55.6(46.5-57.8)	0.18(0.13-0.20)	25.0(23.6-50.0)	✓
		0.20	60.0(50.8-63.3)	0.18(0.16-0.18)	33.3(29.2-33.3)	✓
Pima Indians Diabetes	NN	0.01				×
		(linear)	0.05	12.5(11.6-14.2)	0.06(0.06-0.07)	11.4(10.6-13.8)
	(linear)	0.10	24.9(21.6-28.1)	0.10(0.09-0.11)	21.8(19.8-23.4)	✓
		0.15	34.5(28.6-36.0)	0.14(0.14-0.15)	29.7(29.3-33.3)	✓
		0.20	42.9(42.9-44.8)	0.19(0.19-0.22)	43.2(36.6-44.4)	✓
		NN (sigmoid)	0.01			
	0.05					×
	0.10		22.5(21.2-24.6)	0.11(0.10-0.12)	24.7(23.4-26.1)	✓
	0.15		31.2(29.0-38.5)	0.14(0.13-0.14)	29.4(28.2-29.7)	✓
	Physionet Challenge 2012	NN (linear)	0.01	4.4(4.2-4.7)	0.01(0.01-0.01)	4.1(3.2-5.0)
0.05			23.4(20.2-24.5)	0.05(0.05-0.05)	18.4(18.0-21.4)	✓
0.10			40.3(36.0-40.4)	0.09(0.09-0.10)	35.2(34.4-37.6)	✓
0.15			54.0(52.8-54.3)	0.15(0.14-0.15)	48.6(45.7-50.0)	✓
0.20			62.9(61.7-64.5)	0.20(0.18-0.20)	56.4(55.5-59.6)	✓
NN (sigmoid)		0.01	4.2(3.9-4.5)	0.01(0.01-0.01)	3.7(1.9-4.7)	✓
		0.05	18.8(16.9-20.2)	0.05(0.04-0.05)	13.9(12.4-14.0)	✓
		0.10	28.1(27.3-30.3)	0.10(0.10-0.10)	26.9(25.2-29.5)	✓
		0.15	42.7(40.0-43.8)	0.15(0.14-0.15)	41.7(35.5-42.9)	✓
		0.20	53.9(53.1-55.7)	0.20(0.18-0.20)	48.1(45.3-52.4)	✓
Fashion MNIST	CNN	0.01	4.4(3.8-4.7)	0.01(0.01-0.01)	4.1(3.2-4.8)	✓
		0.05	16.8(16.5-16.9)	0.05(0.05-0.05)	13.8(12.0-16.2)	✓
		0.10	30.9(30.5-31.6)	0.10(0.10-0.10)	28.2(24.8-28.4)	✓
		0.15	43.4(42.3-43.5)	0.15(0.15-0.16)	40.3(38.8-42.2)	✓
		0.20	52.9(52.3-54.3)	0.20(0.19-0.21)	50.7(48.6-53.3)	✓

Table A.2: Loss reduction comparison in test set with $f(x)$ trained on $\Delta = \{\mathbf{x}_n^{(train)}, \boldsymbol{\psi}_n^{(train)}\}$ compared against $g(x)$. The loss reduction symbolized by $r_{test}(\%)$ is the percentage of the loss that could be corrected in test set which is evaluated for $f(x)$ and $g(x)$. The Jaccard index J shows how similar the instance selection is for both functions. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$f(x) > \eta$	$g(x) > \theta$	J	Reject H_0
			$r_{test}(\%)$	$r_{test}(\%)$		
Social Network Ads	SVM	0.05	20.0(12.5-23.1)	10.0(8.3-10.0)	0.07(0.00-0.17)	✓
		0.10	50.0(30.0-53.9)	20.0(12.5-23.1)	0.18(0.06-0.45)	✓
		0.15	53.9(40.0-58.3)	25.0(23.1-40.0)	0.44(0.33-0.50)	✓
		0.20	58.3(50.0-70.0)	50.0(50.0-50.0)	0.50(0.47-0.52)	✓
	GPC	0.05	11.1(0.0-26.0)	18.8(9.4-32.6)	0.92(0.91-0.94)	✓
		0.10	22.2(11.1-23.6)	25.0(12.5-40.3)	0.90(0.89-0.92)	✓
		0.15	25.0(23.6-50.0)	55.6(40.3-65.3)	0.88(0.88-0.89)	✓
		0.20	33.3(29.2-33.3)	25.0(20.8-40.3)	0.83(0.82-0.88)	✓
Pima Indians Diabetes	NN (linear)	0.01				×
		0.05	11.4(10.6-13.8)	11.4(8.4-13.8)	0.11(0.06-0.14)	✓
		0.10	21.8(19.8-23.4)	20.6(18.5-22.4)	0.09(0.05-0.13)	✓
		0.15	29.7(29.3-33.3)	29.0(25.7-30.6)	0.13(0.10-0.13)	✓
		0.20	43.2(36.6-44.4)	33.3(32.3-34.1)	0.21(0.20-0.21)	✓
	NN (sigmoid)	0.01				×
		0.05				×
		0.10	24.7(23.4-26.1)	22.5(18.9-25.3)	0.80(0.79-0.81)	✓
		0.15	29.4(28.2-29.7)	29.4(28.2-31.2)	0.79(0.76-0.80)	✓
		0.20	38.2(37.8-38.5)	36.1(35.3-40.6)	0.74(0.74-0.77)	✓
Physionet Challenge 2012	NN (linear)	0.01	4.1(3.2-5.0)	3.0(2.1-4.5)	0.06(0.00-0.14)	✓
		0.05	18.4(18.0-21.4)	19.5(16.7-21.3)	0.35(0.18-0.39)	✓
		0.10	35.2(34.4-37.6)	31.2(27.7-33.3)	0.43(0.33-0.46)	✓
		0.15	48.6(45.7-50.0)	40.7(36.2-42.2)	0.46(0.45-0.49)	✓
		0.20	56.4(55.5-59.6)	50.0(44.7-51.8)	0.49(0.48-0.50)	✓
	NN (sigmoid)	0.01	3.7(1.9-4.7)	3.8(2.9-4.8)	0.98(0.98-0.98)	✓
		0.05	13.9(12.4-14.0)	17.6(13.3-17.8)	0.94(0.93-0.94)	✓
		0.10	26.9(25.2-29.5)	34.3(26.7-35.2)	0.89(0.88-0.89)	✓
		0.15	41.7(35.5-42.9)	45.8(43.5-46.7)	0.86(0.85-0.88)	✓
		0.20	48.1(45.3-52.4)	55.1(54.6-55.2)	0.84(0.83-0.84)	✓
Fashion MNIST	CNN	0.01	4.1(3.2-4.8)	6.4(5.7-7.1)	0.98(0.98-0.98)	✓
		0.05	13.8(12.0-16.2)	22.5(21.3-24.4)	0.92(0.92-0.93)	✓
		0.10	28.2(24.8-28.4)	43.0(42.2-43.9)	0.87(0.86-0.87)	✓
		0.15	40.3(38.8-42.2)	57.3(54.5-57.8)	0.83(0.83-0.83)	✓
		0.20	50.7(48.6-53.3)	70.7(70.2-71.9)	0.79(0.79-0.80)	✓

Table A.3: Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \left(\mathbf{x}_n^{(train)}, \hat{h}(\mathbf{x}_n^{(train)}) \right), \boldsymbol{\psi}_n^{(train)} \right\}$ for classification tasks. The loss reduction symbolized by $r_{val}(\%)$ and $r_{test}(\%)$ are the percentages of the loss that could be corrected in validation and test set, respectively. The signal rate $\hat{\rho}$ shows how often the human assistance is required in the test set. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$r_{val}(\%)$	$\hat{\rho}$	$r_{test}(\%)$	H_0		
Social Network Ads	SVM	0.05	18.3(15.8-21.2)	0.03(0.02-0.05)	16.2(11.3-23.3)	✓		
		0.10	33.3(27.8-37.5)	0.08(0.05-0.11)	41.7(30.0-53.9)	✓		
		0.15	46.7(44.4-50.0)	0.10(0.09-0.11)	41.7(40.0-70.0)	✓		
		0.20	61.1(53.3-66.7)	0.18(0.12-0.18)	70.0(50.0-75.0)	✓		
		GPC	0.05				×	
	GPC	0.10	39.6(36.5-46.3)	0.09(0.06-0.10)	20.8(15.3-28.1)	✓		
		0.15	44.4(37.5-50.0)	0.10(0.08-0.15)	22.2(16.7-25.0)	✓		
		0.20	58.3(51.4-59.2)	0.14(0.12-0.17)	25.0(20.8-29.2)	✓		
		Pima Indians Diabetes	NN (linear)	0.01	3.6(3.5-4.2)	0.01(0.01-0.01)	0.0(0.0-1.4)	✓
				0.05	10.7(10.5-13.4)	0.06(0.04-0.07)	14.3(10.4-15.5)	✓
NN (sigmoid)	0.10		21.3(18.5-24.9)	0.12(0.10-0.14)	24.0(18.6-29.4)	✓		
	0.15		28.1(27.6-32.0)	0.16(0.15-0.17)	31.7(22.6-37.1)	✓		
	0.20		44.0(39.3-48.3)	0.20(0.18-0.22)	42.9(34.1-43.2)	✓		
Physionet Challenge 2012	NN (linear)	0.01	3.9(3.7-4.0)	0.01(0.01-0.01)	0.0(0.0-1.4)	✓		
		0.05	11.1(10.5-11.8)	0.06(0.05-0.06)	14.4(10.7-17.7)	✓		
	NN (sigmoid)	0.10	22.8(21.6-24.6)	0.12(0.10-0.13)	26.2(21.4-30.4)	✓		
		0.15	32.3(31.2-38.5)	0.14(0.14-0.19)	32.4(31.2-38.2)	✓		
		0.20	40.6(38.7-46.1)	0.21(0.19-0.24)	38.5(37.8-52.9)	✓		
Fashion MNIST	NN (linear)	0.01				×		
		0.05	18.4(18.1-21.4)	0.05(0.05-0.06)	17.6(16.4-18.1)	✓		
	NN (sigmoid)	0.10	33.0(32.6-34.2)	0.10(0.10-0.10)	32.0(27.7-32.4)	✓		
		0.15	43.8(43.4-44.7)	0.15(0.15-0.15)	40.7(38.3-46.4)	✓		
		0.20	53.9(50.0-56.4)	0.20(0.19-0.22)	48.9(47.2-56.2)	✓		
		0.01	12.0(11.7-14.3)	0.04(0.04-0.05)	13.1(9.3-16.2)	✓		
		0.10	29.9(29.3-31.5)	0.11(0.10-0.12)	29.9(24.8-33.3)	✓		
0.15	42.9(42.7-45.7)	0.15(0.15-0.17)	40.7(39.2-41.9)	✓				
0.20	54.5(54.3-56.2)	0.20(0.20-0.21)	50.5(50.0-56.1)	✓				
Fashion MNIST	CNN	0.01	6.0(5.2-6.5)	0.01(0.01-0.01)	5.6(4.7-7.9)	✓		
		0.05	24.6(24.1-27.1)	0.05(0.05-0.06)	25.7(20.6-26.7)	✓		
		0.10	43.1(40.9-43.6)	0.10(0.10-0.11)	45.0(38.1-45.8)	✓		
		0.15	58.7(58.1-58.8)	0.15(0.15-0.15)	57.4(54.2-58.2)	✓		
		0.20	70.5(69.8-71.1)	0.20(0.20-0.20)	69.1(68.1-70.0)	✓		

Table A.4: Loss reduction comparison in test set with $f(x)$ trained on $\Delta = \left\{ \left(\mathbf{x}_n^{(train)}, \hat{h}(\mathbf{x}_n^{(train)}) \right), \psi_n^{(train)} \right\}$ compared against $g(x)$. The loss reduction symbolized by $r_{test}(\%)$ is the percentage of the loss that could be corrected in test set which is evaluated for $f(x)$ and $g(x)$. The Jaccard index J shows how similar the instance selection is for both functions. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$f(x) > \eta$	$g(x) > \theta$	J	Reject H_0	
			$r_{test}(\%)$	$r_{test}(\%)$			
Social Network Ads	SVM	0.05	16.2(11.3-23.3)	9.2(8.2-10.6)	0.00(0.00-0.00)	✓	
		0.10	41.7(30.0-53.9)	20.0(12.5-23.1)	0.22(0.00-0.45)	✓	
		0.15	41.7(40.0-70.0)	25.0(23.1-40.0)	0.38(0.33-0.50)	✓	
		0.20	70.0(50.0-75.0)	50.0(50.0-50.0)	0.55(0.53-0.64)	✓	
		GPC	0.05				×
		0.10	20.8(15.3-28.1)	37.5(18.8-51.4)	0.90(0.89-0.90)	✓	
		0.15	22.2(16.7-25.0)	25.0(10.0-55.6)	0.94(0.90-0.94)	✓	
		0.20	25.0(20.8-29.2)	25.0(20.8-40.3)	0.90(0.90-0.92)	✓	
	Pima Indians Diabetes	NN (linear)	0.01	0.0(0.0-1.4)	0.0(0.0-4.2)	0.00(0.00-0.00)	✓
			0.05	14.3(10.4-15.5)	16.1(13.8-17.8)	0.00(0.00-0.03)	✓
0.10			24.0(18.6-29.4)	19.1(18.5-20.2)	0.14(0.09-0.18)	✓	
0.15			31.7(22.6-37.1)	29.0(25.7-30.6)	0.16(0.11-0.18)	✓	
0.20			42.9(34.1-43.2)	33.3(32.3-34.1)	0.21(0.20-0.28)	✓	
NN (sigmoid)		0.01	0.0(0.0-1.4)	9.4(4.7-9.8)	0.97(0.96-0.97)	✓	
		0.05	14.4(10.7-17.7)	13.1(7.0-18.1)	0.90(0.89-0.91)	✓	
		0.10	26.2(21.4-30.4)	22.5(18.9-25.3)	0.81(0.79-0.84)	✓	
		0.15	32.4(31.2-38.2)	29.4(28.2-31.2)	0.77(0.77-0.83)	✓	
		0.20	38.5(37.8-52.9)	36.1(35.3-40.6)	0.76(0.73-0.79)	✓	
Physionet Challenge 2012	NN (linear)	0.01				×	
		0.05	17.6(16.4-18.1)	19.5(16.7-21.3)	0.71(0.38-0.75)	✓	
		0.10	32.0(27.7-32.4)	31.2(27.7-33.3)	0.72(0.69-0.80)	✓	
		0.15	40.7(38.3-46.4)	40.7(36.2-42.2)	0.81(0.80-0.84)	✓	
		0.20	48.9(47.2-56.2)	50.0(44.7-51.8)	0.78(0.74-0.81)	✓	
	NN (sigmoid)	0.01				×	
		0.05	13.1(9.3-16.2)	17.6(13.3-17.8)	0.95(0.95-0.97)	✓	
		0.10	29.9(24.8-33.3)	34.3(26.7-35.2)	0.94(0.93-0.95)	✓	
		0.15	40.7(39.2-41.9)	45.8(43.5-46.7)	0.93(0.93-0.94)	✓	
		0.20	50.5(50.0-56.1)	55.1(54.6-55.2)	0.93(0.93-0.94)	✓	
Fashion MNIST	CNN	0.01	5.6(4.7-7.9)	6.0(4.3-7.2)	0.98(0.98-0.98)	✓	
		0.05	25.7(20.6-26.7)	24.7(20.3-24.9)	0.95(0.93-0.95)	✓	
		0.10	45.0(38.1-45.8)	41.3(38.5-42.6)	0.91(0.90-0.91)	✓	
		0.15	57.4(54.2-58.2)	53.0(51.9-57.8)	0.90(0.89-0.90)	✓	
		0.20	69.1(68.1-70.0)	66.3(65.2-70.9)	0.90(0.90-0.91)	✓	

Table A.5: Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \mathbf{T} \left(\mathbf{x}_n^{(train)}, \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \boldsymbol{\psi}_n^{(train)} \right\}$ for classification tasks. The loss reduction symbolized by $r_{val}(\%)$ and $r_{test}(\%)$ are the percentages of the loss that could be corrected in validation and test set, respectively. The signal rate $\hat{\rho}$ shows how often the human assistance is required in the test set. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$r_{val}(\%)$	$\hat{\rho}$	$r_{test}(\%)$	H_0
Social Network Ads	SVM	0.05	22.5(18.3-25.0)	0.01(0.01-0.03)	13.9(11.9-16.5)	✓
		0.10	33.3(27.8-37.5)	0.06(0.06-0.15)	50.0(25.0-66.7)	✓
		0.15	50.0(46.7-53.3)	0.10(0.09-0.18)	70.0(40.0-75.0)	✓
		0.20	61.1(60.0-66.7)	0.20(0.12-0.21)	70.0(60.0-83.3)	✓
	GPC	0.05	25.0(24.3-26.2)	0.06(0.04-0.06)	18.1(8.3-25.0)	✓
		0.10	37.5(37.5-44.4)	0.09(0.06-0.10)	22.2(20.0-25.0)	✓
		0.15	37.5(37.5-55.6)	0.14(0.09-0.15)	22.2(20.0-25.0)	✓
		0.20	52.8(47.9-56.7)	0.14(0.09-0.19)	26.1(20.8-41.2)	✓
Pima Indians Diabetes	NN (linear)	0.01	4.0(3.7-4.4)	0.01(0.01-0.02)	2.8(1.4-6.3)	✓
		0.05				×
		0.10	24.1(22.7-27.6)	0.11(0.11-0.12)	23.9(19.7-28.8)	✓
		0.15	34.5(28.1-40.0)	0.14(0.13-0.16)	31.4(27.0-31.7)	✓
		0.20	44.0(37.5-44.8)	0.19(0.17-0.21)	35.1(31.7-42.9)	✓
	NN (sigmoid)	0.01				×
		0.05	12.0(11.1-13.5)	0.06(0.06-0.06)	10.0(8.3-12.7)	✓
		0.10	22.6(21.4-26.9)	0.11(0.10-0.12)	25.0(22.2-26.5)	✓
		0.15	29.0(28.1-34.6)	0.14(0.13-0.14)	28.1(27.8-29.4)	✓
		0.20	40.6(38.7-42.3)	0.19(0.19-0.22)	35.1(33.3-41.7)	✓
Physionet Challenge 2012	NN (linear)	0.01	4.9(4.2-5.3)	0.01(0.01-0.01)	1.8(1.6-3.2)	✓
		0.05	19.7(18.0-20.8)	0.05(0.05-0.06)	17.6(16.0-22.7)	✓
		0.10	36.8(31.5-39.0)	0.10(0.10-0.11)	33.6(33.0-38.4)	✓
		0.15	48.7(48.3-50.6)	0.15(0.15-0.15)	46.8(44.0-47.3)	✓
		0.20	57.3(56.4-58.4)	0.21(0.21-0.21)	55.5(51.9-57.5)	✓
	NN (sigmoid)	0.01	4.3(3.7-4.4)	0.01(0.01-0.01)	4.7(4.3-5.6)	✓
		0.05	16.9(15.7-18.8)	0.05(0.05-0.06)	16.8(12.0-17.1)	✓
		0.10	30.0(28.0-31.2)	0.10(0.10-0.11)	32.7(29.6-33.3)	✓
		0.15	42.7(41.6-43.8)	0.16(0.16-0.17)	47.7(39.8-48.6)	✓
		0.20	53.1(50.6-53.3)	0.20(0.20-0.22)	54.2(42.6-55.2)	✓
Fashion MNIST	CNN	0.01	4.8(4.6-5.1)	0.01(0.01-0.01)	5.1(4.3-5.5)	✓
		0.05	20.8(20.4-20.9)	0.05(0.05-0.05)	16.9(16.7-18.2)	✓
		0.10	33.5(33.2-34.4)	0.10(0.10-0.10)	32.9(30.2-33.9)	✓
		0.15	45.2(44.2-46.8)	0.15(0.15-0.15)	45.1(40.0-46.1)	✓
		0.20	54.8(53.8-59.0)	0.20(0.20-0.20)	54.7(54.4-56.1)	✓

Table A.6: Loss reduction comparison in test set with $f(x)$ trained on $\Delta = \left\{ \mathbf{T} \left(\mathbf{x}_n^{(train)}, \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \psi_n^{(train)} \right\}$ compared against $g(x)$. The loss reduction symbolized by $r_{test}(\%)$ is the percentage of the loss that could be corrected in test set which is evaluated for $f(x)$ and $g(x)$. The Jaccard index J shows how similar the instance selection is for both functions. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$f(x) > \eta$	$g(x) > \theta$	J	Reject H_0
			$r_{test}(\%)$	$r_{test}(\%)$		
Social Network Ads	SVM	0.05	13.9(11.9-16.5)	10.0(9.4-10.6)	0.10(0.00-0.28)	✓
		0.10	50.0(25.0-66.7)	20.0(12.5-23.1)	0.12(0.09-0.38)	✓
		0.15	70.0(40.0-75.0)	25.0(23.1-40.0)	0.35(0.31-0.50)	✓
		0.20	70.0(60.0-83.3)	50.0(50.0-50.0)	0.54(0.50-0.57)	✓
	GPC	0.05	18.1(8.3-25.0)	18.8(9.4-32.6)	0.92(0.92-0.93)	✓
		0.10	22.2(20.0-25.0)	25.0(10.0-50.0)	0.92(0.91-0.94)	✓
		0.15	22.2(20.0-25.0)	25.0(10.0-55.6)	0.94(0.92-0.95)	✓
		0.20	26.1(20.8-41.2)	32.5(22.9-43.9)	0.89(0.88-0.91)	✓
Pima Indians Diabetes	NN (linear)	0.01	2.8(1.4-6.3)	8.3(4.2-8.4)	0.00(0.00-0.00)	✓
		0.05				×
		0.10	23.9(19.7-28.8)	20.6(18.5-22.4)	0.10(0.06-0.15)	✓
		0.15	31.4(27.0-31.7)	29.0(25.7-30.6)	0.16(0.14-0.22)	✓
		0.20	35.1(31.7-42.9)	33.3(32.3-34.1)	0.24(0.17-0.25)	✓
	NN (sigmoid)	0.01				×
		0.05	10.0(8.3-12.7)	8.2(6.8-10.7)	0.91(0.90-0.93)	✓
		0.10	25.0(22.2-26.5)	24.3(20.6-25.0)	0.82(0.80-0.83)	✓
		0.15	28.1(27.8-29.4)	29.4(28.2-31.2)	0.79(0.78-0.80)	✓
		0.20	35.1(33.3-41.7)	36.1(35.3-40.6)	0.74(0.72-0.78)	✓
Physionet Challenge 2012	NN (linear)	0.01	1.8(1.6-3.2)	3.4(2.1-5.0)	0.11(0.07-0.18)	✓
		0.05	17.6(16.0-22.7)	19.5(16.7-21.3)	0.38(0.32-0.46)	✓
		0.10	33.6(33.0-38.4)	31.2(27.7-33.3)	0.45(0.39-0.48)	✓
		0.15	46.8(44.0-47.3)	40.7(36.2-42.2)	0.49(0.48-0.51)	✓
		0.20	55.5(51.9-57.5)	50.0(44.7-51.8)	0.56(0.52-0.58)	✓
	NN (sigmoid)	0.01	4.7(4.3-5.6)	3.8(2.8-4.3)	0.98(0.98-0.98)	✓
		0.05	16.8(12.0-17.1)	17.6(13.3-17.8)	0.94(0.93-0.94)	✓
		0.10	32.7(29.6-33.3)	34.3(26.7-35.2)	0.91(0.90-0.92)	✓
		0.15	47.7(39.8-48.6)	45.8(43.5-46.7)	0.90(0.89-0.92)	✓
		0.20	54.2(42.6-55.2)	55.1(54.6-55.2)	0.90(0.89-0.91)	✓
Fashion MNIST	CNN	0.01	5.1(4.3-5.5)	6.8(6.0-8.2)	0.98(0.97-0.98)	✓
		0.05	16.9(16.7-18.2)	23.1(22.9-24.1)	0.92(0.91-0.92)	✓
		0.10	32.9(30.2-33.9)	42.5(37.4-43.6)	0.87(0.86-0.87)	✓
		0.15	45.1(40.0-46.1)	56.8(54.5-60.8)	0.83(0.83-0.84)	✓
		0.20	54.7(54.4-56.1)	67.7(65.5-71.0)	0.82(0.81-0.82)	✓

Table A.7: Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ (\sigma_p \circ \dots \circ \sigma_1) (\mathbf{x}_n^{(train)}), \psi_n^{(train)} \right\}$ for classification tasks. The loss reduction symbolized by $r_{val}(\%)$ and $r_{test}(\%)$ are the percentages of the loss that could be corrected in validation and test set, respectively. The signal rate $\hat{\rho}$ shows how often the human assistance is required in the test set. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$r_{val}(\%)$	$\hat{\rho}$	$r_{test}(\%)$	H_0
Fashion MNIST	CNN	0.01	5.1(4.9-5.2)	0.01(0.01-0.01)	6.5(6.1-6.6)	✓
		0.05	21.8(21.7-21.9)	0.04(0.04-0.05)	19.3(17.9-24.1)	✓
		0.10	40.3(39.2-41.1)	0.10(0.09-0.10)	39.4(35.4-41.3)	✓
		0.15	56.3(54.6-58.9)	0.15(0.15-0.15)	54.7(52.7-57.4)	✓
		0.20	69.3(68.6-69.4)	0.20(0.20-0.20)	65.6(64.0-67.9)	✓
CIFAR10	CNN	0.01	1.9(1.9-2.0)	0.01(0.01-0.01)	1.5(1.4-2.2)	✓
		0.05	9.1(8.8-9.3)	0.05(0.05-0.05)	8.4(8.2-9.7)	✓
		0.10	16.9(16.7-17.2)	0.10(0.10-0.10)	16.7(16.5-18.2)	✓
		0.15	24.5(24.2-24.6)	0.14(0.14-0.15)	23.0(22.9-24.5)	✓
		0.20	32.3(31.8-32.5)	0.19(0.18-0.19)	30.1(29.4-30.5)	✓
IMDB	RNN (linear)	0.01	3.4(3.2-3.9)	0.01(0.01-0.01)	3.3(2.1-4.2)	✓
		0.05	15.5(14.5-16.6)	0.05(0.05-0.05)	17.2(16.7-17.6)	✓
		0.10	26.1(26.0-28.9)	0.09(0.09-0.10)	27.6(26.2-28.9)	✓
		0.15	36.1(36.0-38.4)	0.15(0.13-0.15)	38.2(35.5-38.5)	✓
		0.20	46.2(45.0-46.9)	0.19(0.19-0.20)	47.5(44.0-47.6)	✓
	RNN (sigmoid)	0.01	3.2(2.9-3.7)	0.01(0.01-0.01)	3.4(3.4-4.0)	✓
		0.05	14.3(14.1-15.3)	0.05(0.05-0.05)	16.4(13.9-16.9)	✓
		0.10	28.7(28.1-28.8)	0.11(0.10-0.11)	30.1(29.1-31.0)	✓
		0.15	39.6(39.0-39.8)	0.15(0.15-0.15)	39.6(38.6-40.2)	✓
		0.20	47.9(47.0-48.9)	0.20(0.20-0.21)	48.9(46.9-49.7)	✓
German	NN	0.05				×
Credit Data	(linear)	0.10	20.0(20.0-27.7)	0.08(0.08-0.10)	18.8(15.4-20.0)	✓
		0.15	31.1(30.2-34.0)	0.14(0.10-0.16)	25.0(22.7-28.3)	✓
		0.20	44.4(42.5-46.8)	0.18(0.18-0.19)	37.5(32.2-39.4)	✓
	NN (sigmoid)	0.01				×
		0.05				×
		0.10	25.4(24.0-26.4)	0.08(0.06-0.11)	12.5(9.4-16.8)	✓
		0.15	32.7(30.5-36.0)	0.15(0.12-0.17)	21.7(13.8-29.2)	✓
		0.20	43.2(39.5-44.9)	0.18(0.15-0.21)	28.7(21.8-34.2)	✓

Table A.8: Loss reduction comparison in test set with $f(x)$ trained on $\Delta = \left\{ (\sigma_p \circ \dots \circ \sigma_1)(\mathbf{x}_n^{(train)}), \psi_n^{(train)} \right\}$ compared against $g(x)$. The loss reduction symbolized by $r_{test}(\%)$ is the percentage of the loss that could be corrected in test set which is evaluated for $f(x)$ and $g(x)$. The Jaccard index J shows how similar the instance selection is for both functions. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$f(x) > \eta$	$g(x) > \theta$	J	Reject H_0	
			$r_{test}(\%)$	$r_{test}(\%)$			
Fashion MNIST	CNN	0.01	6.5(6.1-6.6)	6.1(5.9-7.1)	0.98(0.98-0.98)	✓	
		0.05	19.3(17.9-24.1)	24.5(21.5-24.5)	0.93(0.93-0.93)	✓	
		0.10	39.4(35.4-41.3)	41.7(37.4-42.5)	0.90(0.89-0.91)	✓	
		0.15	54.7(52.7-57.4)	56.1(51.4-58.3)	0.88(0.88-0.89)	✓	
		0.20	65.6(64.0-67.9)	68.9(66.7-71.3)	0.88(0.87-0.88)	✓	
CIFAR10	CNN	0.01	1.5(1.4-2.2)	1.5(1.2-2.3)	0.99(0.98-0.99)	✓	
		0.05	8.4(8.2-9.7)	9.3(8.2-9.3)	0.93(0.93-0.93)	✓	
		0.10	16.7(16.5-18.2)	16.3(16.3-18.9)	0.88(0.88-0.89)	✓	
		0.15	23.0(22.9-24.5)	25.9(24.6-26.5)	0.83(0.83-0.85)	✓	
		0.20	30.1(29.4-30.5)	30.1(29.8-33.8)	0.80(0.79-0.83)	✓	
IMDB	RNN (linear)	0.01	3.3(2.1-4.2)	2.2(2.1-2.4)	0.00(0.00-0.05)	✓	
		0.05	17.2(16.7-17.6)	12.2(10.9-13.5)	0.34(0.29-0.35)	✓	
		0.10	27.6(26.2-28.9)	24.1(24.0-25.6)	0.45(0.38-0.48)	✓	
		0.15	38.2(35.5-38.5)	36.8(36.3-37.1)	0.49(0.45-0.57)	✓	
		0.20	47.5(44.0-47.6)	45.6(44.6-46.4)	0.56(0.50-0.57)	✓	
	RNN (sigmoid)	0.01	3.4(3.4-4.0)	3.2(3.0-3.7)	0.98(0.98-0.98)	✓	
		0.05	16.4(13.9-16.9)	14.7(12.9-16.9)	0.96(0.94-0.96)	✓	
		0.10	30.1(29.1-31.0)	28.2(26.1-28.7)	0.93(0.91-0.94)	✓	
		0.15	39.6(38.6-40.2)	39.3(35.9-41.1)	0.90(0.90-0.92)	✓	
		0.10	18.8(15.4-20.0)	13.6(13.5-21.7)	0.20(0.17-0.46)	✓	
Credit	(linear)	0.10	18.8(15.4-20.0)	13.6(13.5-21.7)	0.20(0.17-0.46)	✓	
German	NN	0.05				×	
Credit Data	(linear)	0.10	18.8(15.4-20.0)	13.6(13.5-21.7)	0.20(0.17-0.46)	✓	
		0.15	25.0(22.7-28.3)	19.2(18.6-25.0)	0.29(0.20-0.70)	✓	
		0.20	37.5(32.2-39.4)	30.8(27.1-31.7)	0.51(0.35-0.57)	✓	
		NN	0.01				×
		(sigmoid)	0.05				×
		0.10	12.5(9.4-16.8)	17.5(15.2-21.0)	0.89(0.88-0.91)	✓	
		0.15	21.7(13.8-29.2)	27.6(23.4-31.5)	0.85(0.85-0.86)	✓	
		0.20	28.7(21.8-34.2)	33.5(30.0-39.9)	0.81(0.81-0.83)	✓	

Table A.9: Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \left((\sigma_p \circ \dots \circ \sigma_1) \left(\mathbf{x}_n^{(train)} \right), \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \psi_n^{(train)} \right\}$ for classification tasks. The loss reduction symbolized by $r_{val}(\%)$ and $r_{test}(\%)$ are the percentages of the loss that could be corrected in validation and test set, respectively. The signal rate $\hat{\rho}$ shows how often the human assistance is required in the test set. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$r_{val}(\%)$	$\hat{\rho}$	$r_{test}(\%)$	H_0		
Fashion MNIST	CNN	0.01	6.5(6.3-6.5)	0.01(0.01-0.01)	3.6(3.3-6.5)	✓		
		0.05	26.5(25.4-27.1)	0.05(0.05-0.05)	25.6(25.4-25.8)	✓		
		0.10	45.4(45.2-47.0)	0.10(0.10-0.11)	45.6(43.5-46.2)	✓		
		0.15	61.8(61.5-64.2)	0.16(0.15-0.16)	61.7(58.6-65.9)	✓		
		0.20	75.6(74.8-75.8)	0.21(0.20-0.21)	74.2(73.7-74.8)	✓		
CIFAR10	CNN	0.01	2.0(2.0-2.2)	0.01(0.01-0.01)	2.3(2.2-2.4)	✓		
		0.05	9.3(9.1-9.6)	0.05(0.04-0.05)	8.8(7.9-9.6)	✓		
		0.10	18.2(18.0-18.5)	0.10(0.09-0.10)	18.7(17.3-19.2)	✓		
		0.15	27.2(26.5-27.2)	0.15(0.14-0.15)	27.7(23.7-27.7)	✓		
		0.20	35.5(34.5-35.5)	0.20(0.19-0.20)	35.0(31.7-37.5)	✓		
IMDB	RNN (linear)	0.01	3.2(3.2-3.9)	0.01(0.01-0.01)	3.3(2.1-4.2)	✓		
		0.05	15.5(14.5-16.6)	0.05(0.05-0.05)	16.7(16.6-17.3)	✓		
		0.10	26.3(26.0-28.9)	0.09(0.09-0.10)	27.9(25.9-28.9)	✓		
		0.15	36.0(36.0-38.6)	0.14(0.13-0.16)	38.2(35.5-38.2)	✓		
		0.20	46.0(44.8-46.9)	0.19(0.19-0.20)	46.9(44.0-47.2)	✓		
	RNN (sigmoid)	0.01	3.6(2.5-3.8)	0.01(0.01-0.01)	4.7(3.0-5.0)	✓		
		0.05	14.9(14.8-15.6)	0.05(0.05-0.05)	17.8(15.5-17.9)	✓		
		0.10	28.2(28.1-28.7)	0.10(0.10-0.10)	28.3(27.8-29.0)	✓		
		0.15	38.7(36.5-39.3)	0.16(0.15-0.16)	39.4(39.0-40.8)	✓		
		0.20	46.8(44.1-48.5)	0.21(0.21-0.21)	48.8(48.1-49.8)	✓		
		German Credit Data	NN (linear)	0.01				×
				0.05	13.1(12.7-14.5)	0.04(0.04-0.04)	8.7(7.3-9.8)	✓
				0.10	25.5(23.3-26.7)	0.08(0.08-0.09)	15.2(15.0-16.9)	✓
				0.15	32.6(31.9-40.0)	0.13(0.12-0.14)	26.7(23.7-32.7)	✓
0.20	44.4(42.5-47.5)			0.20(0.18-0.20)	40.9(33.3-42.3)	✓		
NN (sigmoid)	0.01	2.9(2.7-2.9)	0.00(0.00-0.01)	0.0(0.0-1.1)	✓			
	0.05	13.3(11.4-15.8)	0.04(0.04-0.05)	10.6(8.9-11.4)	✓			
	0.10	22.6(21.2-25.8)	0.12(0.10-0.13)	21.1(16.1-25.2)	✓			
	0.15	32.4(28.9-34.2)	0.17(0.13-0.17)	27.3(26.7-29.8)	✓			
	0.20	37.1(35.1-42.1)	0.22(0.21-0.24)	38.6(31.9-40.8)	✓			

Table A.10: Loss reduction comparison in test set with $f(x)$ trained on $\Delta = \left\{ \left((\sigma_p \circ \dots \circ \sigma_1) \left(\mathbf{x}_n^{(train)} \right), \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \psi_n^{(train)} \right\}$ compared against $g(x)$. The loss reduction symbolized by $r_{test}(\%)$ is the percentage of the loss that could be corrected in test set which is evaluated for $f(x)$ and $g(x)$. The Jaccard index J shows how similar the instance selection is for both functions. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$f(x) > \eta$	$g(x) > \theta$	J	Reject H_0
			$r_{test}(\%)$	$r_{test}(\%)$		
Fashion MNIST	CNN	0.01	3.6(3.3-6.5)	7.2(6.5-7.2)	0.98(0.98-0.98)	✓
		0.05	25.6(25.4-25.8)	24.4(22.5-25.4)	0.93(0.92-0.93)	✓
		0.10	45.6(43.5-46.2)	39.6(35.5-41.2)	0.91(0.91-0.91)	✓
		0.15	61.7(58.6-65.9)	59.6(52.2-59.9)	0.91(0.90-0.91)	✓
		0.20	74.2(73.7-74.8)	71.2(64.9-72.8)	0.89(0.89-0.92)	✓
CIFAR10	CNN	0.01	2.3(2.2-2.4)	1.9(1.7-2.4)	0.98(0.98-0.98)	✓
		0.05	8.8(7.9-9.6)	8.2(8.2-8.3)	0.93(0.92-0.94)	✓
		0.10	18.7(17.3-19.2)	16.8(16.2-17.4)	0.87(0.86-0.89)	✓
		0.15	27.7(23.7-27.7)	25.0(23.0-26.5)	0.84(0.83-0.85)	✓
		0.20	35.0(31.7-37.5)	32.4(29.4-34.2)	0.81(0.79-0.82)	✓
IMDB	RNN (linear)	0.01	3.3(2.1-4.2)	2.2(2.1-2.4)	0.00(0.00-0.05)	✓
		0.05	16.7(16.6-17.3)	12.2(10.9-13.5)	0.33(0.29-0.35)	✓
		0.10	27.9(25.9-28.9)	24.1(24.0-25.6)	0.46(0.37-0.49)	✓
		0.15	38.2(35.5-38.2)	36.8(36.3-37.1)	0.49(0.43-0.56)	✓
		0.20	46.9(44.0-47.2)	45.6(44.6-46.4)	0.56(0.52-0.57)	✓
	RNN (sigmoid)	0.01	4.7(3.0-5.0)	3.4(2.3-4.0)	0.98(0.98-0.98)	✓
		0.05	17.8(15.5-17.9)	13.1(12.8-14.6)	0.95(0.94-0.95)	✓
		0.10	28.3(27.8-29.0)	28.2(26.9-28.3)	0.91(0.91-0.92)	✓
		0.15	39.4(39.0-40.8)	39.3(37.2-39.7)	0.89(0.88-0.90)	✓
		0.20	48.8(48.1-49.8)	47.5(47.1-51.5)	0.88(0.87-0.89)	✓
German Credit Data	NN (linear)	0.01				×
		0.05	8.7(7.3-9.8)	7.1(4.9-9.7)	0.07(0.05-0.12)	✓
		0.10	15.2(15.0-16.9)	13.6(13.5-21.7)	0.18(0.14-0.35)	✓
		0.15	26.7(23.7-32.7)	19.2(18.6-25.0)	0.31(0.24-0.42)	✓
		0.20	40.9(33.3-42.3)	30.8(27.1-31.7)	0.48(0.35-0.65)	✓
	NN (sigmoid)	0.01	0.0(0.0-1.1)	2.0(1.0-3.8)	0.98(0.97-0.99)	✓
		0.05	10.6(8.9-11.4)	9.3(6.8-10.6)	0.91(0.89-0.91)	✓
		0.10	21.1(16.1-25.2)	17.5(16.5-21.0)	0.87(0.85-0.87)	✓
		0.15	27.3(26.7-29.8)	31.1(24.1-32.6)	0.82(0.81-0.82)	✓
		0.20	38.6(31.9-40.8)	35.6(31.5-40.9)	0.78(0.76-0.79)	✓

Table A.11: Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \mathbf{T} \left((\sigma_p \circ \dots \circ \sigma_1) \left(\mathbf{x}_n^{(train)} \right), \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \psi_n^{(train)} \right\}$ for classification tasks. The loss reduction symbolized by $r_{val}(\%)$ and $r_{test}(\%)$ are the percentages of the loss that could be corrected in validation and test set, respectively. The signal rate $\hat{\rho}$ shows how often the human assistance is required in the test set. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$r_{val}(\%)$	$\hat{\rho}$	$r_{test}(\%)$	H_0
Fashion MNIST	CNN	0.01	6.0(5.4-6.6)	0.01(0.01-0.01)	4.3(3.6-7.0)	✓
		0.05	24.4(24.0-24.6)	0.05(0.05-0.05)	25.8(19.3-27.0)	✓
		0.10	46.2(42.9-46.3)	0.10(0.10-0.11)	43.4(43.3-46.0)	✓
		0.15	61.1(60.1-63.0)	0.16(0.15-0.16)	59.8(58.5-60.6)	✓
		0.20	74.7(72.8-75.1)	0.20(0.20-0.21)	72.6(69.3-74.9)	✓
CIFAR10	CNN	0.01	2.0(1.9-2.1)	0.01(0.01-0.01)	1.6(1.5-1.7)	✓
		0.05	9.1(9.1-9.4)	0.05(0.04-0.06)	9.2(7.8-9.3)	✓
		0.10	17.6(17.3-17.9)	0.09(0.09-0.10)	16.4(16.3-17.3)	✓
		0.15	25.9(25.3-26.1)	0.15(0.14-0.15)	24.8(24.1-26.1)	✓
		0.20	33.6(33.4-34.0)	0.20(0.19-0.20)	33.9(33.2-34.1)	✓
IMDB	RNN (linear)	0.01	3.5(2.8-3.7)	0.01(0.01-0.01)	2.4(2.1-2.5)	✓
		0.05	14.3(13.5-16.1)	0.05(0.04-0.06)	15.9(13.6-17.0)	✓
		0.10	26.3(25.3-27.9)	0.09(0.09-0.09)	25.6(25.3-27.1)	✓
		0.15	35.8(35.0-37.9)	0.14(0.14-0.15)	36.5(36.5-38.7)	✓
		0.20	44.3(43.2-46.2)	0.20(0.18-0.20)	45.3(45.1-46.1)	✓
	RNN (sigmoid)	0.01	3.7(3.5-3.7)	0.01(0.01-0.01)	3.1(3.0-3.4)	✓
		0.05	15.0(14.1-15.7)	0.05(0.05-0.05)	14.6(13.9-16.9)	✓
		0.10	28.9(26.8-29.2)	0.10(0.09-0.10)	28.8(26.2-29.1)	✓
		0.15	38.8(37.0-40.9)	0.15(0.15-0.16)	39.0(36.9-41.7)	✓
		0.20	46.3(45.4-48.1)	0.20(0.20-0.20)	47.5(46.4-48.5)	✓
German Credit Data	NN (linear)	0.01				×
		0.05	12.2(11.2-12.9)	0.04(0.04-0.04)	8.4(8.3-9.0)	✓
		0.10	25.0(23.3-26.7)	0.08(0.08-0.08)	15.2(14.6-16.7)	✓
		0.15	37.8(34.0-38.5)	0.12(0.12-0.12)	25.0(21.1-28.3)	✓
		0.20	46.1(40.4-48.9)	0.19(0.18-0.20)	33.3(30.8-41.7)	✓
	NN (sigmoid)	0.01	2.7(2.7-2.8)	0.02(0.01-0.02)	2.3(2.1-3.2)	✓
		0.05	13.2(12.1-13.3)	0.05(0.04-0.07)	8.2(6.4-9.6)	✓
		0.10	21.1(19.4-21.3)	0.10(0.09-0.12)	15.6(12.3-20.0)	✓
		0.15	27.0(26.9-29.3)	0.17(0.17-0.17)	30.6(28.9-30.9)	✓
		0.20	32.9(32.4-34.9)	0.23(0.21-0.25)	34.5(27.3-37.3)	✓

Table A.12: Loss reduction comparison in test set with $f(x)$ trained on $\Delta = \left\{ \mathbf{T} \left((\sigma_p \circ \dots \circ \sigma_1) \left(\mathbf{x}_n^{(train)} \right), \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \psi_n^{(train)} \right\}$ compared against $g(x)$. The loss reduction symbolized by $r_{test}(\%)$ is the percentage of the loss that could be corrected in test set which is evaluated for $f(x)$ and $g(x)$. The Jaccard index J shows how similar the instance selection is for both functions. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$f(x) > \eta$	$g(x) > \theta$	J	Reject H_0
			$r_{test}(\%)$	$r_{test}(\%)$		
Fashion MNIST	CNN	0.01	4.3(3.6-7.0)	7.3(4.9-8.5)	0.98(0.98-0.98)	✓
		0.05	25.8(19.3-27.0)	24.0(23.7-24.4)	0.93(0.93-0.93)	✓
		0.10	43.4(43.3-46.0)	42.9(40.5-46.4)	0.91(0.91-0.91)	✓
		0.15	59.8(58.5-60.6)	59.8(57.5-62.2)	0.89(0.89-0.91)	✓
		0.20	72.6(69.3-74.9)	71.7(71.2-73.0)	0.90(0.88-0.91)	✓
CIFAR10	CNN	0.01	1.6(1.5-1.7)	2.5(1.6-2.6)	0.98(0.98-0.99)	✓
		0.05	9.2(7.8-9.3)	9.5(8.5-10.0)	0.93(0.93-0.93)	✓
		0.10	16.4(16.3-17.3)	18.3(17.4-18.5)	0.88(0.88-0.89)	✓
		0.15	24.8(24.1-26.1)	25.8(24.8-27.0)	0.84(0.84-0.84)	✓
		0.20	33.9(33.2-34.1)	32.3(31.5-34.1)	0.80(0.80-0.82)	✓
IMDB	RNN (linear)	0.01	2.4(2.1-2.5)	2.2(2.1-2.4)	0.00(0.00-0.00)	✓
		0.05	15.9(13.6-17.0)	12.2(10.9-13.5)	0.29(0.28-0.35)	✓
		0.10	25.6(25.3-27.1)	24.1(24.0-25.6)	0.46(0.40-0.48)	✓
		0.15	36.5(36.5-38.7)	36.8(36.3-37.1)	0.50(0.44-0.59)	✓
		0.20	45.3(45.1-46.1)	45.6(44.6-46.4)	0.57(0.55-0.60)	✓
	RNN (sigmoid)	0.01	3.1(3.0-3.4)	3.0(2.8-3.7)	0.98(0.98-0.98)	✓
		0.05	14.6(13.9-16.9)	14.7(12.9-17.9)	0.96(0.94-0.96)	✓
		0.10	28.8(26.2-29.1)	27.6(26.1-28.2)	0.95(0.92-0.96)	✓
		0.15	39.0(36.9-41.7)	40.5(37.2-41.1)	0.92(0.90-0.93)	✓
		0.20	47.5(46.4-48.5)	48.3(44.9-48.8)	0.90(0.89-0.90)	✓
German Credit Data	NN (linear)	0.01				×
		0.05	8.4(8.3-9.0)	7.1(4.9-9.7)	0.10(0.04-0.16)	✓
		0.10	15.2(14.6-16.7)	13.6(13.5-21.7)	0.10(0.07-0.33)	✓
		0.15	25.0(21.1-28.3)	19.2(18.6-25.0)	0.21(0.16-0.38)	✓
		0.20	33.3(30.8-41.7)	30.8(27.1-31.7)	0.38(0.30-0.50)	✓
	NN (sigmoid)	0.01	2.3(2.1-3.2)	2.3(2.2-3.9)	0.96(0.95-0.97)	✓
		0.05	8.2(6.4-9.6)	6.8(6.5-13.4)	0.91(0.89-0.92)	✓
		0.10	15.6(12.3-20.0)	18.4(17.1-23.6)	0.80(0.80-0.85)	✓
		0.15	30.6(28.9-30.9)	32.6(31.9-33.4)	0.78(0.78-0.78)	✓
		0.20	34.5(27.3-37.3)	38.2(34.5-43.9)	0.74(0.72-0.75)	✓

Table A.13: Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \{\mathbf{x}_n^{(train)}, \boldsymbol{\psi}_n^{(train)}\}$ for regression tasks. The loss reduction symbolized by $r_{val}(\%)$ and $r_{test}(\%)$ are the percentages of the loss that could be corrected in validation and test set, respectively. The signal rate $\hat{\rho}$ shows how often the human assistance is required in the test set. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$r_{val}(\%)$	$\hat{\rho}$	$r_{test}(\%)$	H_0
Wine Quality	Linear Regression	0.01				×
		0.05	8.8(8.5-8.9)	0.04(0.04-0.06)	6.9(6.3-8.7)	✓
		0.10	15.6(15.1-16.1)	0.14(0.13-0.15)	19.9(18.2-21.4)	✓
		0.15	21.4(21.2-22.4)	0.18(0.15-0.19)	23.5(23.2-25.5)	✓
		0.20	27.9(27.8-28.5)	0.23(0.23-0.23)	30.5(27.9-32.1)	✓
	Lasso	0.01	2.3(2.2-2.4)	0.01(0.01-0.01)	1.3(0.9-2.1)	✓
		0.05	8.4(8.0-8.8)	0.05(0.04-0.06)	8.1(7.1-9.2)	✓
		0.10	15.9(15.4-16.4)	0.14(0.12-0.15)	20.8(18.3-21.4)	✓
		0.15	21.6(20.7-22.2)	0.18(0.17-0.19)	24.4(21.2-25.1)	✓
		0.20	27.6(27.4-28.1)	0.23(0.22-0.25)	30.9(29.9-33.8)	✓
	SVR	0.01				×
		0.05	8.7(8.3-9.0)	0.05(0.04-0.06)	8.1(7.4-9.4)	✓
		0.10	15.7(15.3-16.3)	0.13(0.13-0.14)	19.0(18.4-19.7)	✓
		0.15	22.0(20.8-22.6)	0.18(0.16-0.18)	24.8(23.0-25.2)	✓
		0.20	28.6(27.1-29.1)	0.23(0.22-0.23)	29.6(28.2-31.7)	✓
Boston Housing	Linear Regression	0.05	16.5(14.5-17.2)	0.05(0.04-0.08)	16.3(13.8-20.3)	✓
		0.10	28.7(26.4-30.1)	0.10(0.08-0.15)	25.6(22.6-31.7)	✓
		0.15	38.4(35.9-39.0)	0.14(0.11-0.18)	31.7(28.6-35.2)	✓
		0.20	42.3(33.9-44.9)	0.15(0.14-0.16)	32.2(24.4-34.1)	✓
	Lasso	0.05	17.0(16.6-17.7)	0.04(0.04-0.09)	14.2(12.2-19.9)	✓
		0.10	29.2(27.0-29.8)	0.11(0.09-0.15)	22.6(19.3-29.3)	✓
		0.15	37.4(35.0-38.4)	0.14(0.11-0.19)	32.0(28.8-36.6)	✓
		0.20	43.7(33.9-44.9)	0.16(0.15-0.22)	32.4(25.5-43.2)	✓
	SVR	0.05	18.5(14.1-22.9)	0.08(0.07-0.08)	19.9(18.0-23.9)	✓
		0.10	30.1(21.4-30.2)	0.09(0.08-0.11)	20.7(20.3-33.4)	✓
		0.15	39.4(28.4-45.6)	0.16(0.12-0.17)	33.4(29.1-39.5)	✓
		0.20	51.5(35.1-53.6)	0.16(0.14-0.21)	42.1(25.6-47.4)	✓

Table A.14: Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \left(\mathbf{x}_n^{(train)}, \hat{h}(\mathbf{x}_n^{(train)}) \right), \psi_n^{(train)} \right\}$ for regression tasks. The loss reduction symbolized by $r_{val}(\%)$ and $r_{test}(\%)$ are the percentages of the loss that could be corrected in validation and test set, respectively. The signal rate $\hat{\rho}$ shows how often the human assistance is required in the test set. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$r_{val}(\%)$	$\hat{\rho}$	$r_{test}(\%)$	H_0	
Red Wine Quality	Linear Regression	0.01	2.0(2.0-2.2)	0.01(0.01-0.01)	2.9(1.7-3.1)	✓	
		0.05	7.8(7.7-8.3)	0.05(0.04-0.05)	8.4(7.3-8.6)	✓	
		0.10	15.0(13.8-15.5)	0.08(0.08-0.13)	14.1(13.4-16.6)	✓	
		0.15	21.4(21.1-21.6)	0.17(0.16-0.17)	23.3(21.7-23.6)	✓	
		0.20	27.0(26.9-27.9)	0.22(0.21-0.22)	29.8(28.5-30.2)	✓	
	Lasso	0.01					×
		0.05	7.9(7.7-8.3)	0.06(0.05-0.06)	8.5(6.7-9.6)	✓	
		0.10	14.9(14.5-15.4)	0.11(0.10-0.12)	15.8(15.6-17.2)	✓	
		0.15	21.2(21.0-22.0)	0.16(0.16-0.16)	22.3(21.4-23.3)	✓	
		0.20	27.0(26.7-28.2)	0.23(0.22-0.24)	29.0(28.6-30.4)	✓	
	SVR	0.01	2.1(1.9-2.3)	0.01(0.01-0.01)	0.5(0.3-1.4)	✓	
		0.05	7.5(7.0-8.3)	0.04(0.04-0.04)	8.5(7.2-9.1)	✓	
		0.10	14.5(14.2-15.1)	0.11(0.08-0.13)	15.3(12.3-17.5)	✓	
		0.15	20.7(20.6-20.9)	0.17(0.15-0.18)	22.2(22.0-22.9)	✓	
		0.20	27.1(26.6-28.4)	0.21(0.21-0.24)	28.5(28.4-30.4)	✓	
Boston Housing	Linear Regression	0.05	14.5(11.5-17.2)	0.04(0.03-0.07)	12.0(9.3-17.0)	✓	
		0.10	29.1(26.9-30.3)	0.11(0.08-0.16)	26.3(23.2-31.4)	✓	
		0.15	37.2(28.8-38.9)	0.14(0.14-0.14)	29.4(28.3-30.9)	✓	
		0.20	44.1(33.9-44.8)	0.17(0.15-0.23)	32.2(30.2-44.1)	✓	
	Lasso	0.05	17.0(14.6-17.7)	0.04(0.04-0.09)	14.2(10.9-19.9)	✓	
		0.10	29.4(27.1-30.6)	0.12(0.09-0.16)	26.4(23.6-31.6)	✓	
		0.15	37.3(27.0-39.3)	0.13(0.13-0.14)	25.8(25.1-31.2)	✓	
		0.20	43.7(33.9-44.9)	0.17(0.15-0.23)	31.5(30.3-44.3)	✓	
	SVR	0.05	22.7(16.6-26.8)	0.04(0.04-0.06)	15.9(14.9-17.0)	✓	
		0.10	32.5(21.4-36.5)	0.10(0.09-0.12)	30.5(20.7-37.0)	✓	
		0.15	41.6(28.0-45.7)	0.15(0.11-0.16)	30.5(26.9-40.2)	✓	
		0.20	47.8(36.6-54.0)	0.16(0.14-0.31)	42.1(28.4-53.2)	✓	

Table A.15: Synergistic performance in validation set and test set assuming perfect predictions from the human with $f(x)$ trained on $\Delta = \left\{ \mathbf{T} \left(\mathbf{x}_n^{(train)}, \hat{h} \left(\mathbf{x}_n^{(train)} \right) \right), \boldsymbol{\psi}_n^{(train)} \right\}$ fore regression tasks. The loss reduction symbolized by $r_{val}(\%)$ and $r_{test}(\%)$ are the percentages of the loss that could be corrected in validation and test set, respectively. The signal rate $\hat{\rho}$ shows how often the human assistance is required in the test set. Only the results for cases where the evaluation loss for signaled instances is statistically significant (reject H_0) are reported.

Data set	Predictor	ρ	$r_{val}(\%)$	$\hat{\rho}$	$r_{test}(\%)$	H_0	
Red Wine Quality	Linear Regression	0.01				×	
		0.05	8.4(8.4-8.7)	0.05(0.04-0.08)	7.7(6.7-14.0)	✓	
		0.10	14.9(14.8-15.4)	0.12(0.09-0.15)	16.5(13.8-20.3)	✓	
		0.15	20.8(20.5-21.4)	0.15(0.14-0.18)	21.0(20.5-23.8)	✓	
		0.20	26.9(26.0-28.0)	0.23(0.19-0.25)	33.0(26.3-33.2)	✓	
	Lasso	0.01					×
		0.05	8.7(8.2-9.0)	0.07(0.05-0.08)	10.5(7.6-13.2)	✓	
		0.10	15.1(14.1-15.7)	0.12(0.09-0.12)	16.8(13.1-20.5)	✓	
		0.15	20.8(20.1-21.7)	0.15(0.11-0.18)	21.2(15.2-26.7)	✓	
		0.20	26.5(26.0-27.2)	0.23(0.18-0.26)	32.6(25.2-33.6)	✓	
	SVR	0.01					×
		0.05	8.4(8.2-8.8)	0.06(0.04-0.08)	9.2(6.6-12.7)	✓	
		0.10	15.6(15.2-16.4)	0.11(0.09-0.12)	15.9(12.8-19.4)	✓	
		0.15	21.3(21.2-21.7)	0.16(0.13-0.21)	21.5(18.4-29.8)	✓	
		0.20	27.7(26.4-28.0)	0.24(0.20-0.25)	33.0(26.1-34.2)	✓	
Boston Housing	Linear Regression	0.05	13.6(11.0-15.9)	0.04(0.04-0.06)	14.5(11.4-15.6)	✓	
		0.10	21.9(20.2-24.7)	0.11(0.09-0.14)	22.7(19.0-27.9)	✓	
		0.15	34.5(30.8-37.2)	0.18(0.15-0.21)	33.7(29.8-37.3)	✓	
		0.20	42.4(39.2-45.4)	0.20(0.17-0.24)	38.2(37.3-40.7)	✓	
	Lasso	0.05	13.7(11.1-16.0)	0.05(0.04-0.06)	15.4(11.4-17.0)	✓	
		0.10	22.1(20.4-25.0)	0.11(0.09-0.14)	22.9(19.2-28.0)	✓	
		0.15	34.8(31.2-37.4)	0.18(0.15-0.20)	32.3(29.5-35.0)	✓	
		0.20	46.0(42.2-47.8)	0.22(0.19-0.25)	39.8(37.3-43.4)	✓	
	SVR	0.05	18.8(9.7-20.7)	0.05(0.04-0.05)	8.1(7.8-15.9)	✓	
		0.10	26.4(20.3-31.9)	0.08(0.07-0.09)	19.3(14.8-20.3)	✓	
		0.15	40.3(26.4-42.7)	0.12(0.11-0.12)	34.2(24.8-36.2)	✓	
		0.20	47.2(33.6-51.6)	0.16(0.15-0.16)	41.9(30.0-43.1)	✓	